NTNU
Det skapende universitet

Å finne gammelt nytt

Bruk av NewsML i dagspressen

Marit Ingeberg

Master i informatikk
Oppgaven levert: August 2006
Hovedveileder: Ingeborg Torvik Sølvberg, IDI

Norges teknisk-naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

Sammendrag

Sentralt i hovedfagsoppgaven står det klassiske problemet; der innføring av ny teknologi krever utvikling av nye metoder og arbeidsrutiner, samtidig som de gamle rutinene får bestå tilnærmet uendret. Dette gjør at kompleksiteten i organisasjonenes systemer øker, da både gamle og nye metoder samt arbeidsrutiner må vedlikeholdes. Dette har skjedd hos Adresseavisen. Ved innføring av ny teknologi, og nye arbeidsmetoder, har man ikke hatt en helhetlig tenkning på arbeidsflyten og organisasjonsstrukturen. Dette ser man ved valg av løsning på analoge vs. digitale bilder, og på papiravis vs. nettutgaven av avisen.

Konkret tar denne hovedoppgaven for seg gjenfinningen av dokumenter i dagsaviser, der hvordan NewsML-formatet kan brukes til dette formålet står sentralt. NewsML ble lansert av International Press and Telecommunications Council (IPTC), i samarbeid med nyhetsbyråene Reuters og AFP, i 2000. Målet med NewsML er å få et helhetlig format som dekker alle ledd i nyhetsobjektets livssyklus, der samme artikkel kan lagres med ulikt språk og med støtte for alle formater og visningsmedier.

Informasjonsgjenfinning er viktig både for journalistene som skal lage en god avis, og for leserne som ønsker å søke etter artikler, for eksempel på internett. Gode søkesystemer kan være avgjørende om en avis skal greie å holde på en leser. I hovedoppgaven er søkeprosessen i seg selv beskrevet, i tillegg til ulike kriterier som et søkesystem kan vurderes etter.

Som eksempel på hvordan systemet er i en dagsavis, beskrives Adresseavisens sitt system for lagring og gjenfinning av data. Adresseavisen skiller klart mellom ulike nyhetsobjekt som for eksempel tekst, lyd, bilder eller film. Disse ulike nyhetsobjektene til papiravisen lagres i ulike databaser, mens nyhetsobjekter til internettavisen lagres i en annen database. Disse ulike nyhetsobjektene beskrives med ulikt metadataformat.

Dette gjør at når man ønsker å finne igjen alle delene av en nyhetsartikkel i Adresseavisen må man søke i ulike databaser som har ulike metadataformat. Formatet Adresseavisen bruker gjør at søkeprosessen kan bli noe tungvint. Med mindre personen som utfører søket har god kjennskap til oppbyggingen av databasene og metadataformata, risikerer man å gå glipp av verdifull informasjon.

Det er laget mange ulike standarder og formater som skal støtte både gjenfinning, utveksling og lagring av nyheter. Noen av de mest brukte blir kort beskrevet, før NewsML beskrives i detalj. NewsML gir en elegant løsning på problemet Adresseavisen har, med ulike databasser og ulike metadataformater, ved å definere alle nyhetsobjektene som et element uansett hva slags medietype det er av og hva det skal brukes til. Et NewsML-dokument inneholder en kompleks XML struktur. I denne strukturen blir en hele artikkel representert ved hjelp av et element som innholder metadata som er felles for hele nyhetsartikkelen. Inne i dette elementet kan man ha ulike nyhetskomponenter, som igjen kan innholde nye nyhetskomponenter, eller innholdselementer. De ulike nyhetskomponentene kan for eksempel skille mellom samme artikkel på ulike språk.

NewsML-standarden har enkelte obligatoriske metadatafelt, men er likevel åpen nok til at ulike mediekonsern med ulike interesser og fokus kan tilpasse et NewsML-system slik at det passer til deres behov.

For å illustrere praktisk hvordan NewsML kan bli brukt i Adresseavisen er det laget konverteringstabeller mellom ulike metadataposter i Adresseavisen, opp i mot et NewsML-dokument. Videre er det implementert en prototyp som benytter NewsML, for å vise fordeler og utfordringer ved bruk av NewsML som format. Prototypen implementerer alle NewsML-elementene. Grensesnittet i prototypen viser utfordringene med den komplekse XML-strukturen som NewsML inneholder.

Adresseavisen sitt system har en stadig økende kompleksitet, og bør på sikt endres. Ved en slik endring har Adresseavisen flere valg. Et alternativ er å ta i bruk prinsippet NewsML har med å lagre ulike nyhetselementer som like objekter. Disse objektene kan lagres ved hjelp av relasjonsdatabase, slik Adresseavisen har det i dag. Dette vil gjøre at avisen får et enhetlig og oversiktlig system der alle nyhetsobjekter har felles metadataformat, men systemet mister NewsML sin fordel ved at man kan lagre ulike versjoner av en artikkel med samme metadata. En annen løsning er å innføre NewsML med sin komplekse struktur. Man trenger ikke nødvendigvis implementere alle NewsML-elementene med engang, men forenkle strukturen noe og heller utvide ved behov.

Abstract

This thesis looks at the approach to the problem arising when new technology is being introduced to an organisation, which involves expansion of the existing computer system, without looking at the whole of the system.

This thesis studies the system used for storing and retrieval of different news objects at Adresseavisen (a local newspaper company in Norway). Further, the thesis describes different news information formats and focuses on NewsML. To illustrate the practical use of NewsML at Adresseavisen, a model for converting is made. The model demonstrates how the different metadata fields used in Adresseavisen's system can be converted to NewsML. A prototype implementing the most important components of NewsML is also made. Finally, a recommendation for further development of Adresseavisens's system is constructed.

Innholdsfortegnelse

Figurliste

Tabelloversikt

Kapittel 1 Innledning

”Jeg fant, jeg fant” sa Espen Askeladd der han vandret i skogen og plukket opp gamle skosåler og bukkehorn. Askeladden argumenterte for at materialet kunne komme til nytte ved en senere anledning og puttet det i sekken sin.

Nå ble det slik i eventyret at Askeladden fikk bruk for alt han hadde i sekken sin, men i dagens samfunn er sekken med informasjon mye større enn den Askeladden hadde. Nyheter, forskning og annen informasjon av ulikt slag skal ut til de potensielle brukere i høyt tempo. Kildene for informasjonen er mange, og det er derfor viktig å finne effektive system for å håndtere dette. I en avis er kravet til effektivitet enda høyere enn før. Tidligere kom avisene ut i papirutgave, gjerne om morgenen, og avisene hadde ikke så stor konkurranse om kundene som i dag. I dag har både TV og radio direktesendinger fra steder der noe skjer: alle tv-selskap og aviser som formidler nyhetsmateriale har nettsider, der det forventes at alt materiale er ”up to date” til enhver tid. Dette krever effektiv informasjonsbehandling, og mulighet for gjenbruk av for eksempel bilder eller tekst.

Ved gjenbruk kreves det at alt som kan være av interesse er enkelt for brukeren å finne igjen, om det er en journalist, eller ”mannen i gata”. Innholdet i sekken til Askeladden hadde vært lite verd om han ikke greide å finne det han ønsket når han trengte det, eller i verste fall ikke engang visste at han hadde det med.

1.1 Bakgrunn

Hvordan en avis fant igjen sine filer hadde jeg ikke tenkt på før jeg leste Anita Oppedal sin hovedoppgave, hvor hun viser Adresseavisens sitt system for nyhetsformidling[1]. Hun viste deres system med bruk av mange ulike metadataformater for å beskrive de ulike nyhetsobjektene i ulike arbeidsprosesser. Adresseavisen har også mange databaser der artikler, bilder, informasjon om fysisk film og lignende ligger lagret. Noen av disse databasene kan kommunisere med hverandre, mens andre ikke gjør det.
Systemet kan virke tungvint i forhold til gjenfinning av de ulike delene av et nyhetsobjekt. Hvis man for eksempel ønsker å finne et bilde som stod i en artikkel vil dette bildet for eksempel være lagret under fotografens navn, med en eller flere andre gjenfinningsnøkler. Det hjelper ikke å vite tittel på artikkelen bildet ble brukt i. Dette kan skape problemer og hindre ustrakt gjenbruk av de ulike nyhetsobjektene.

Dette gjorde meg nysgjerrig, var det andre standarder som kunne løse denne oppgaven med lagring og gjenfinning uten de samme problemene?

Det er gjort mye på dette området. International Press Telecommunications Council (IPTC) har gjennom en årrekke utviklet og publisert industristandarder for utveksling av nyhetsdata. IPTC laget i år 2000 NewsML (også kalt IPTC 2000) som en standard for uniform håndtering av nyhetsobjekter[2].

NewsML er et XMLbasert, strukturert rammeverk for nyheter, utviklet av International Press Telecommunications Council (IPTC). Det er mediauavhengig, og i stand til å representere nyhetsobjekter i alle stadier hele "livssyklusen" i et elektronisk servicemiljø: lagring, overføring, levering og arkivering[3]. Formatet brukes til nyhetsformidling av en del av de store nyhetsbyråene som Reuters og AFP og er tilgjenglig i enkelte aviser som for eksempel The Wall Street Journal[4].

NewsML introduserer en tankegang innenfor informasjonsbehandling og metadata som går ut på å behandle alle nyhetsobjekter, uansett type og hva slags medium det skal presenteres i, som like objekter. Det kan være tekst som inngår i en artikkel i papiravisen eller som skal ut på internett, det kan være lyd, bilde, film med mer. En artikkel består som regel av innhold av ulike typer, en artikkel i en nettavis innholder gjerne tekst, et eller flere bilder, link til relaterte saker, og link til forfatteren ev. fotografen som har bidratt til artikkelen. Alt dette kan kalles nyhetsobjekter som er lenket sammen til å skape en komplett artikkel. En nettavisartikkel innholder også en del reklame som kan være et forstyrrelende element, uten redaksjonelt innhold. internettbaserte nyhetsformidlere kan også formidle objekter med ”samme” innhold, som for eksempel nyhetsartikler i ulike språk, eller filmsnutter i ulikt format, kan også dele et samleobjekt, som tar seg av felles metadata.

Et nyhetsobjekt oppstår når en journalist lagrer sin tekst, bilde og lignende inn i datasystemet. Et nyhetsobjekt kan bli sendt mellom nyhetsbyråene og deres kunder, internt i mediebedriftens redaksjonelle system, og fra mediebedriften til sine lesere. Et eksempel er når et nyhetsbyrå lager en artikkel. Nyhetsobjektet tilføres metadata som blant annet innholder en beskrivende tekst som forteller hva objektet innholder, før nyhetsobjektet blir overført til nyhetsbyrået sine abonnenter, eventuelt knyttet sammen med andre nyhetsobjekter. Når den enkelte avis finner ut at de ønsker å bruke deler eller hele artikkelen, må artikkelen konverteres til det datasystemet avisen har, dersom de ikke bruker samme system som nyhetsbyrået. Selve innholdet kan beholdes slik det er, men metadatane som knyttet til må konverteres slik at det kan legges inn i avisens system. Dette krever både programvare som er spesiallaget for det systemet avisen har. Dette krever vedlikehold hvis formatet til avisen eller nyhetsbyrået endrer seg. I tillegg er ressurskrevende både i forhold til tid og maskinressurser. Når artikkelen ligger i avisen sitt system kan det enten lenkes opp mot internettsidene til avisen direkte slik den står med referanse til nyhetsbyrået, eller gjennomgå en redaksjonell endring for å presenteres som avisens egen versjon av innholdet. Alle artikler som brukes lagres i avisens databasesystem, og hentes fram ved passende anledninger hvis en journalist eller en leser i avisen ønsker å gjenfinne informasjonen.

NewsML tar sikte på å kunne støtte alle ledd i et nyhetsobjekts livssyklus, dette gjør at man slipper å konvertere alle objektene når de for eksempel kommer inn fra et nyhetsbyrå, eller skal lagres for bruk på internett. Standarden tar også sikte på å være åpen nok til at det er mulig for den enkelte bedrift å beholde eller legge til egne metadata. Det å slippe å forholde seg til ulike metadataformater på stoff som kommer inn, stoff som skal lagres internt, og stoff som skal ut til eksterne kilder vil lette det tekniske arbeidet i en avis, i og med at det kun er en standard å forholde seg til, ikke mange[5].

Metadata er et sett med strukturerte elementer som brukes til å beskrive nyhetsobjektene blant annet for å lette gjenfinning og identifiser rettigheter i forhold til copyright. NewsML definerer 5 like typer metadata: Beskrivende, redaksjonell, identifisering, rolle og fysiske metadata og de kjennetegnes på følgende måte[6]:

- Beskrivende metadata forteller hva nyhetsposten handler om, hva den referer til og hvem den kan være av interesse for. Slik metadata gjør for eksempel at redaksjonen i en avis slipper å gå inn på alle nyhetsobjektene som kommer inn fra et nyhetsbyrå for å se om artikkelen er av interesse for deres avis.
- Redaksjonelle metadata forteller hvordan, når, hvor, hvorfor og av hvem innholdet er laget, publisert og distribuert. Det forteller også hvem som eier rettighetene til nyhetsposten, bruksrestriksjoner og lignende. Slik informasjon er nyttig blant annet i forbindelse med gjenfinning av nyhetsobjekter i etterkant.
- Identifiseringsmetadata innholder vesentlige attributter til nyhetsposter eller objekter, for eksempel dato og identifikasjon.
- Rollemetadata er knyttet til enkeltstående deler, og forteller hvorfor denne delen er tilknyttet artikkelen.
- Fysiske metadata angir medietype, språk, og fysiske egenskaper ved innholdet. Dette kan være høyde, vidde, lengde, avspillingstid, fargesammensetning og lignende. Dette kommer til nytte når en skal tilby en nyhetsartikkel i ulike medium som for eksempel både på internett og over mobil er det relevant å vite, og for å kunne tilby samme artikkel i ulike språk.

1.2 Gjenfinning

Når det gjelder gjenfinning i en avis eller ved et bibliotek er det vanlig at en søker i et metadatafelt som for eksempel forfatter, emne eller tittel. Når en søker i et slikt system etter for eksempel ”elefanter” i emnefeltet, ønsker man å få tilbake alle artikler i systemet som omhandler elefanter, men avgrenset til de som kun omhandler elefanter. Disse bør være presenter på en slik måte at brukeren enkelt kan plukke ut de artiklene som er av interesse. Ideelt sett bør også brukeren selv kunne velge hvordan han vil rangere treffene. Det kan være verd å merke seg at det kan lønne seg å utvide søkeuttrykket med flere søketermer slik at man får færre treff og lettere kan finne det en søker. For å avgrense søket over kan man for eksempel søke etter afrikanske elefanter.

En journalist i Adresseavisen og en avisleser kan ha ulike motiv og forutsetninger til å finne fram en lagret artikkel, som er lagret i systemet. For en journalist er det viktig å kunne finne fram til tidligere artikler fordi journalistikk ikke bare handler om hva som skjer i verden akkurat i dag, eller hva som skjedde i går. For å kunne være kritisk må man ha bakgrunnsinformasjon. I avisene ser man ofte at i forbindelse med for eksempel en rettssak, at leserne får et sammendrag av hendelsene rundt ugjerningen. Vi ser også gjenbruk av artikler på sommerstid, hvis det er lite å skrive om. Agurknyheter er gjerne artikler som ikke er relatert til et bestemt tidsrom og kan likegodt være skrevet i går som for 4 år siden. Når en journalist skal finne igjen en artikkel eller bakgrunnsinformasjon

vedrørende en hendelse har han tilgang til hele systemet der artikler, bilder og lignende lagres i. Ønsker journalisten å finne alle nyhetsobjektene som tilhører i en artikkel, må han søke i flere databaser i tillegg, med mindre artikkelen er helt ny. Dette kompliserer søkeprosessen brukeren må for det første vite i hvilken database han skal lete for å finne det han ønsker, og videre må han også vite hvilket metadataformat som gjelder for denne databasen. Dette gjør også at det tar lang tid å lære seg systemet. Dette gjør at man må ha erfarne brukere for å få en stor gjenfinningsprosent.

En avisleser kan ha helt andre motiv enn en journalist for å lete etter en artikkel. Generelt kan man se en rask økning i nettavislesing i befolkningen. Nyheter er ferskvare og konkurransen er stor. Videre søkes det etter artikler og folk er nok i mye større grad aktive i utvelgelsesprosessen enn før. De søker på samme måten som journalistene fram til bakgrunnsinformasjon om dagsaktuelle saker, eller saker som de har en spesiell interesse for. Vanlige mennesker har også ofte et ønske å finne igjen artikler eller reportasjer som de selv har vært med på. Artikler som er lagt ut på internett er søkbare, men med et helt annen tilgang og søkegrensesnitt enn journalistene har. Leserne av avisen har også dårligere forutsetninger til å finne det de leter etter både fordi de har tilgang til mindre materiale, og fordi de kanskje ikke kjenner systemet. Ønsker leseren tilgang på en artikkel trykt i avisen kan han henvende seg dit og få utskrift av det han ønsker. Eller gå til enkelte biblioteket som har avisen på mikrofilm, eller i sitt arkiv.

Når en bruker skal søke etter informasjon må han bestemme søkekilde og formulere en korrekt spørring til denne søkekilden. Videre må det som returneres fra søket vurderes i forhold til relevans. Recall og Presisjon er to begreper for å vurdere resultatet av et informasjonssøk[7]. Recall er den prosentandelen av de relevante dokumenter som er framfunnet innenfor det aktuelle tema. Det er et mål på effektiviteten til søket. Man ønsker alltid høyest mulig recall. Presisjon er den prosentandelen av dokumentmassen fra et søk som egentlig er relevant for brukeren. Disse begrepene blir ytterligere beskrevet i kapittel 2.

1.3 Problemstilling

Adresseavisen har mange databaser og ulike metadataformat, som skaper problemer og gjør det vanskelig å vite hvor og hvordan man skal søke. Et annet problem er at samme metadata lagres flere steder. Dobbeltlagring tar både unødig plass og gjør vedlikehold og oppdatering av innholdet vanskeligere.

Jeg ønsker i denne oppgaven å se på rammeverk som støtter nyhetsformidling og tar utgangspunkt i NewsML og det ad-hoc systemet Adresseavisen bruker for sin lagring og gjenfinning av artikler, bilder og lignende.

For å forstå og vurdere hvordan systemet virker for en potensiell bruker har jeg valgt følgende problemstilling;

Hvordan kan NewsML fungere som rammeverk for metadata i Adresseavisen, og hvilke fordeler og ulemper gir dette for gjenfinning av data?

1.4 Arbeidsoppgaver

For å belyse og svare på problemstillingen er det nødvendig å beskrive systemet NewsML, se på brukesområder og funksjoner. For å forstå den praktiske anvendelsen har jeg oppsøkt Adresseavisen for å danne meg et bilde av hvordan deres system fungerer. Jeg har beskrevet Adresseavisens nåværende metadataformat. I dette arbeidet har jeg i tillegg hatt stor nytte av Oppedal sin hovedoppgave[1]. For å tenke NewsML omsatt til praktisk handling har jeg laget en applikasjon der jeg bruker nettsiden til Norges Astma og Allergiforbund som eksempel, og laget en applikasjon for å kunne legge inn og søke etter artikler i deres system.

På bakgrunn av disse eksemplene har jeg gjort en vurdering av fordeler og ulemper med NewsML.

For å svare på problemstillingen er følgende oppgaver utført:

1. Beskrive Adresseavisens nåværende systemer for informasjons- lagring og gjenfinning.
2. Case: Implementering av NewsML
3. Vurdere nytten Adresseavisen vil kunne ha ved å gå over til bruk av NewsML i forhold til dagens system.

1.5 Avgrensinger

Det er mange aspekter en må ta hensyn til når en skal lage et system som støtter gjenfinning og gjenbruk. Denne oppgaven tar sikte på å beskrive hvordan NewsML kan brukes til å lagre nyheter som objekter og hvordan en på denne måten kan får et uniformt metadatasett for alle nyhetsobjektene. Den tar ikke sikte på å lage et komplett implementert system der en kan teste både brukervennlighet, gjenfinnings- hastighet og prosent. Prototypen inneholder de fleste funksjonene ferdig implementert. Det er lagt mindre vekt på å lage et godt grensesnitt en det som ville vært nødvendig dersom systemet reelt skulle anvendes. Fokuset ved implementeringen har vært på å avdekke potensielle problemer både teknisk og på den brukerrelaterte siden.

Kapittel 2 Informasjonsforvaltning

”Only librarians like to search, everyone else likes to find”[8]

Gjenfinning av artikler og dokumenter er en voksende utfordring ettersom det er stadig flere som bruker datamaskiner og nettaviser aktivt. Dokumentmengden øker drastisk og konkurransen mellom avisene øker. Det kan synes som et konkurransefortrinn at avisen kan tilby best mulig tjenester, slik at de kan greie å holde på sine brukere. Denne oppgavens mål er å se om NewsML kan brukes for å forenkle gjenfinningen av deler eller hele artikler i Adresseavisen.

I dette kapittelet vil det presenteres to brukerscenario, og det vil videre inneholde utdrag av sentrale emner som er avgjørende for informasjonsgjenfinning: informasjonssøking, vurdering av resultat, metadata, og noe om implementering av metadata. Til slutt oppsummeres fordeler og ulemper sett fra brukernes side.

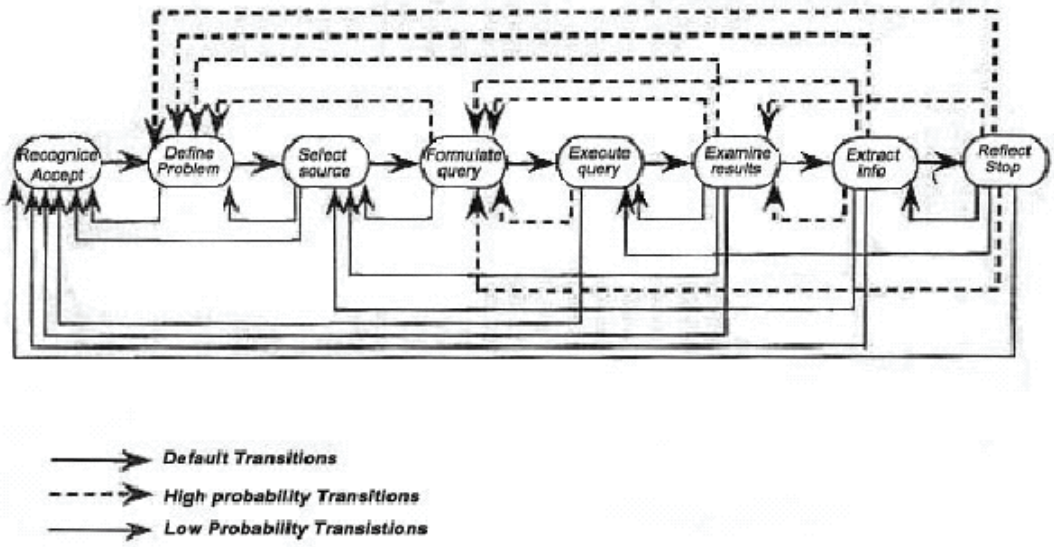
2.1 Brukerscenario

Scenario 1. Jens er journalist i Adresseavisen og skal dekke saken i Trondheim Tingrett. En 32-årig estlender ble holdt fanget og torturert i en hel uke etter en mislykket narkotikahandel. Før journalisten går til rettssalen vil han oppdatere seg på saken og søker derfor i stoffarkivet i SIFT-databasen for å finne artikler fra hendelsen. Når han har funnet artiklene han er ute etter, søker han i filmarkivet etter noen passende bilder fra saken. For å sjekke om det har vært noe interessant i nettavisen utfører han også et søk i internettdatabasen. Nå har Jens funnet den informasjonen han trenger for å skrive denne saken. Mot slutten av rettssaken skriver han sin artikkel, og sender den inn i systemet for videre behandling.

Scenario 2. Nina går på videregående og skal skrive oppgave om Irak-krigen. Siden denne krigen er for ny til at det står noe om det i lærebøkene hennes må hun finne andre informasjonskilder. Hun leser Adresseavisen daglig, og vet de hadde en god dekning av saken. Hun bestemt seg for å bruke deres nettsider som kilde. Hun går inn på www.adressa.no og skriver inn ”krigen i Irak” i søkevinduet på forsiden. Hun får over 500 treff. Nina har jobbet mye med søk på nett før og skjønner at det er nytteløst å lete på måfå gjennom trefflisten. Hun legger derfor til flere søkeord for å spesifiserer søket sitt ytterligere før hun til slutt finner to artikler som hun anser som relevante for sin oppgave.

2.2 Søkeprosess

Vi ser at både Jens og Nina utfører en søkeprosess, resultatet kan muligens bli noe ulikt siden Jens har tilgang til alle artikler som er skrevet i Adresseavisen, mens Nina må nøye seg med det som er tilgjenglig på nett. For å beskrive hvordan en slik prosess kan foregå har jeg valgt å ta utgangspunkt Marchionini sin[9] beskrivelse av informasjonssøkeprosessen. Han deler selve denne prosessen inn i åtte delprosesser:



Figur 1 Information Seeking Process[9].

1. Gjenkjenne og godta informasjonsproblemet: Det er ulike grunner til at en ønsker å finne en bestemt type informasjon. Man kan enten ha en indre motivasjon, et genuint ønske om å vite mer om emnet eller man kan ønske å finne informasjonen fordi en trenger den til å løse en oppgave.
2. Definere og forstå problemet: For å kunne søke etter informasjon må man ikke bare ha en idé om hva man ønsker å finne ut av. Man må definere problemet så spesifikt at det er mulig å lete etter informasjon om det. For å kunne greie dette på en bra måte trenger man ha en viss kunnskap om domenet man ønsker å søke på.
3. Velge søkekilde: Hvor en bruker søker kommer mye av personens tidligere erfaringer og kunnskapsnivå innen domenet, personens generelle kunnskapsnivå, og hva personen ønsker å finne. Kunnskapen om domenet er viktig i valg av søkekile; har du høyt kunnskapsnivå om domene har du også et høyt kunnskapsnivå om søkesystemet innenfor domenet. Personens generelle kunnskapsnivå har mye å si for personens generelle evne til å lete fram og vurdere interessant informasjon.
4. Formulere spørring: Formuleringen av søkestreng krever en del av brukeren. Søkeren må ha et ordforråd innenfor søkedomenet for å kunne greie å formulere en god søkestreng.
5. Utføre søket: Dette er fysisk å utføre søket.
6. Undersøke resultatene: Et søk resulterer i ulike resultater avhengig av hvilken søkestreng vi bruker, og hvilken kilde man søker i.
7. Skille ut interessant informasjon. Å skille ut den interessante og relevante informasjonen fra massen er en viktig oppgave i forbindelse med et informasjonssøk. En artikkel kan være interessant uten at den er relevant i forhold til problemstillingen

vi tar utgangspunkt i. For å skille ut relevant informasjon bruker man evner som; lesing, tekstsøking, klassifisering, kopiering og lagring av informasjon.
8. Reflektere over resultatet; stoppe eller gjenta søkeprosessen. Et informasjonssøk er sjelden over etter første forsøk. Ofte kan resultatet av et informasjonssøk rettlede brukeren til videre søking.

Her ser vi at Nina og Jens har både sammenfallende behov og ulike forutsetninger for å løse oppgaven. – De prøver begge å finne den informasjonen de søker fordi de har en spesifikk oppgave som skal løses. Dette krever at resultatet må være mer spesifikt enn om man bare ønsker noe som er interessant. De har begge et klart bilde av hva de ønsker å finne informasjon om, men kunnskapen om domenet de søker innen er forskjellig. Jens har en enklere oppgave med å formulere rett spørring, siden han har mye erfaring og kunnskap om søkedomenet sin oppbygning. Resultatet fra de ulike spørringene vil også bli noe forskjellig.

På avisens internettsider har brukere ingen mulighet rangere resultatet fra søk. Man må rett og slett begynne på toppen og bla nedover. Når man søker internt i avisen bruker man metadatasøk mot databasen, noe som gir mer presisjon i søket. Nina søker etter et tema der det finnes mye skrevet materiale. Hun får veldig mange treff som indikerer at hun må spesifisere søkestrengen bedre, altså må hun legge til flere ord i søkestrengen.

Jens skriver om en mindre sak, men som likevel fikk en del mediedekning. Han kan likevel oppleve å ikke få noen treff i det hele tatt, noe som gjør at han enten bør utvide søkestrengen noe, eller søke på annen type metadata hvis han bruker et slikt type søk. Problemet kan være at man ikke finner den informasjonen man leter etter ved første søk og må gjenta hele søkeprosedyren.

Hvis man holder fast ved den opprinnelige problemstillingen, kan man enten endre spørringen, eller finne en annen søkekilde, noe som eventuelt er mer relevant for Nina enn Jens.

2.3 Vurdering av søkeresultat

Det finnes ulike måter en kan måle effektiviteten til et informasjonsøk. Cleverdon [10] har identifisert 6 kriterier for evaluering av et informasjonsgjenfinningsystem:

1. *Recall*; systemet evne til å finne igjen alle relevante dokumenter.
2. *Precision*; systemets evne til å gjenfinne kun de dokumentene som er relevante.
3. *Time lag*; hvor lag tid det fra brukeren sender inn søkeforespørselen til databasen returnerer svaret.
4. *Effort*; hvor mye krever det av brukeren intellektuelt og fysisk for å innhente svar.
5. *Form of presentation*; hvordan blir søkeresultatet presentert. Dette vil påvirke brukernes evne til å nyttiggjøre innhentede dokumenter.
6. *Coverage of the collection*; i hvilken grad systemet inneholder relevant materiale.

For å kunne svare på de første spørsmålene må en først avgjøre hva som er relevant. Til syvende og sist er det søkeren som må avgjøre treffene er relevant i forhold til sin problemstillingen. Hvis man søker i en fagdatabase, bokdatabase eller i en avisdatabase

som tilfelle med Nina og Jens, kan den være satt opp med et felt for "hovedemneord" for artikkelen/boken som er indeksert. Hvis man da søker kun på hovedemneord vil resultatet mest sannsynlig bli bedre enn om vi skulle søkt på alle emneord.

2.3.1 Recall

Recall er den prosentandelen av de relevante dokumenter som er framfunnet innenfor det aktuelle tema. Og kan regnes ut slik [9]:

$$\text{recall} = \frac{\text{antall relevante dokumenter funnet}}{\text{alle relevante dokumenter for spørringer}} \times 100$$

Recall blir et mål på effektiviteten til søket. Noen ganger kan det være vanskelig å regne eksakte tall, da man ikke vet hvor mange dokumenter det er innenfor hvert tema. Man ønsker alltid høyest mulig recall, men i tilfelle med Nina, der det er mange artikler som er relevante for det hun skriver om, er gjenfinningsgraden ikke avgjørende for hennes vurdering av søket, da hun likevel finner nok informasjon.

2.3.2 Presisjon

Presisjon er den prosentandelen av dokumentmassen fra et søk som egentlig er relevant for brukeren. Dette kan regnes ut slik:

$$\text{presisjon} = \frac{\text{antall virkelig interessante dokumenter}}{\text{antall dokumenter funnet}} \times 100$$

Recall med høy presisjon er ofte ønskeresultatet når man utfører et søk[11]. Ved søk på internett ser man at dette ikke alltid er tilfelle. Som i Ninas tilfelle er problemet at man finner mye som ikke er interessant uansett hva slags type spørring man har. Mye av grunnen til dette er at de fleste artikler på internett ikke blir tilordnet metadata, og derfor kun blir indeksert av søkemotorene på grunnlag av ordene i teksten[12]. I Adresseavisen er det i utgangspunktet et høyt fokus på gjenfinning, ved at alle artikler blir tilordnet metadata.

2.3.3 Kombinasjon av recall og presisjon

Det er laget ulike metoder for å kombinere recall og presisjon:[13]

- The harmonic mean F of recall and precision:

$$F(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}}$$

Der r(j) er recall for dokument j og P(j) er presisjonen for dokument j, begge oppgitt som et tall mellom 0 og 1. Resultatet F(j) vil da bli et tall mellom 0 og 1, der resultatet vil bli

0 om ingen relevante dokumenter er funnet, og 1 når alle gjenfunnet dokumenter er funnet. Dette gjør at F blir høy kun når både recall og presisjonen i søket er høyt. Her blir altså recall og presisjon like mye vektlagt.

- The E measure:

$$E(j) = 1 - \frac{1 + b^2}{\frac{b^2}{r(j)} + \frac{1}{P(j)}}$$

Der r(j) er recall for dokument j og P(j) er presisjonen for dokument j, begge oppgitt som et tall mellom 0 og 1. b er her en brukerdefinert variabel som spesifiserer hvor stor vekt brukeren ønsker å ha på henholdsvis recall eller presisjon. B-verdier over 1 tilsier at brukeren er mest interessert i presisjon, mens verdier mindre enn 1 har fokus på recall. Ved B-verdi = 1 vil The harmonic mean F of recall and precision og The E measure gi samme resultat i en gitt søking.

2.3.4 Time lag

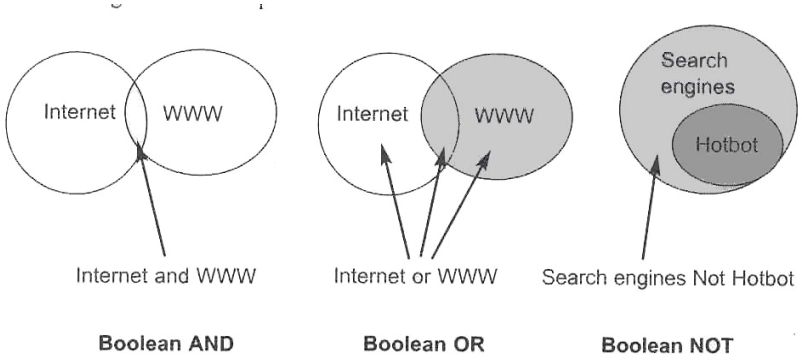
Dette definerte Cleverdon til å være hvor lag tid det fra brukeren sender inn søkeforespørselen til databasen returnerer svaret. I dag de aller fleste databaser så raske at vente tiden her blir ubetydelig. Hvis man utvider time lag begrepet til å omfatte hele søkeprosessen, inkludert prosessen med å velge søkekilde, formulere spørring og evaluere treffene. Hvor lang tid hele søkeprosessen tar er et mer relevant spørsmål for dagens søkesystemer. Roy Tennant sier at brukere gjerne vil ha et sted å søke, istedenfor å flere[8]. Dette har helt klart en praktisk side, men med et godt søkesystem er det også en god tidsinvestering.

2.3.5 Effort

Hvor mye krever det av brukeren intellektuelt og fysisk for å innhente svar. Ideelt skal et søkesystem ha veldig lav inngangsterskel og gir kun de svarene man er ute etter uten at vi egentlig trenger å uttrykke oss klart. Datamaskiner er ikke tankelesere, og de fleste systemer krever at man kan noe om hvordan man formulerer en spørring. Med et godt system bør brukeren finne det han søker etter uten å måtte gå igjennom Marchioninis søkeprosess mange ganger.

2.3.6 Form of presentation

Hvordan blir søkeresultatet presentert? Dette vil påvirke brukernes evne til å nyttiggjøre innhentede dokumenter. Har man mange svar på en spørring er det en fordel om brukeren har mulighet til å sortere svarlisten etter for eksempel dato. Det kan være en fordel å få mye informasjon om hvordan de ulike dokumentene er rangert i forhold til selve søketermen. En slik rangering kan for eksempel gjøres ved en boolsk sammenligning der man skiller dokumentene ved hjelp av en logisk funksjon som ”AND”, ”OR” og ”NOT”, mellom søketermene i spørringen[14].



Figur 2 Boolske sammenligningsoperatorer [10].

Vektorbasert sammenligning er en annen metode for vekting av spørringer. Den baserer seg på full kjennskap til alle dokument som finnes i databasen og dermed også den totale termmengden som finnes i den. Hvert dokument vil representeres av en vektor som beskriver dokumentets relevans i forhold til hver enkelt term i databasen – beregnet ved hjelp av en formel: DOC j = (termj1, termj2, termj3…)
Det enkleste er å bruke binære vekter for å regne ut dokumentvektoren, da settets t=0 hvis termen ikke finnes i dokumentet og t=1 hvis termen finnes[10].

2.3.7 Coverage of the collection

Dette begrepet beskriver i hvilken grad systemet inneholder relevant materiale. Dette er alltid et relevant spørsmål uansett hvem som søker. For å få et svar på en spørring krever det selvfølgelig at databasen innholder artikler som er relevant for spørringen. Har kan det hjelpe med en god beskrivelse av innholdet til databasen. Dette blir ikke direkte relevant for oppgaven.

2.4 Metadata

Metadata er selve kjernen når man snakker om gjenfinning i strukturerte data som lagringssystemene i Adresseavisen er et eksempel på. Metadata betyr ”data om data”, eller sagt med andre ord: informasjon som beskriver et annet sett med data[15]. En artikkel trenger ikke nødvendigvis å innholde alle relevante søketermer i selve teksten. Metadata vil kunne ufylle konseptuelle mangler i søketermmengden som er relevante for gjenfinning av artikkelen.

Metadata begrepet kommer fra museumsansatte, bibliotekarer og arkivarer og refererte til innholdet i kataloger eller indekser. Denne informasjonen ble brukt til å organisere, beskrive og på andre måter forbedre tilgangen til et informasjonsobjekt[16]. Etter hvert har begrepet blitt utvidet fra å gjelde tradisjonelle dokumenter som finnes i biblioteket, eller objekter på et museum, til å også gjelde for andre objekter, for eksempel personer eller gjenstander, eller elektroniske dokumenter. Aalberg og Hegna’s definisjon av metadata dekker også disse objektene[17].

”Metadata er en formell beskrivelse av indre og ytre karakteristika hva tradisjonelle og digitale dokumenter og objekter som understøtter formidlingen av dem (dokumenter og objekter) til personer.”

Det snakkes ofte om metadataformat i denne sammenheng. Et metadataformat er et regelsett for å strukturere metadata, det vil si at formatet forteller hvilken informasjon som skal være med og hvordan denne skal kodes. Metadataformatet som brukes i Adresseavisen blir beskrevet nærmere i kapittel 3.

Når artikler lagres med metadata vil de bli lettere søkbare, som når Jens skal søke etter artikler som omhandler hendelsen som skal opp i Tingretten. Han kan søke på ulike metadatafelt, som for eksempel hvilken sak det gjelder, spesifikk dato, journalistnavn, sted med mer. Han kan også kombinere flere felt i samme søk, for eksempel søke på journalistens navn og dato samtidig, og dermed begrenset antall svar fra spørringen betraktelig. Fordelen med dette er at søket får både en høyere recall og presisjon i forhold til om man skulle søkt gjennom teksten slik søkemotorer på internett gjør.

2.5 Metadata i Adresseavisen

Teksten til en artikkel i Adresseavisen som brukes både i papirutgaven og i internettutgaven, vil bli lagret i tre forskjellige databaser. Som vi vil se i kapittel 3 benytter Adresseavisen ulike metadataformat i disse databasene, noe som gjør at metadatapostene blir unike i hver database. Vi vil også se at det legges til metadata i flere ledd etter at journalisten har sendt fra seg artikkelen. Sannsynligvis er dette for å gjøre arbeidet til journalisten enklere, men det krever arbeidsinnsats fra arkivpersonalet som ikke har jobbet med artikkelen.

At ulike versjoner av en artikkel lagres i forskjellige databaser fører også til vanskeligheter for brukeren som skal finne informasjon internt på Adresseavisen. Det er vanskelig å skaffe oversikt over de ulike metadataformatene og brukeren må vite hvor det er hensiktsmessig å søke. Den eksterne brukeren har kun tilgang til artikler på internett, og her finnes det pr i dag ikke støtte for metadatasøk.

Dobbeltlagring har også flere ulemper; det tar stor lagringsplass og gjør at databasene går saktere enn de ellers ville ha gjort. Det gjør også at man kan få flere treff enn nødvendig fra et søk ved at systemet henter samme artikkel flere steder, som igjen gjør vanskeliggjør prosessen i å plukke ut det som er relevant. Oppdatering og vedlikehold av dataen krever mer arbeid siden den samme informasjonen finnes flere steder. Det er fort gjort å glemme å oppdatere alle instansene av den samme informasjonen. Akkurat dette er ikke et stort problem i en papiravis der artikler lagres og ikke oppdateres, men artikler i nettaviser oppdateres gjerne etter hvert som journalistene selv får mer informasjon.

2.6 Oppsummering

Når en bruker søker etter informasjon må han først finne ut hva han søker etter og deretter finne rett søkekilde. Videre må han formulere en spørring som passer søkekilden. Når han har utført søket, må han vurdere resultatene og eventuelt utføre et nytt søk. For

brukeren er det viktig at han finner den informasjonen han er ute etter. For å understøtte dette må søkesystemet være bygget opp slik at det både er enkelt for brukeren å finne ut hvor og hvordan han skal søke. Samtidig må søkesystemet være strukturert slik at det er stor sjanse for at søkeresultatet inneholder alle de dokumentene som er relevante i forhold til søketermene samtidig som irrelevante dokumenter utelates. Dette gjøres mest effektivt ved hjelp av data som beskriver innholdet til dokumentene i databasen, eller metadata.

Slik systemet til Adresseavisen er i dag, er det relativt enkelt for en utenforstående person å søke i artikler som ligger i nettutgaven de andre artiklene har han ikke direkte tilgang til. Personen har ingen mulighet til å velge hvilket felt han skal søke på. Resultatet kan være varierende, uten sorteringmuligheter og vanskelig å etterprøve for utenforstående. Men på et enkelt søkt har jeg prøvd å søke etter artikkelen med tittel « Livsfarlig bruklatring» som stod i dagens avis, både full tittel og bare bruklatring, og 0 treff.

En person som jobber i Adresseavisen har også mulighet til å søke i nettavisen, men for å få et mer presist resultat kan det være en fordel å søke internt i systemet. Han må da vite hvilken database han skal søke i for å finne det han leter etter, han må også kjenne til metadataformatet til den aktuelle databasen. En erfaren journalist i Adresseavisen vil derfor søke mer effektiv, enn en som ikke kjenner systemet og de ulike metadataformatene like godt.

Kapittel 3 Adresseavisen

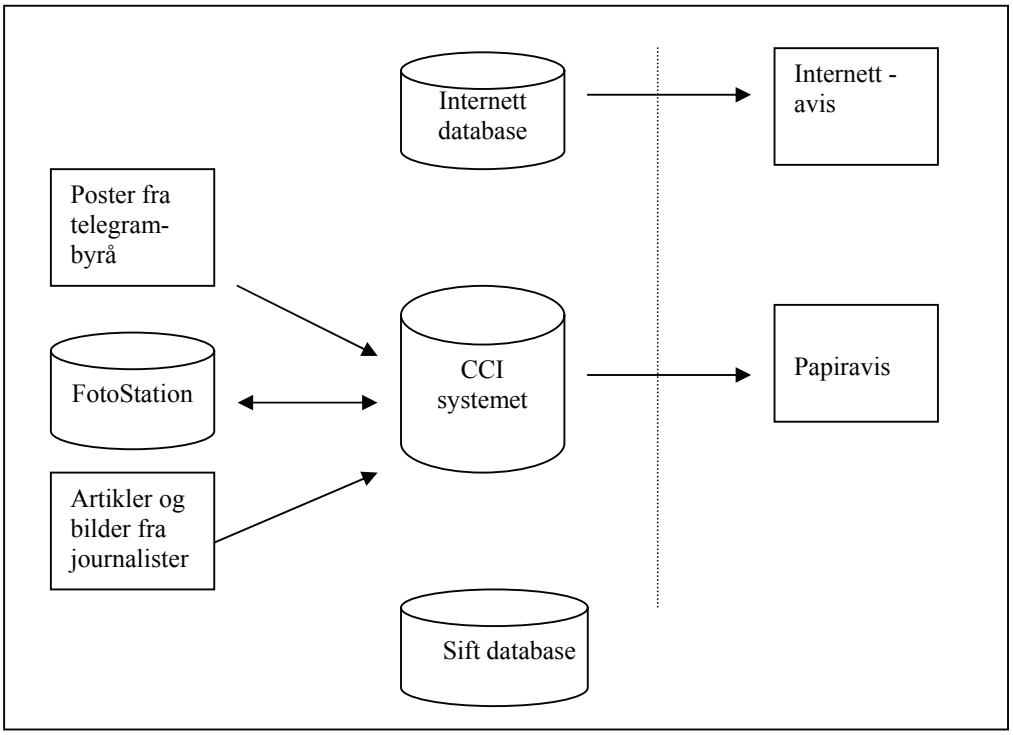
Et av målene med denne oppgaven er å se på om Adresseavisen sitt system kunne vært tjent med å innføre NewsML. Som bakgrunn for dette har jeg først beskrevet nåværende situasjon.

Innledningsvis beskrives de ulike databasene Adresseavisen har, med hovedvekt på SIFT databasen og deretter de ulike metadataformatene til de enkelte databasene. Videre beskrives hvordan indekseringen foregår og hvordan søkemulighetene er i systemet. Avslutningsvis i dette kapitlet gis en vurdering av Adresseavisens nåværende system.

For å fremskaffe kunnskap om nåværende situasjon har jeg også besøkt Adresseavisen (januar 2003). Jeg har også hatt stor utbytte av Oppedals hovedfagsoppgave, hvor hun har beskrevet avisens bruk av metadata[1]. Deler av dette kapittelet bygger på hennes oppgave.

3.1 Databasene i Adresseavisens nettverk

I dag anvendes fem ulike databaser. Illustrasjonen under viser hvilke databaser som finnes og hvordan de kommuniserer med hverandre.

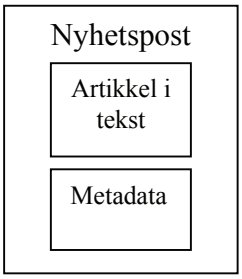


Figur 3 Adresseavisens databaser[18].

3.1.1 CCI systemet

CCI står for Comment Computer Interface. Dette er produksjonssystemet som brukes for å produsere avisen. CCI blir altså kjernen for alt det produktive arbeidet. I CCI-systemet blir også den ferdige papirutgaven av avisen produsert.

Som vi ser fra figur 3 mottar altså CCI-systemet informasjon fra NTB og journalistene, og har en interaksjon med FotoStation, Sift og internettbasen. Disse beskriver jeg nærmere under. Alt av bilder og artikler fra redaksjonen går inn her. I tillegg kommer reklame, annonsesider med mer, fra annonse – og markedsføringsavdelingen. Systemet består av en rekke av det Adresseavisen kaller ”basket’er”, som er mapper for hvert spesielt tema. Disse innholder de ulike artiklene, notatene, reklamene og lignende. En journalist som sender inn en artikkel kan velge hvilken basket den aktuelle artikkelen skal gå inn i, eller man kan sender den til en generell nyhetsbasket. En i redaksjon kan gå inn i denne nyhetsbasketen og hente ut en kopi av artikkelen -, men originalen av artikkelen blir alltid liggende uforandret, hvis ikke annet blir spesifisert. Brukte poster i basketen, lagres i en egen basket for den dagen den ble brukt, før den etter 7 dager blir lagret i Sift. Poster som ikke har vært brukt slettes etter 7 dager. I nyhetsbasketene blir artikler og bilder lagret separat fra de tilhørende metadata.



Figur 4 En nyhetspost i Adresseavisen[18].

3.1.2 Poster fra telegrambyrå

Adresseavisen abonnerer på nyhetsmeldinger fra Norsk Telegrambyrå(NTB). NTB sender ut ca 250 nyhetsmeldinger på en vanlig dag. Disse artiklene er å finne i denne basen. Bilder som sendes ut fra NTB finnes også i FotoStation.

3.1.3 FotoStation

FotoStation er et system som består av en database med en programvare som er beregnet for produksjonsmiljøer der store mengder digitale bilder blir behandlet. Benyttes av de fleste mediebedrifter i Norge og innen en rekke andre bransjer som bl.a. ulike høgskoler, enkelte fylkeskommuner og forsvarets forsknings institutt[19].

Her ligger alle bilder, illustrasjoner og lignende som er skannet inn for å brukes på trykk i avisen. Bildenes opprinnelige format kan variere, og er uten reel betydning da dette omarbeides ved senere bruk. Bilder fra byrå (for eksempel fra NTB) kommer også inn

her. Bildene lagres i forskjellige virtuelle mapper avhengig av deres opprinnelse. Sammen med selve bilde lagres også metadata, som vist i understående tabell. Metadatafeltene i indekseringsformatet kan enkelt fjernes eller det kan legges til nye, etter behov. Overskriftene på gruppen av metadatafeltene kan også endres, men dette er noe mer komplisert. Dette er gjort for at den enkelte bedrift skal kunne optimalisere til sitt behov. Det er enkelt å koble opp mot et intranett eller internett som gjør søking i databasen mer tilgjenglig.

Hovedgruppe	Undergruppe	Metadata	Bruk av metadatafeltet.
Bildetekst	Ingen definerte	Gruppe (objekt navn)	Frie emnegrupper
		Sak (overskrift)	Tema/tittel til saken bildet/ illustrasjonen er brukt i forbindelse med.
		Fotograf	Angir fotograf/tegner.
		Motiv	Beskrivelse av innhold
		Instruksjoner	Dette feltet brukes til å skrive in ny ”brukt dato”, ”produkt”, og ”side” for hver gang bildet er brukt på nytt.
Kategorier og nøkkelord	Ingen definerte	Kreditt	Brukes ikke av adresseavisen.
		Hva er skannet	Angir typen objekt.
		Journalist	Angir opphavsmann.
		Opprinnelige overførings referanse.	Brukes ikke av adresseavisen.
		Returnert til (opphavsrett)	Angir ev. navn til opphavsmann hvis objektet er returnert til ham/henne.
		Nøkkel ord (gruppe)	Angir emne ord fra kontrollert vokubalar.
	Kategorier	Produkt (kategori)	Angir en forkortelse av emnegruppa innenfor kategorien. Max tre bokstaver. Kontrollert vokabular.
		Produkt tillegg.	Hvilket bilag artikkelen skal brukes i. Her har vi Adr’ut (utmagasinet), Ard’sport (lille-sportsavis), adr’uke (uke Adressa), Adr’Nyhet, Adr’kultur. ADR oppgis dersom det ikke står under noen av de andre produktene.
Data og status	Dato og tid	Fotodato	Angir dato for ”produksjon” av bildet, illustrasjonen osv.
		Tid opprettet.	Brukes ikke av adresseavisen.
		Utgivelses dato	Angir dato objektet står på trykk
		Utgivelses tid	Brukes ikke av adresseavisen.
	Status	Edit status	Ingen av disse feltene brukes av adresseavisen.
		Priorty	
		Objekt cycle	
	Plassering	Brukt dato	Identisk med utgivelses dato over.
		Side (provins/stat)	Angir siden objektet er brukt på.
		Landskode	Brukes ikke av adresseavisen.
		Land	Brukes ikke av adresseavisen.
		Original referanse	Brukes ikke av adresseavisen.
	Diverse	Merknad	Andre opplysninger som kan være av interesse, slik som for eksempel at filmen mangler.
		Film-nr + motiv-nr.	Angir et filmnummer og hvilken bilde nr det aktuelle har på negativstripen. Referanse til film-id i SIFT filmarkivet.
Egendefinerte felter	Ingen definerte	Egendefinert felt 1	Brukes ikke av adresseavisen. Kan multipliseres.
Nøkkelord		Nøkkelord	Brukes ikke av adresseavisen.
		Gruppe Nøkkelord	Brukes ikke av adresseavisen.

Tabell nr. 1 Metadataskjema for indeksering av digitale bilder/dokumenter i FotoStation [20].

Selv om enkelte av feltene enkelt kunne vært fjernet, ser vi at Adresseavisen ikke har gjort dette, selv om de ikke bruker feltene. En del felter har også et annet navn i FotoStation enn i SIFT, men avisen har ikke endret navnene på feltet. En ser også at hvis

et bilde brukes flere ganger, vil det beholde sin opprinnelige metadata, pluss et lite tillegg. Dette fungerer helt greit med situasjonsbilder til en sak, men f.eks. portretter kan være nyttig å benytte i ulike saker, og bør derfor ha mulighet til å knyttes opp mot ulike metadataskjema.

3.1.4 Artikler med bilder fra journalist

En journalist skriver inn en artikkel i Word, der bruker han en spesiell template, som har felt for å kunne legge inn ulike metadata, disse er ”Tittel”, ”Mellomtittel”, ”Undertittel”, ”Ingress”, ”Bildetittel”, ”Bildetekst”, ”Bildesignatur”, ”Vignett” og ”Undervignett”, i tillegg til selve artikkelen. Så sendes artikkelen inn, med ev. bilde. Dette gjøres i rtf, for å oppta så lite kapasitet som mulig. Journalistene har ingen mulighet til å legge til mer metadata hvis de ikke sitter på intranettet i en av avisens lokaler. Artikkelen som blir sendt går inn i en nyhetsbank i avisen, som ligger i CCI systemet. Der kan de velge å sende artikkelen til en spesiell nyhetsbank, som for eksempel sporten, Puls, eller Ukeadressa, eller en kan sende artikkelen til den generelle nyhetsbasen. Når journalisten har sendt fra seg artikkelen, har han eller hun ingen muligheter til å gjøre noe mer med den, så lenge han ikke får den returnert fra en redaktør for videre bearbeidelse. Her kan artikkelen tilføres flere metadata poster og sendes videre til de ulike redaksjonene. Redaksjonene står fritt til å endre artikkelen slik at den passer til det mediet de ønsker å bruke artikkelen i.

Alle artiklene som brukes, lagres med teksten på et sted, sammen med en referanse metadatasettet som ligger lagret et annet sted. Artikkelens bilder lagres på en tredje plass. Adresseavisen har planer om å endre sitt lagringssystem, slik at tekst og bilder blir lagret sammen. Dette vil bli enklere å administrere, og enklere å finne fram i.

Adresseavisen sier selv at det mest hensiktsmessige ville vært om en og samme redaktør hadde redigert samme artikkel for bruk i de ulike mediene. Men de organisatoriske endringene som skal til for at en få til dette, er vanskelige å få igjennom i organisasjonen.

3.1.5 Internettdatabasen

Denne databasen innholder artikler fra Internett i fulltekst og vedlegg som for eksempel bilder relatert til artiklene, med tilhørende metadata. Dette innholdet er produsert for Adresseavisens Internettutgave. Ved hjelp av stylesheets blir disse artiklene og tilhørende metadata gjort om til nettsider som er leselige ved hjelp av en nettleser. Det er denne databasen vanlige internettbrukere får tilgang til gjennom Adresseavisens hjemmesider.

Metadata navn	Forklaring
Artikkelref	Primærnøkkel for artikkelen i databasen. Brukes også i URL'en.
Tittel	Hovedtittel til artikkelen
Undertittel	Ev. undertittler som henger sammen med hovedtittelen.
Kategori	Hvilken kategori artikkelen går under. Hentes ikke fra et kontrollert vokabular, kategorier lages etter behov.
Temakategori	Utdyping av kategorien.
Publisert dato	Dato artikkelen lanseres i internettavisen.
Malfil	Referanse til mal som angir hvordan artikkelen skal se ut.
Kommentarer	Frie kommentarer.
Serredato	Her legges det inn den dato for når artikkelen vil bli sperret fram til.
Status	Angir om artikkelen er publisert eller ikke.
Byline	Angir hvem som har skrevet artikkelen.
Ingress	Dette er en kort innledning til artikkelen.
Forsidetekst	Angir hvilken tekst som skal stå (ev. har stått) på forsiden av internettavisen.
Brødtekst	Hele artikkelen i råtekst.
Forsidetittel	Tittelen artikkelen har fått på forsida av internettavisen.
Emneord	Frie emneord.
Vedlegg	Referanse til ev. vedlegg, som har egne metadata.

Tabell nr. 2 Metadataskjema for indeksering av artikler i internettdatabasen[20].

Som nevnt tidligere kan internettartiklene også ha vedlegg. Disse er hovedsakelig bilder, men kan også være for eksempel grafer eller illustrasjoner.

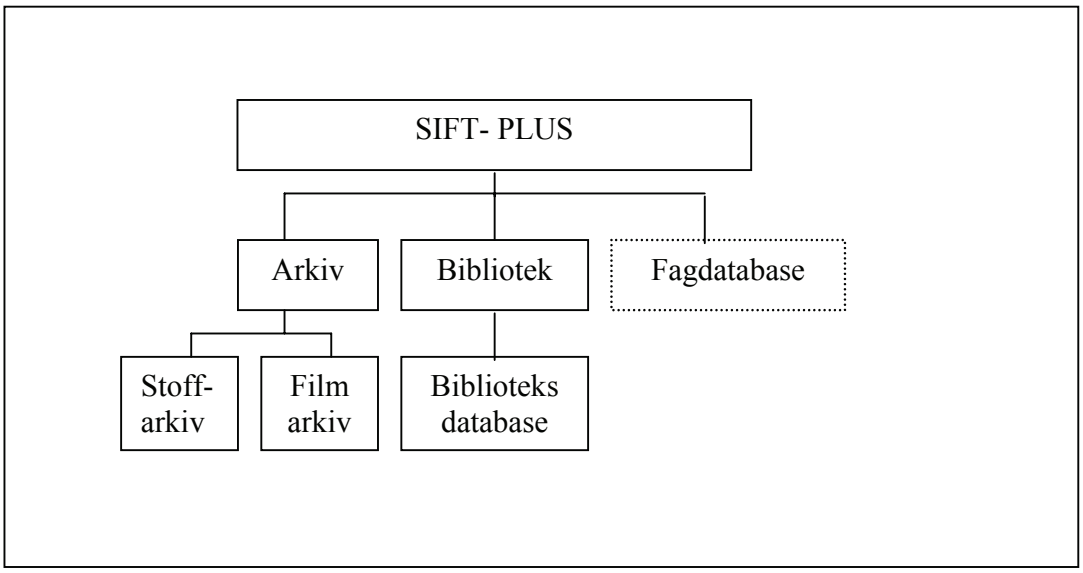
Metadatanavn	Forklaring
Filnavn	Angir filnavnet til vedlegget.
Vedleggsnummer	Angir en unik identifikator for bildet.
Vedleggstype	Angir hvilken type vedlegget er.
Bredde	Angir bredden til vedlegget.
Høyde	Angir høyden til vedlegget.
Byline	Angir opphavsmann til vedlegget.
Tekst	Angir ev. bildetekst og lignende som skal være med i internettartikkelen.

Tabell nr. 3 Metadataskjema for artikkelvedlegg i internettdatabasen [20].

I tillegg til å knytte vedlegg til en internettartikkel, kan en og legge til ulike linker som er relatert til saken. Dette gjøres gjennom standard URL, i artikkelen. Linker til relaterte saker internt, altså andre artikler i avisen genereres ut ifra emneord.

3.2 Siftdatabase

Denne databasen innholder artikler i fulltekst, metadataposter for fysiske filmmapper, illustrasjoner og bøker. SIFT står for "Searching in free text", og ble utviklet av Statens datasentral tidlig på 80 tallet. Det er et datasystem utviklet for å støtte gjenfinning av informasjon i fritekst. SIFT pluss som Adresseavisen bruker – innehar funksjoner som søking, lagring og registrering. Systemet håndterer rene tekstlige dataobjekter, og strukturerte og/eller formaterte data. Systemet finnes i både kommando og web grensesnitt utgave. Adresseavisen bruker begge grensesnittene[1].



Figur 5 SIFT – PLUS med tilhørende underdatabaser [1].

Som det fremgår av figuren er dette et omfattende arkivsystem. Adresseavisen har alle disse modulene med unntak av fagdatabasen.

3.2.1 Stoffarkivet

I dette arkivet blir alt tekstlig materiale som har stått på trykk i Adresseavisen registrert og indeksert. Indekseringsformatet for dette arkivet er utarbeidet av Statens datasentral som en standard for artikkel indeksering i aviser. Dette er et generelt format som tar sikte på å dekke alle avisers behov. Dette arkivet er ikke spesielt tilpasset Adresseavisen. Basen ble tatt i bruk i 1982, men først i 1993 ble SIFT innført som database system. Alt stoff i basen som er lagt inn etter innføringen av SIFT ligger indeksert, mens tidligere materiale ligger uindeksert. SIFT databasen er organisert etter årstall, dvs. at du må åpne det eller de årstallene som er interessante for søk i flere databaser[1].

Metadata navn	Forklaring
Skjermes	Skal stoffet skjermes fra inn syn. Mulige verdier er JA eller NEI.
Prod –dato	Dato for produksjon. Genereres automatisk ved overføring til SIFT.
Navn	Navnet på artikkelen. Kan kun innholde tekst.
Ant –linjer	Antall linjer artikkelen består av. Genereres automatisk ved overføring til SIFT.
Produkt	Hvilket bilag artikkelen skal brukes i. Her har vi Adr'ut (utmagasinet), Ard'sport (lille-sportsavis), adr'uke (uke Adressa), Adr'Nyhet, Adr'kultur. ADR oppgis dersom det ikke står under noen av de andre produktene av Adresseavisen.
Illustrasjon	Hvilken type av illustrasjon er brukt. Ved foto skives FOTO, ellers skives det inn illustrasjon, tegning og lignende avhengig av hvilken type illustrasjon som er brukt.
Kilde	Angir kilden til illustrasjonen, dvs. hvilken organisasjon for eksempel "eget arkiv", "AP", "NTB". Angir også navnet på opphavperson. Dersom det er en illustrasjon som er returnert skrives en kun inn navnet på fotograf eller tegner.
Original	Dette feltet er ikke brukt av Adresseavisen.
Journalist	Navnet på journalisten som har skrevet artikkelen.
Emneord	Kontrollerte emneord. Adresseavisen har over 30 ulike emnegrupper totalt.
Merknad	Feltet brukes til å beskrive artikkelen ytterligere. For eksempel ved andre mulige søke termer, bedrive spesielle faktorer ved artikkelen osv.
Stikkord*	Stikkord som er veiledende for hva artikkelen handler om. Det er kun dette feltet som vises på trefflisten, ved søk i databasen. Disse er ikke søkbare ved metadatasøk i Adresseavisen.
Tekst	Artikkelen i fulltekst, med alle titler, ingresser osv.

Tabell nr. 4 Metadataskjema for indeksering av artikler [1].

Det er vært å merke seg at metadata feltene som journalistene fyller ut, og som følger med artikkelen, ikke er tatt med i dette metadataformatet. Tittelen legges som første linje i tekst feltet. Årsaken til hvorfor dette er gjort slik er ikke kjent.

3.2.2 Filmarkivet

Filmarkivet har to indekseringsformaterer. Et for fysisk film(negativer), og et for illustrasjoner i form av disas, tegninger, illustrasjoner, plakater med mer. Mange av postene innholder referanser til fysiske objekt, og da er posten i filmarkivet kun en metadatapost. Formatene som benyttes er ad-hoc formater, som er utviklet i samarbeid med arkivleder i Adresseavisen og en representant fra Statens datasentral[1].

* Ikke søkbart ved metadatasøk I Adresseavisen.

Metadata Navn	Forklaring
Jobb-id	Dato pluss en påbegynnende jobb id. Genereres automatisk.
Film-nr	Filmene lagres i filmmapper på tre filmstriper pr mappe. En kan ved behov lage flere filmapper. Dette metadatasettet angir hvilken mappe som er den aktuelle mappen for eksempel 1 av 2 eller 2 av 3.
Film-id	Dato pluss en femsifret indikator for eksempel 10082001/44983.
Film-dato	Dato for når filmen ble tatt/oppbrukt.
Prod-dato	Hvilken type film ble brukt. For eksempel farge pos, farge neg eller svart hvit.
Fotograf	Angir fotografens som har tatt bildene.
Journalist	Angir journalisten som har skrevet artikkelen som bildene er knyttet til. Feltet kan gjentas hvis flere journalister har skrevet artikkelen.
Gruppe	Angir gruppe fra et kontrollert vokabular. Kan gjentaes ved behov.
Sak	Saken filmen er brukt i forbindelse med. Saken må være konsistent gjennom hele filmmappen.
Merknad	Dersom fil mangler av en eller annen grunn, skrives det opp her.
Retur	Er filmen blitt returnert? Verdier ”Ja” og ”Nei”.
Retur-adr	Adressen som returnert film er sendt til. Feltet vil være blankt hvis ovenstående felt har verdien ”nei”.
Bilde-nr	Her skrives nr på negativstripen. Feltet gjentaes hver gang et nytt bilde blir brukt.
Brukt-dato	Dato bildene ble brukt i avisen. Feltet gjentaes hver gang et nytt bilde blir brukt.
Produkt	Hvilket produkt ble bildet brukt i. Feltet gjentaes hver gang et nytt bilde blir brukt.
Side	Sidenummeret bildet ble brukt. Feltet gjentaes hver gang et nytt bilde blir brukt.
Motiv	En beskrivelse av bildet. Med navn på personer, om bilde er et ”action” bilde, portrett og lignende. Har en begrensning på to linjer. Feltet gjentaes hver gang et nytt bilde blir brukt.

Tabell nr. 5 Metadataskjema for indeksering av fysisk film[1].

Vi ser at en får en lignende problemstilling her som vi gjorde med. De fem siste metadatapostene blir gjentatt ved gjenbruk, men en sier ingenting om det bildet er bruk i forhold til den saken som bildet opprinnelig ble lagret til.

I filmarkivet blir illustrasjoner registrert.

Metadata navn	Forklaring
Antall	Hvis en sak innholder flere illustrasjoner lagres disse samlet i databasen. Dette metadatafeltet forteller hvor mange illustrasjoner det dreier seg om.
Ill-id	Genereres automatisk.
Illistrasjon	Type illustrasjon. Eks. bilder, tegning etc.
Ill-type	Her legges format typen inn. Eks fargefilm, sort- hvit eller dias film
Prod-dato	Dato for når illustrasjonen ble produsert.
Opphav	Opphavet til illustrasjonen.
Sak	Saken illustrasjonen er brukt i sammenheng med.
Gruppe	Angir intern gruppe. Feltet er blankt hvis illustrasjonen er ekstern, dvs. at den kommer fra NTB, privat person og lignende.
Merknad	Spesielle merknader.
Retur	Skriver hvor illustrasjonen ev. er returnert, med tlf.nr og adresse.
Brukt-dato	Dato for når illustrasjonen stod i avisen. Gjentas hvis illustrasjonen brukes ved flere anledninger.
Produkt	Hvilket bilag illustrasjonen skal brukes i. Her har vi Adr'ut (utmagasinet), Ard'sport (lille-sportsavis), adr'uke (uke Adressa), Adr'Nyhet, Adr'kultur. ADR oppgis dersom det ikke står under noen av de andre produktene av Adresseavisen. Gjentas hvis illustrasjonen brukes ved flere anledninger.
Side	Hvilken side illustrasjonen stod i. Gjentas hvis illustrasjonen brukes ved flere anledninger.

Tabell nr. 6 Metadataskjema for indeksering av illustrasjoner[1].

Det er vert å merke seg at dette metadataformatet ikke sier noe om hva illustrasjonen er av/om, og støtter derfor dårlig gjenbruk. Blir en illustrasjon brukt ved to anledninger vil den få to registreringer i SIFT.

3.2.3 Bibliotek databasen

Denne databasen omfatter alt det som kan regnes som avisens interne bibliotek. Her er bøker, kart, og ulike dokumenter indeksert. Det fysiske innholdet i biblioteket er lokalisert rundt om på de ulike avdelingene i avisen, og på kontorer til de ansatte. Indekseringsformatet som er benyttet i denne databasen er utformet på grunnlag av NORMARC formatet[1].

3.3 Indekseringspraksis

Slik systemet fungerte da Oppedal skrev sin hovedoppgave i 2000, var det 3 instanser som indekserte stoffet som skulle inn i de ulike databasene. Dette var arkivpersonale, internettredaksjonen, og delvis journalistene som skrev sine artikler i CCI Word, hvor metadata som tittel, mellomtittel, undertittel, ingress, tekst, bildetittel, bildetekst, bildesignatur, vignett og undervignett ble lagt til dokumentene. Arkivpersonalet indekserte SIFT databasene, der alt av artikler, bilder, illustrasjoner og fysisk film som blir brukt i avisen er indeksert og lagret. I tillegg blir alle bildene, både digitale bilder og fysisk film skannet inn, indeksert og lagret i SIFT databasen. internettredaksjonen benytter sitt eget indekseringsformat, og indekserer og lagrer artikler som brukes på nett i internettdatabasen. Dette gjør at alle artikler som brukes i avisens internettutgave blir indeksert i to ulike formater, både av internettredaksjonen og av arkivet.

Når en artikkel lages i CCI Word og lagt til bildene og noe metadata, importeres den over til en mellomdatabase som lagrer artiklene, og igjen sender artikkelen videre. Alle artiklene som blir lagret her, importeres også til SIFT. Når artikkelen importeres til SIFT, blir hele artikkelen i råtekst overført. Metadata som journalisten påførte i Word blir lagt til metadata feltet ”stikkord”. Arkivet påfører manuelt metadata som emneord, fotograf, hvilken illustrasjon det er til saken, og merknader. Metadata som produksjonsdato, artikkelnavn, hvor artikkelen har stått, samt artikkelens størrelse blir automatisk generert ved import fra CCI systemet.

Artikkelen blir så overført til HTML databasen hvor alle artikler blir HTML tagget. Vedlegg til artikkelen, som regel et bilde, blir lagret som et eget objekt i databasen, med en referanse til artikkelen den er en del av. De artikler som blir valgt ut til internettutgaven av avisen blir så indeksert og lagret i internettdatabasen, bildet blir lagret som et eget objekt i databasen. Dersom en artikkelen blir slått opp på første siden av avisen, vil saken bli laget som to enheter i SIFT databasen, en for selve artikkelen og en for forsiden.

Dersom en artikkel står i både papir og internettutgaven av avisen, vil artikkelen være lagret i fire ulike databaser. Dette er HTML databasen, SIFT, CCI systemet og internettdatabasen.

Hvis vi tar opp igjen eksempelet med Jens i kapittel 2, kan vi se at etter at han er ferdig med å skrive sin artikkel, legger til metadata når han lagrer artikkelen i CCI-systemet. Artikkelen kommer så på trykk og blir lagt i SIFT-databasen. I denne prosessen legger arkivpersonalet til noe mer metadata enn det Jens skrev i utgangspunktet. Hvis internettredaksjonen velger å bruke artikkelen, legger de til noe mer metadata og lagrer den i internettdatabasen.

3.4 Søkemuligheter i Adresseavisens systemer

Siden artiklene som trykkes og de som brukes på internett legges i to forskjellige databaser, må du vite hvilken artikkel du er ute etter, og så søke i den respektive basen.

3.4.1 Søking i SIFT

SIFT databasen tilbyr ulike og avanserte søkemuligheter, som kan deles i fire grupper[1].

- SIFT vanlig søkespråk. Her kan både fritekstsøk, og metadatasøk benyttes. Denne typen søk er meget nyttige for alle som kjenner systemet, og vet hvilke poster som er interessante å søke på.
    - Metadatasøk. Det er mulig å definere alle elementene i de ulike metadataformatene som søkbare, dette kan gjøres ved behov. I søk på metadata kan en ta i bruk boolske operatorer, som ”og”, ”eller” og ”ikke”. En har også mulighet til å sammenligne metadata verdier, som største og minste. Slik søking forutsetter at du vet hvilke poster som er søkbare og har klart definert hva du søker etter.
    - Fritekstsøk. I et fritekstsøk kan du i tillegg til å søke i metadatafeltene også søke i råtekstfeltene. Du har ingen mulighet til å spesifisere hvor (i hvilket

felt) søketermen din skal forekomme. Ved denne typen søk vil gjenfinningsraten være mye større, men de kan være ”unøyaktige”, i forhold til at man kan få en del treff som ikke er interessante. Dette kan være nyttig hvis en ønsker å søke på noe som det finnes lite stoff om i databasen.

- Søkeskjemaer. Her har en tilgang til hele formatet gjennom skjemaer, og en fyller ut de feltene en ønsker å se på. En kan utføre et metadatasøk også her, men siden en har et skjema å fylle ut, slipper man å huske hva de aktuelle metadatapostene heter. Egner seg for brukere som ikke bruker systemet ofte.
- Makrosøk. Her defineres et ferdig søkeoppsett. Egner seg for uttak av lister med spesielle utvalg.
- Nøkkeloppslag.
  Dette er en spesial bruk av søkeskjema. Egner seg for gjenfinning av katalogposter som ennå ikke er indeksert.

3.4.2 Søking i internettdatabasen

I internettdatabasen har en mulighet til å søke på de ulike metadatafeltene internt på ”huset”. Man har ikke muligheten til å søke i den frie teksten slik det er i SIFT. Dette stiller større krav til at metadatapostene fylles ut grundig slik at en ikke går glipp av informasjon ved søk. Søkesystemet for leserne er satt ut til Sesam[21]. Dette er et firma som spesialiserer seg på internettsøk. Som nevn tidligere har en ekstern leser ingen mulighet til å spesifisere søket sitt utover å skrive flere ord inn i spørringen. Resultatet blir her heller ikke mulighet for sortering av noe slag.

3.5 Vurdering av Adresseavisen

Systemet til Adresseavisen har tydelig fokus på at det skal være enkelt for den enkelte journalist å kunne levere artiklene sine. Når journalisten skriver en artikkel legger han til noe metadata, men denne nyttiggjøres til liten grad. Når en artikkel kommer inn i systemet legger internettredaksjonen og arkivpersonalet til mer metadata til artikkelen før den legges inn i ulike databasesystemer.

Illustrasjoner og filmer er slikt man i en avis gjerne kan bruke flere ganger, men disse er ikke indekserer i forhold til motiv, eller hva filmen/illustrasjoner dreier seg om. Dette gjør at de er vanskelig å gjenbruke.

Det at artiklene lagres i ulike databaser, med ulikt metadataformat, gjør gjenfinningen vanskeligere enn om en kunne søkt i en felles database. Ulikt metadataformat gjør også at det ikke er mulig å gjenbruke metadata som alt er skrevet for en database. En slik dobbeltlagring tar også både unødig lagringsplass, og gjør vedlikehold og oppdatering av innholdet vanskeligere. Dette bruker også menneskelige ressurser i form av tid.

Samlet sett fremstår Adresseavisen sitt metadataformat som noe tungvint og uoversiktlig. For nye brukere er terskelen for å sette seg inn i eksisterende systemer større enn hva den kunne ha vært.

Hvis vi går tilbake og vurderer Adresseavisens søkesystem mot Clevordons kriterier for å vurdere et informasjonssøk, beskrevet i kapittel 2.3. vil vi se følgende:

- *Recall*; prosentandelen av de relevante dokumenter som er gjenfunnet. Dette er nesten helt umulig å etterprøve i en virkelig database. I punkt 2.6 beskriver jeg et tilfeldig søk der jeg søkte i internettavisen og ikke fikk noen treff på en artikkel som jeg viste var der, selv om jeg hadde korrekt tittel. Dette viser recall prosenten for nettavisen på akkurat denne emne var sært lav. Normalt får finner den treff, men hvor mange artikler som finnes innen de ulike emnene er uvisst.
- *Precision*; systemets evne til å gjenfinne kun de dokumentene som er relevante. Internt i avisen brukes metadatasøk, der metadata er tilført artiklene av enten journalistene selv, eller av arkivpersonell. Dette gjør at presisjonen på slike søk skal i teorien bli høy, forutsatt at brukeren har søkt på rett søkefelt. For eksterne søkere er problematikken en helt annen. Siden man ikke har mulighet til å utføre metadatasøk eller på noen måte spesifisere søkefelt er det vanskeligere å skrive en spørring som gjør at du får veldig høy presisjon på søket.
- *Time lag*; hvis vi her ser på min utvide definisjon av begrepet; hvor lang tid hele søkeprosessen tar, ser man at dette kan ta unødig lang tid i Adresseavisen. Intern på avisen må man finne fram til ulike databaser og utføre flere søk for å finne alle delene av en artikkel, dette vil ta lenger tid enn å utføre kun ett søk i en database, uavhengig av hvor godt man kjenner systemet. Som leser av Adresseavisen er selve søket enkelt å gjennomføre, men å finne det jeg ønsker en mer omstendelig prosess, da jeg må utvide og presisere søket for å få et overkommelig svar. Søkeren må derfor gjennomgå Marchionini søkeprosses som beskrives i punkt 2.3 flere ganger.
- *Effort*; hvor mye krever det av brukeren intellektuelt og fysisk for å innhente svar. Det er nok her den største utfordringen ligger med Adresseavisens system. En journalist i avisen må rett og slett kjenne systemet godt for å vite hvilken database han skal søke i og hvilke metadatafelt han må bruke for å finne de nyhetsobjektene han er ute etter. En utenforstående leser som søker i nettavisen kan enkelt utføre en spørring. Men på spesifikke emner der det for eksempel finnes mange relevante nyhetsobjekter må han være flink til å skrive gode spørringer for å finne det den søker etter.
- *Form of presentation*; hvordan blir søkeresultatet presentert. Internt på huset har de mulighet til å sortere svaret på en spørring etter ulike felt som for eksempel dato. Men for eksterne brukere finnes det helt klart et stort forbedringspotensial. Ninas spørring om ”Krigen i Irak” er nesten helt verdiløs når den bare blir presentert med overskrift, og bruddstykker av teksten, uten noen form for vekting eller sorteringsmuligheter.
- *Coverage of the collection*; i hvilken grad systemet inneholder relevant materiale. Når man ser på Adresseavisen er dette et rent redaksjonelt spørsmål og faller utenfor denne oppgaven.

Kapittel 4 Ulike standarder og formater

I kapittel 3 ble Adresseavisen sitt system gjort rede for. De har i stor grad laget sitt eget format, som i utgangspunkt skal passer de veldig godt. Flere aviser har gjort det samme. Men hva slags andre alternativ finnes?

I dette kapittelet beskrives ulike alternative måter for datagjenfinning. For å gi et overblikk over hvilke standarder som finnes presenteres de mest kjente som er i bruk innenfor nyhetsindustrien. De formatene som finnes og er i bruk i dag er hovedsakelig utviklet av den internasjonale presse og telekommunikasjons interesseorganisasjon IPTC[2]. Hvilke formater som brukes bestemmes av hver enkelt organisasjon, men organisasjoner som i utgangspunktet skulle ha sammenfallende behov bruker i dag mange ulike formater.

Norsk Telegrambyrå[22] er Norges største og ledende leverandør av nyheter til aviser, radio og tv-stasjoner. NTB distribuerer både tekst i en elektronisk konvolutt som kalles ”Information Interchange Modell” (IIM), utviklet av IPTC. Inne i denne konvolutten ligger teksten enten i IPTC 7901-formatet eller som NIFT format avhengig av hva kunden ønsker[23]. NTB formidler ikke bilder med overlater dette til Scanpix Norge AS[24]. Reuters, som er det britiske telegrambyrå, benytter seg blant annet av NewsML[25].

Det vi ser er at det er laget ulike format for ulike deler av nyhetenes livssyklus. Jeg har valg å sette opp de ulike tematisk etter hvilken del av arbeidsprosessen de støtter. Inndeling av kategorier er ikke absolutt og man vil se at noen av formatene vil dekke flere tema.

4.1 Metadataformater

Under denne kategorien finner kun de formatene som kun tar seg av metadata, altså formater som bare beskriver innholdet til for eksempel en nyhetsartikkel.

4.1.1 IPTC Subject Codes

Dette er en internasjonalt akseptert liste over emnenavn, emneformål og andre emner detaljer for kategorisering av nyhetsinnhold. Emnekvalifikator er utviklet innen sport for å berike referansesystemet. Dette er et standardvokabular som brukes både i NewsML og i NITF. Nyhetsemnene blir organisert i et omfattende hierarki og beskriver typer av nyhetsartikler. De ulike komponentene er objekter, attributter, emnereferanse, synonymer og kvalifikatorer. En artikkel blir altså beskrevet i flere nivåer:

- MediaType. Angir medietype for objektet. Kan være tekst, grafikk, foto, lyd, video og animasjon.
- ObjektType. Hvert objekt er av en type. Det er definert følgende typer: "*Nyheter", "Data", "Veiledning", "Vedlikehold", "Emnesett", "DTD", "Dokument", "Katalog", "Advarsel".*
- Attributter. Det er definert en rekke ulike attributter som kan tilordnes objektet.

Disse attributtene beskriver karakteristiske trekk ved nyhetsobjektet, ikke ved selve innholdet. Attributtene kan også tilordnes objekttypen. Det er definert 35 forskjellige attributter, blant annet: *"Arkivmateriale", "Bakgrunn", "Egenskap", "Hensikt", "Historie", "Pressemelding" og "Intervju"*.

- Emnereferanse. Det er definert 17 ulike emner, som "*Kunst, kultur og underholdning*", "*Kriminalitet, Lov og rettferdighet*" og "*Utdanning"*. Disse emnene blir definert i tre undernivåer.
- Synonymer. Angir ulike ord som kan korrelere til de ulike emnene. Synonymene bør så langt det er mulig være i naturlig språk.
- Kvalifikatorer. Angir hjelpetermer til emnerreferansene. Kan ikke stå alene, men må stå i sammenheng med den emnereferanse den hjelper[26].

4.1.2 PRISM Publishing requirement for Industry Standard Metadata

PRISM er et utvidbart XML-basert pakke og metadataformat basert på RDF. Dette formatet ønsker å tilby et standard metadatavokabular for å oppnå optimal nytte av utvekslingen av nyheter. PRISM tilbyr et rammeverk for utveksling og oppbevaring av nyhetsinnhold og metadata. I tillegg tilbyr de et kontrollert vokabular som beskriver innholdet i det materialet som blir utvekslet. Tilbyr en generell beskrivelse av ressurser som en helhet, spesifiserer en ressurs i forhold til en annen ressurs, definerer intellektuelle egenskaper, rettigheter, premisser og mulighet til å uttrykke nødvendig metadata. Standarden tilbyr ingen støtte for utveksling, rettighetstildelig og ulike sikkerhetsaspekter. Standarden er brukt av blant annet Netscape Corp., Time inc. og Wavo inc[27].

4.2 Formater for utveksling av nyheter

Disse formatene er rene overføringsformater, og tar seg av overføring av nyhetsdata. De innholder gjerne informasjon om for eksempel hvor stor datafilen som skal overføres er.

4.2.1 Information Interchange Modell

Information Interchange Modell er pakkefilformat for nyhetsinformasjon. Denne informasjonen kan være bilder, tekst eller grafikk. Det gjør at man kan knytte redaksjonell informasjon til det dataobjektet som skal sendes. Denne informasjonen kan gjelde alle kjente medietyper. Standarden inkluderer unik identifikasjon av nyhetsobjekter, lenkemekanismer mellom objekter, lyd og data parametere.

Modellen har en fem lags struktur som består av følgende lag:

- Object Envelope Record. Dette segmentet er obligatorisk. Innkapsler alle typer av objektdata som er en samling av binærdata, slik som bilder og lignende som er den essensielle dataen som skal bli presentert.
- Application Records. Gir støtte for tidligere versjoner av IIM.
- Pre-ObjectData Descriptor Record. Segmentet er obligatorisk og beskriver størrelsen på den kommende objektdatafilen.
- ObjectData Record. Er obligatorisk og innholder objektdatafilen, med tilhørende informasjon.

- Post-ObjectData Descriptor Record. Segmentet er obligatorisk og bekrefter størrelsen på den objektdatafilen som er sendt[28].

4.2.2 Digital News photo Parameter Recorder

Digital News photo Parameter Record (DNPR) er et filformat for innkapsling av digitale nyhets-fotografiske data. Formatet tillater lagring av redaksjonell og teknisk informasjon i samme fil. Formatet definerer et "tagged" fil format, som ligner på TIFF som er standard format for lagring av bilder. Filformatet brukes blant annet til lagring av bilder i Adobe Photoshop i deres eget filformat (.psd). Datasettet som følger med bildefilen er ganske stort. Det innholder blant annet - "*Record Version*" - Et binært nr. som angir hvilken versjon av DNPR som har vært brukt. - "*Picture Number"* - Angir et universelt unikt referansenummer til bildet. Består bl.a. av tilbyder id og dato. - "*Pixels per Line"* - Angir antall piksler som de skannede linjene innholder. - "*Number of Lines"* - Angir antall skannede linjer[29].

4.2.3 ICE information and Content Exchange

En XML-basert åpen protokoll som forvalter og automatiserer datautveksling og resultatanalyse og syndikatforhold. ICE er en ren protokoll som vil kunne erstatte bruk av HTTP og FTP innenfor nyhetsdistribusjon[30].

4.3 Formater for strukturering av nyhetsobjeker

Disse formatene er mindre spesialiserte en de andre kategoriene. Noen av de vil kunne håndtere overføringsinformasjon, og noe metadata, men felles for disse formatene er at de i tillegg tilfører noe informasjon om struktureringen av dataene.

4.3.1 IPTC 7901. The IPTC Recommended message format

Denne globale standarden for nyhetsformidling ble lansert i 1979. Formatet er anbefalt for overføring av binærtekst til avisredaksjoner, nyhetsagenter og andre mottakere, og er utviklet for overføring av telekommunikasjonsmedier som telefaks, teleks, modem og satellitt.

Den siste versjonen(R5) av IPTC-7901 kom i 1995, og formatet regnes som ferdig utviklet, da IPTC nå fokuserer på mer internettrelaterte standarder som NewsML og News Industry Text Format som er XML-baserte.

Formatet har hatt veldig stor oppslutning i Europa og til dels i USA, men har også vært brukt andre steder i verden. Bruken av formatet er nå på vei ned, etter hvert som det har kommet andre, mer omfattende og oppdaterte standarder.

Formatet er sammensatt av fire hovedseksjoner:

- Preheader information: Dette feltet er ikke spesifisert av IPTC, men det kan innholde kontrollkoder som trenges for overføringer, slikt som koder for å klargjøre og synkronisere avsender- og mottaker-maskinene.
- Message header: Dette feltet overfører informasjon som er påkrevet hos mottaker, for at redaksjonen som mottar meldingen skal kunne bruke den. Det er definert følgene meldingshoder: Start av meldingshode, kildeidentifikasjon og meldingsnummer,

prioritet, kategori, antall ord, fri informasjon, nøkkelord, start av tekst. Markerer at selve meldingen kommer.
- Message text: Dette feltet innholder selve meldingen, og kan innholde alle tegn som er definert i tegnsettet, med enkelte unntak som jeg velger å ikke komme inn på her.
- Post-text information: Markerer slutten på meldingen og legger i tillegg til metadata om dato og tid, før den markerer slutten på overføringen[31].

4.3.2 News Industry Text Formats (NITF)

NITF er utviklet av IPTC og Newspaper Association of America (NAA), og er en XML-basert standard, dette betyr at NITF bruker XML til innhold og struktur i et nyhetsdokument[32]. NITF er som HTML delt i to hoveddeler, ”<head>” og ”<body>”.

NITF innholder mange elementer som kan skille innhold som opptrer innen overskrifter, ”bylines”, tabeller, lister og avsnitt. Disse berikede tekstelementene tillater utgiver å indeksere og markere dokumenter bedre, og legge hyperlenker til rikere kilder av relatert informasjon og til det som skal arkiveres. Attributter av disse berikede tekstelementene tillater utgiver å lagre—”inline” med teksten – beskrivende koder. Disse attributtene kan sørge for konsistens og pålitelighet blant forfattere, skriftstil og språk. Denne berikede teksten støtter ”markup” for:

- hvem eier opphavstretten til et nyhetsobjekt, hvem kan utgi det på nytt og hvem handler det om
- hvilke emner, organisasjoner, og hendelser dekker nyhetsobjektet
- når ble nyhetsobjektet rapportert, utgitt og revidert
- hvor ble nyhetsobjektet skrevet, og hvor hendelsen fant sted, og hvor det kan utgis
- hvorfor har nyhetsobjektet interesse, basert på redaksjonens analyse av metadata[33]

4.3.3 XMLNews

XMLNews er et undersett av NITF som ble utviklet av David Meggison i 1999. Formatet er basert på RDF, og ønsker å gi større fleksibilitet og utvidelsesmuligheter enn NITF. XMLNews består av to deler:

- XMLNews - Story, som definerer innholdet i et tekstlig nyhetsdokument
- XMLNews – Meta, definerer et sett av metadata om nyhetsobjektet. XMLNews – Meta kan benyttes til å oversende metadata om hvilket som helst nyhetsobjekt, om det er en XMLNews – story, altså ren tekst, eller om det er en annet type objekt som innholder for eksempel bilder eller videoklipp[34].

NITF har etter utviklingen av XMLNews blitt forenklet og blitt enklere i bruk, og dekker behovene som XMLNews ble laget for i større grad. XMLNews blir ikke videreutviklet[35].

4.4 Et helhetlig format

På markede i dag finnes det kun et offisielt som kan brukes i alle deler av nyhetsproduksjonen til for eksempel en avis.

4.4.1 NewsML

NewsML er utviklet av IPTC og har til hensikt å kunne støtte alle deler av nyhetsformidlingens livssyklus i et elektronisk miljø. Det håndterer utveksling og lagring, metadatahåndtrening. Det kan gi nyhetsobjekter struktur i forhold til hverandre, uavhengig av hva slags nyhetsmedium nyheten skal publiseres i. Det som er spesielt med NewsML er støtter lagring av alle typer mediefiler som skal i en artikkel innhold. NewsML representerer ikke noen form for layout eller andre presentasjonselementer.

Man kan logisk representere NewsML ved hjelp av følgene hovedelementer:

- Nyhetsstruktur: NewsML representerer strukturen til et nyhetsobjekt ved hjelp av en kompleks XML struktur, som innholder ulike andre komponenter. Denne strukturen er utvidbar, slik at den kan støtter alle mulige former for nyhetsobjekt, uavhengig av om de er en artikkel som skal publiseres på ulike språk og i ulike medium
- Nyhetsmetadata. Nyhetsmetadata som beskriver de ulike delene av et nyhetsobjekt er delt inn i 5 kategorier: beskrivende metadata, redaksjonell metadata, identifisering metadata og fysiske metadata.
- Nyhetsinnhold. I NewsML blir nyhetsinnhold lenket til selve XML strukturen ved hjelp av et innholdselement.
- Nyhetsforvaltningsdata. Også dette blir representert ved hjelp av et XML element. Her kan man tilføre blant annet den overføringinformasjonen man finner nødvendig[6].

4.5 Oppsummering

Som vi ser over er det mange formater som går over i hverandre, og nyhetsformidlerne må ta i bruk ulike formater for å dekke sine behov. Dette har gjort at ulike nyhetsformidlere enten har brukt flere format parallelt og/eller laget sine egne format. Adresseavisen har laget sine egne format, mens for eksempel nyhetsbyrå som NTB bruker NITF og IIM. NewsML er et format fra IPTC som tar sikte på å være et helhetlig format som skal kunne brukes gjennom hele livssyklusen til et nyhetsobjekt.

Kapittel 5 NewsML

Et av målene for oppgaven er å se om NewsML kan løse noen av problemene til Adresseavisen i forhold til at de har mange databaser og ulike metadataformat. I dette kapittelet gir jeg et bilde av NewsML og beskriver oppbyggingen av formatet.

5.1 Hensikten med NewsML

NewsML tar sikte på å være et kompakt, utvidbar og fleksibel strukturelt rammeverk for nyheter. NewsML har til hensikt å kunne støtte alle deler av nyhetsformidlingens livssyklus i et elektronisk miljø, og tillatter man å endre en nyhetspost over tid. NewsML er primært utviklet som et format for utveksling av nyheter. Det kan i tillegg brukes som et format for nyhetslagring, som støtte for utvikling, redigering, forvaltning og publisering av nyheter i et databasert nettverk.

Det er ikke spesialisert for å støtte ensidig papirbasert nyhetsutgiving, men inkluderer også andre spesifikke produksjonsmiljø ved å ta inn eksterne definisjoner i designet. NewsML er ikke laget som et verktøy for produksjon og redigering av nyheter, men kan ligge som en grunnmur for en applikasjon som tar seg av dette arbeidet. NewsML representerer altså ikke noen form for layout eller andre presentasjonselementer. Slike elementer blir inkluderte ved bruk av for eksempel ”stylesheets”, SMIL og lignende som et tillegg til NewsMLdokumentet.

NewsML baserer seg på XML, og kan ta i bruk i andre passende standarder og spesifikasjoner, som for eksempel IPTC Subject Codes som blir beskrevet i kapittel 2. Et NewsML-dokument vil altså være et XML-dokument, og må være valid (gyldig) i forhold til dokument-type-definisjonen (DTD) eller skjemaet til NewsML. NewsML støtter representasjon av elektroniske nyhetsposter, samlinger av nyhetsposter, relasjoner mellom nyhetsposter og postenes tilknyttede metadata. NewsML tillater flere representasjoner av den samme informasjonen og takler vilkårlige blandinger av medietyper, formater, språk og kodinger. Det er medieuavhengig, men har allikevel spesifikke mekanismer for behandling av tekst. Den legger ingen føringer for innhold og metadata, det er opp til den enkelte ansvarlige redaktør å bestemme.

5.2 Nyhetsstruktur

Hensikten med NewsML er å tilby et uniformt rammeverk innenfor de applikasjonene som formidler informasjon om ulike nyhetsposter og samlinger av nyhetsposter. For å oppnå dette målet må strukturen som blir brukt til å representere nyhetsobjekter være kompleks nok til å ha en abstrakt arkitektur.

5.2.1 Nyhetsobjekter

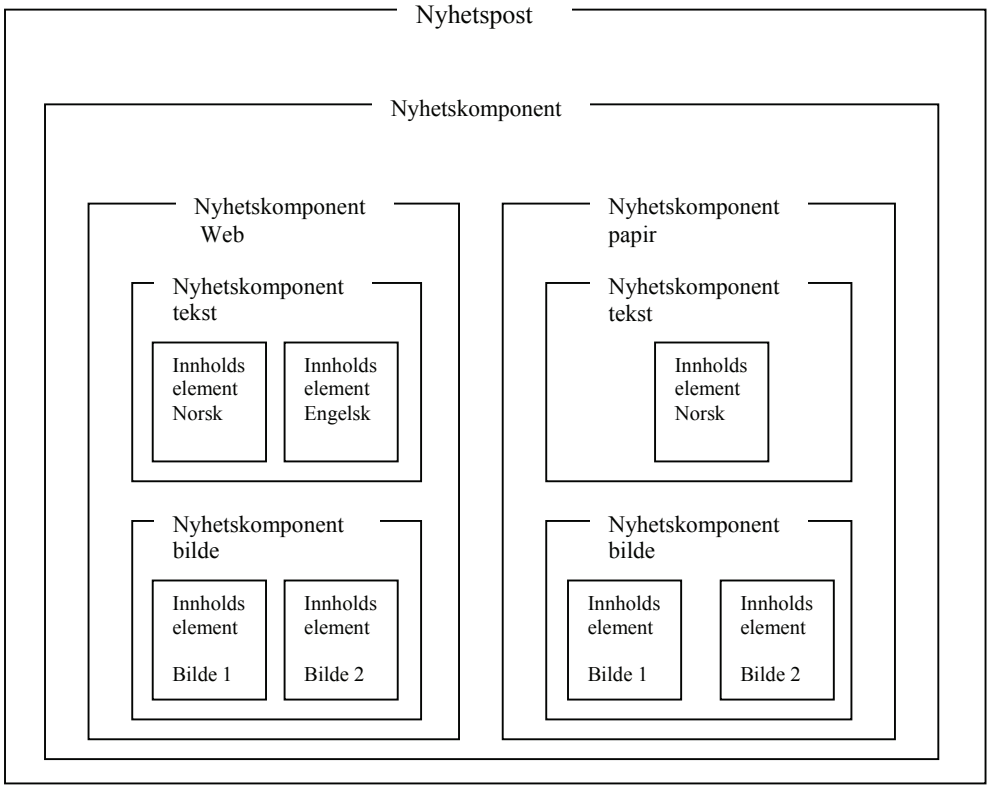
Dette er en av hovedelementene i et NewsML-dokument. Komponentene som inneholder eller referer til data kalles nyhetsobjekter. De tilfører publiseringsinformasjon til innholdet og på denne måten tilbyr de et synspunkt som relaterer til et spesifikt tidspunkt[36]. Denne navngivingen har til hensikt å omslutte begrepet om at data er kodet på en slik måte at de er i stand til å stå alene, men ikke gjør det i kontekst av en

nyhetspost. Et nyhetsobjekt har en lokal og en global identifikator og et sett med nyhetsmetadata som relaterer dem til tidspunktet og kilden som de representerer.

Siden man ser på et nyhetsobjekt, uavhengig av dets innhold, vil NewsML heller ikke kunne skille mellom nyhetsobjekter av ulik medietype, format eller kode. Man har ulike typer nyhetsobjekter; nyhetsposter, nyhetskonvolutter, nyhetskomponenter, og innholds elementer.

5.2.2 Nyhetspost

En nyhetspost representerer konseptet av en enkeltstående del av nyheter, overført av en organisasjon i henhold til deres egne redaksjons- eller produktregler om hva som utgjør "a piece of content of this kind". Innholdet i et nyhetsobjekt vil gjerne være synspunkter relatert til en eller flere hendelser, eller det kan være bilder, illustrasjoner, lydfiler, eller filmsnutter. NewsML legger ingen føringer for innholdet, men overlater dette til hver enkelt redaksjon. Uansett innhold skal et nyhetsobjekt innholde en unik identifikator, en tittel og en dato. Den kan i tillegg innholde et eller flere innholdselementer av ulik medietype. Nyhetsobjektet kan også innholde metadata og forvaltnings/administrative data[37].



Figur 6 En nyhetspost i NewsML, med ulike underkomponenter.

Ved å se i en vanlig avis i papirform eller på internett, eller går inn på internettsidene til et tv-selskap vil vi se at den vanligste nyhetsposten i dag består av strukturert tekst,

muligens tilknyttet et bilde. Men en ser at ”verden” utvikler seg i retning av at vi tilbys flere og flere muligheter, og redaksjonene trenger derfor muligheter til å kunne støtte visning av ulike typer nyhetsposter, som kan være knyttet til hverandre. NewsML støtter komplekse nyhetspoststrukturer, slik at en nyhetspost kan bestå av ulike deler av vilkårlig medietype, format, kode og språk. Delene kan igjen bestå av andre underdeler, nyhetsposter mm. NewsML definerer ikke koding for noen medietype.

Flere nyhetsposter kan utgjøre en samling, som blir dynamisk sammensatt som ved for eksempel respons til spørring i systemet eller de kan være forhåndsdefinert som ved en redaksjonell laget artikkel. I NewsML vil begge disse samlingstypene bli representerte på samme måte. Siden nyhetsposter er deler av samlinger som igjen kan være nyhetsposter, skiller ikke NewsML mellom nyhetsposter og samlinger av nyhetsposter. En nyhetspost er inngangspunkt til et nett av nyhetsposter relatert ved navngitte roller/lenker.

Et slikt inngangspunkt kan referere til store mengder nyhetsposter. Det er viktig å tilby en stor grad av fleksibilitet i forhold til inkludering og ekskludering av spesifikke innholdselementer for hver nyhetspost. På samme måte er det viktig å tilby en høy grad av fleksibilitet i forhold til hvilke deler av innhold som er eksplisitt levert som en del av en nyhetspost, og hvilke som er levert som referanse. Konseptet med inkludering og ekskludering er ikke begrenset til deler, men kan også referere til spesifikke elementtyper.

Objekter, som for eksempel bilder, kan bli inkludert eksplisitt eller ved referanse.
En måte NewsML er forskjellig fra en del andre systemer er muligheten til å sette ulike alternativer på samme nyhetsdel. Dette kan for eksempel være at brukeren foretrekker noen typer av data framfor noe annet. Brukeren kan for eksempel velge at han eller hun kun ønsker bildene vist i JPEG format, der det er mulig. På samme måte kan noen deler være fraværende, referert til ved en URI, eller innebygd i en nyhetspost.

Det er ikke mulig å bestemme ved å se på den fysiske formen til en nyhetspost om den er lik en annen nyhetspost. NewsML tilbyr attributter som blir knyttet til en nyhetspost, som løser dette problemet.

- Publisher: en URI som identifiserer utgiver og fastsetter navnerommet for det kommende ”item-id”.
- Item-id: En unik id som entydig bestemmer hvilken nyhetspost den aktuelle nyhetsposten stammer fra. NewsML dikterer ikke formatet til denne identifikatoren. Utgiveren må bare sikre at tildelingen er slik at mottaker ved hjelp av strengen kan sammenligne likheten til andre poster, men utgiver står fritt til å bruke IPTC UTO, som er designet med dette for øyet. Her gis med andre ord en valgmulighet i systemet.
- Revisjonsnummer: For å kunne se utviklingen av en nyhetspost over tid, blir et sekvensielt nummer tildelt nyhetspostene slik at de ulike revisjonene enkelt kan skilles fra hverandre.

NewsML bruker kombinasjonen av ”publisher”, ”item-id” og revisjonsnummer til å lage en global identifikasjon til nyhetsposten. NewsML bruker den samme metoden til å lage en identifikasjon til nyhetsobjekter. I tillegg gjør den det mulig for utgiver å lage ”stier”

til nyhetspostene. Denne ”stier” vil bestå av en URI som unikt identifiserer nyhetsposten. Det er også mulig å identifisere enkeltelementene en nyhetspost består av. Slike identifikasjoner kan brukes som referanser til de elementene eller sub-elementene, noe som vil lette arbeidet med å holde den publiserte nyhetsposten oppdatert.

Alle delene i en samling spiller forskjellige roller i samlingene. Mulige roller er primær, sekundær, tertiær osv. Disse rollene indikerer hvor viktig delen er i forhold til nyhetsobjektet. Et typisk eksempel på dette er en nyhetspost som hovedsakelig består av tekst, der bilde kommer som en sekundær del. NewsML dikterer ikke naturen til disse rollene, men lar hver redaksjon navngi forholdene som det passer. For å lette utveksling av nyheter er det laget et en standard liste av IPTC[38].

Deler med samme rolle har prioritet på grunnlag av hvor de står i markeringsspråket, men intensjonen med en slik prioritering er å rangere de ulike delene etter deres evne til å fylle sin rolle. En slik prioritering kan også bli brukt i noen applikasjoner som en bestemmelse av rekkefølgen for presentasjon. Meningen med å ha flere deler med samme rolle er å kunne tilby disse delene, i ulike formater, språk med mer. Disse alternativene kan for eksempel være JPEG eller GIF versjoner av samme bilde.

Komponentene til en nyhetspost, og eller en del av en nyhetspost, kan enten være markert som ”utfyllende” eller ”alternativer”. Delene som er merket ”utfyllende” vil sammen bidra til å lage en helhetlig nyhetspost. Er en del merket ”alternativ” vil delen kunne erstatte en av delene i den ordinære nyhetsposten, men vil inneholde den samme informasjonen. Dette kan være tilfelle for eksempel der en artikkel kan vises i ulike språk.

En nyhetspost av samme revisjon trenger ikke nødvendigvis å innhold de samme komponentene. Forskjellige manifestasjoner av en nyhetspost blir her referert til som en representasjon. Representasjon refererer også til leveringskoden i NewsML. Revisjon refererer til en redaksjonell bestemmelse om å endre innholdet.

Nyhetsposter må ha en tittel. Tittel er en tekst som beskriver innholdet. Denne blir brukt når en samling av nyhetsposter blir samlet og presentert i en liste, som ved en respons fra en nyhetsdatabase. Denne teksten kan være det samme som overskriften i nyhetsposten. En nyhetspost kan ha mer enn en tittel, men da må de ulike titlene representere forskjellige språk.

5.2.3 Innholdselementer

Et innholdselement representerer som navnet tilsier en enhet med innhold, administrert til å passe et nyhetsmiljø. Et innholdselement kan bli plassert i ulike kategorier etter medietypen. NewsML 1.0 definerer en rekke medietyper, og har man behov for å bruke andre medietyper utover disse kan man definere disse selv. Det er definert følgene medietyper i NewsML 1.0.:

- Tekst. Representerer en tekstdel som for eksempel kan være en artikkel. Slike strenger kan bestå av ren tekst, eller en XML-tekst.

- Grafikk. Representerer et stillbilde, i form av et bitmap eller en vektor.
- Foto. Representerer et digitalt bilde, et øyeblikksfoto av den virkelige verden.
- Lyd. Representerer en digital lydsekvens.
- Video. Representerer en digital videosekvens. En slik sekvens kan også inkludere en lydfil.
- Animasjon. Representerer en grafisk animasjon, i form av et bitmap eller en vektor. Kan bestå av dynamisk grafikk i 2D eller 3D, den kan være dynamisk eller en ren animasjon[36].

Et innholdselement blir også tildelt andre fysiske metadata som ”Format” ”Mimetype” og ”Betegnelse”, i tillegg til ”karakteristikker” og ”størrelse”[36].

Et innholdselement er ikke beregnet for å ”stå” alene i et nyhetssystem, men de er isteden inkludert i et nyhetsobjekt, hvor de blir lokalt identifisert.

5.2.4 Newslines

Nyhetslinjer er karakteristiske komponenter i et nyhetsobjekt; dette kan være overskrift, bildetekst, signatur, datolinje med mer. Komponentenes felles tema og informasjon om konstruksjonen av nyhetsobjektet blir representert ved metadata. Denne metadata blir representert i en form som er leselig for mennesker og den blir ofte vist ved siden av innholdet i nyhetsobjektet. Nyhetsposter, deler av nyhetsposter, og nyhetsobjekter kan alle ha en nyhetslinje. En kan lage flere nyhetslinjer til en nyhetspost som for eksempel presenterer informasjonene på ulike språk.

I NewsML er følgene ”News-lines” elementer definert.

Headline:	Utsagn som beskriver nyhetspostens innhold.
Caption:	En beskrivelse av nyhetsposten, vanligvis mer utfyllende enn overskriften.
Byline:	Anerkjennelse av personene som har laget nyhetsposten, kan suppleres med denne personens rolle.
Dateline:	Informasjon om når og hvor denne posten er laget.
Tagline:	Informasjon om produksjonen av posten.
Copyright:	Fastsetter eierskap og/eller bruksrestriksjoner.
Citation:	Informasjon om hvor en nyhetspost først var publisert.
Credit:	Anvisning av ”creator”/eier som ikke er nevnt i byline eller i Copyright utsagnet. Vanligvis den organisasjonen som står bak lagingene av posten.

Som vi ser definerer ikke NewsML bruken av ”news-lines”, og de sier ingenting om semantikken til innholdet. Dette er opp til hver enkelt redaktør[37].

5.2.5 Nyhetskonvolutter

NewsML kan også bli brukt til å overføre nyhetsposter mellom ulike redaksjonelle system, som for eksempel mellom Reuters og en lokal avis eller en tv-stasjon. En nyhetskonvolutt tilfører NewsML-dokumentet overføringsinformasjonsdata. Denne overføringsinformasjonen er atskilt fra innholdet i en nyhetspost, noe som tillater inkludering av en nyhetspost inne i en annen nyhetspost, uten gjentagende inkluderingen av produksjonsinformasjonen.

En nyhetskonvolutt får en overføringsidentifikator, informasjon om hvor den er sendt fra, og til hvem, fra hvilken leverandør, dato og tid for overføringen, i tillegg til informasjon om hvem som står for denne nyhetsservicen og nyhetsproduktet, og den kan også angi hvilken prioritet overføringen skal ha[36].

5.3 Nyhetsmetadata

NewsML kan beskrive metadata som faller inn i fem kategorier.

Beskrivende	Forteller hva nyhetsposten handler om, hva den referer til og hvem den kan være av interesse for. Kan enten referere til nyhetsposter i seg selv eller nyhetsposter og deler som en helhet. Verdiene er arvet fra de delene helheten innholder.
Redaksjonell	Forteller hvordan, når, hvor, hvorfor og av hvem innholdet er laget, publisert og distribuert. Forteller også hvem som eier rettighetene til nyhetsposten, bruksrestriksjoner og lignende. Som beskrivende metadata kan denne metadatatypen enten referere til nyhetsposter i seg selv eller nyhetsposter og deler som en helhet. Verdiene er arvet fra de delene helheten innholder.
Identifisering	Innholder vesentlige attributter til nyhetsposter eller objekter. For eksempel dato og identifikasjon (publisher, item.id og revisjon). Kan referer til nyhetsposter eller objekter, men ikke deler. Verdiene er arvet fra de delene helheten innholder.
Rolle	Forteller hvorfor delen er tilknyttet nyhetsposten. Det er kun deler som tilknyttes roller.
Fysiske	Angir mediatype, språk, og fysiske egenskaper ved innholdet. Dette kan være høyde, vidde, lengde, avspillingstid, fargesammensetning og lignende. Fysiske metadata kan kun referere til nyhetsobjekter.

5.3.1 Beskrivende metadata

Beskrivende metadata binder sammen nyhetsposter, deler og nyhetsobjekter. Denne typen metadata beskriver, som navnet tilsier, innholdet i den nyhetsposten eller objektet den er tilknyttet. Beskrivende metadata har to underklasser; kategorisering og navngitte entiteter. Disse klassene har mulighet til å spesifisere et attributt for applikasjonen til metadata, for eksempel mulighet til å spesifisere hvem som har bestemt at en nyhetspost skal være i en bestemt kategori, og mulighet til å spesifisere hvor stor betydning man tror metadataattributtet har. Man har også mulighet til å legge til informasjon om metadata som er blitt automatisk tilordnet, eller tilordnet av mennesker, og si noe om sikkerheten til applikasjonen sin metadata.

- Kategoriseringsmetadata; beskriver hva innholdet handler om, potensielle målgrupper, nyhetspostens prioritet, sjanger, detaljnivå osv. Denne metadataen er

alltid representert ved verdier fra et kontrollert vokabular, for eksempel koder. Det er mulig å tilknytte koder fra ulike parallelle kontrollert vokabular. Hvis man gjør dette må det kontrollert vokabular kodene er hentet fra må identifiseres, samt at man må identifisere hvem som har laget dette skjemaet. Man kan også angi rollen denne koden har, for eksempel hvis koden angir språk, sted og lignende.

- Metadata for navngitte entiteter; identifiserer de fakta som er referert til i innholdet, og spesifiserer stedet til referansen i det relevante media. Navngitte entiteter kan være for eksempel ting, mennesker, steder og lignende. Typen entitet som blir identifisert, er en verdi fra et kontrollert vokabular, som ved kategoriseringsmetadata, men man kan også ha en tekstlig beskrivelse av entiteten[37].

5.3.2 Redaksjonelle metadata

Denne metadataen beskriver publiseringinformasjonen relatert til nyhetsposten eller nyhetskomponenten. Redaksjonell metadata er primært et sett med navngitte entitets referanser, data og lignende. Disse kan også assosieres med koder, for eksempel kan navnet på landet der nyhetskomponenten er laget bli spesifisert ved for eksempel ISO 3166-kode. NewsML tillater bruk av flere skjemaer, og fastsetter ikke hvilke skjemaer som skal brukes[37].

5.3.3 Identifikasjonsmetadata

Som beskrevet før har nyhetsposter og nyhetsobjekter en post-id og et revisjonsnummer som er entydig ved hjelp av en utgiver id. Nyhetsposter har i tillegg i sin identifikasjonsmetadata en dato. Det er viktig det spesifiseres hva denne datoen representerer, den kan for eksempel forteller når posten er laget, når posten er utgitt og lignende. Individuelle elementer kan også ha unike (innenfor nyhetsposten) id'er som tillatter senere vedlikehold og forvaltning av nyhetspostene.

5.3.4 RolleMetadata

Som det fremgår av ovenstående tekst vil de ulike nyhetsdelene ha rollemetadata knyttet til seg, som beskriver forhold til de andre nyhetsdelene de er en del av.

5.3.5 Fysiske metadata

Her beskrives medietype, språk, og fysiske egenskaper ved innholdet. Dette kan være høyde, vidde, lengde, avspillingstid, fargesammensetning og lignende. Fysiske metadata beskriver også hvilket format den originale posten eller objektet var laget med[37].

Korrekte verdier for metadatatyper og verdier blir identifisert av en liste eller et skjema som kalles et kontrollert vokabular. Konsekvent gir ikke NewsML føringer for hva slike skjema skal inneholde, men for å lette overføring mellom nyhetsbyrå og mediebedrift finnes det en liste med standardverdier som blir vedlikeholdt av IPTC. Et slikt skjema kan enten være et firmaspesifikt skjema, eller det kan være et domene skjema. Når man spesifiserer en verdi, må man også identifisere skjema denne verdien er hentet fra.

Når en verdi er spesifisert fra et kontrollert vokabular i NewsML må følgende informasjon oppgis.

- Verdi.
- Skjemanavn der verdien er hentet fra.
- Tilgangsautoritet på navngitte skjema (URI)[6].

5.4 NewsML i praksis.

Her vil jeg kort illustrere hvordan en NewsML kan se ut.

5.4.1 Enkel artikkel i NewsML

Et NewsML-dokument trenger ikke å innholde noe innhold i form av tekst eller bilder. Så helt enkelt kan et NewsML-dokument kun innholde metadata. Hvis en kun tar med de obligatoriske postene vil et NewsML-dokument se slik ut:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE NewsML-system "http://www.naaf.no/dtd/NewsML.dtd">
<NewsML>
        <NewsEnvelope>
                <DateAndTime>20021120T145403+0100</DateAndTime>
        </NewsEnvelope>
        <NewsItem>
                <Identification>
                        <NewsIdentifier>
                                <ProviderId>naaf.no</ProviderId>
                                <DateId>20021120</DateId>
                                <NewsItemId>20021120.01.</NewsItemId>
                                <RevisionId PreviousRevision="”0” Update="U">1</RevisionId>
    <PublicIdentifier>urn:NewsMl:naaf.no:20021120.01.</PublicIdentifier>
                        </NewsIdentifier>
                </Identification>
                <NewsManagement>
                        <NewsItemType FormalName="News"/>
                        <FirstCreated>20021116T145403+0100</FirstCreated>
                        <ThisRevisionCreated>20021120T145403+0100</ThisRevisionCreated>
                        <Status FormalName="Usable"/>
                </NewsManagement>
        </NewsItem>
</NewsML>
```

Vi finner følgene informasjon her:

- ”*NewsEnvelope*”.
    - ”*DateAndTime*” den eneste obligatoriske elementet under nyhetskonvolutter. Formatet på dette punktet er fastsatt i dtden og har derfor ingen egen formatering.
- ”*NewsItem identification*”
    - ”*Identification/NewsIdentifier/ProviderId*: naaf.no. Verdien her er etter henhold til spesifikasjonen, et internett domene navn som er eid av provider ved den aktuelle datoen.
    - ”*Identification/NewsIdentifier/DateId*”: 20021120. Viser dato etter i henhold til spesifikasjonen.
    - ”*Identification/NewsIdentifier/NewsItemId*”: 20021120.01. Dette element innholder datoid, pluss et løpenummer som vil økes for hver nyhetspost. Jeg har valgt å ikke bruke tekst på dette elementet da en ved bruk av de feltene som er påkrevd i min applikasjon vil være andre elementer som overskrift, og stikkord som gjørs oss mennesker i stand til å forstå hva objektet handler om.
    - “*Identification/NewsIdentifier/ RevisionId*” PreviousRevision:””0” Update="U" verdi=1. Default verdi på dette elementet er “0” som indikerer at det er det et originalt objekt og ikke et oppdatert et. Hvis en nyhetspost innholder et eller flere oppdaterte element må en sette et ”*update*” attributt til ”U”. Hvis nyhetsposten kun består av nyhets forvaltningsdata må dette attributtet settes til ”A”. Hvis ikke noe av dette er tilfelle settes attributtet til ”N”.
    - “*Identification/NewsIdentifier/ PublicIdentifier*”: urn:NewsMl:naaf.no:20021120.01. URN som er skrevet i henhold til spesifikasjonen.

- “*NewsItem NewsManagement*”
    - ”*NewsManagement / NewsItemType FormalName="Nyheter"*. Denne verdien er hentet fra et kontrollert vokabular
    - ”*NewsManagement/FirstCreated*”: 20021116T145403+0100. Angir når den opprinnelige nyhetsposten ble laget, altså det objektet denne utgaven er revidert ut ifra.
    - ”*NewsManagement/ThisRevisionCreated*”: 20021120T145403+0100. Angir når denne spesifikke nyhetsposten ble laget.
    - ”*NewsManagement/Status FormalName*”: "Usable". Har her valgt å bruke IPTC sitt ”*topicset*”, da dette dekker det maksimale behovet jeg per dags dato kan se at en kan ha bruk for. Dette består følgende status verdier:
        - ”*Usable*”, - som forteller at nyhetsposten er uten restriksjoner.
        - ”*Embargoed*”, - som er at nyhetsposten og dens innhold er stengt for offentliggjøring inntil den blir godkjent for dette av utgiver.
        - ”*Withheld*”, - nyhetsposten eller dens innhold er ikke klar for utgivelse.
        - ”*Canceled*”, - Verken nyhetsposten eller dens innhold skal under noen omstendigheter offentliggjøres.

5.4.2 Multimedia artikkel i NewsML

For å tilføye dette dokumentet noe innhold er det ikke så mye som må legges til:

```
<NewsComponent>
    <NewsLines>
        <HeadLine> tekst </HeadLine>
        <ByLine> tekst </ByLine>
        <DateLine>20 november 2003</DateLine>
        <CopyrightLine>©2003 NAAF</CopyrightLine>
    </NewsLines>
    <AdministrativeMetadata>
        <Provider> <Party FormalName="NAAF"/> </Provider>
        <Creator> <Party FormalName="Ola Nordmann"/>      </Creator>
    </AdministrativeMetadata>
    <NewsComponent>
        <Role FormalName="Caption"/>
        <ContentItem>
            <MediaType FormalName="Text"/>
            <Format FormalName="bcNITF2.5"/>
            <DataContent>
                <p> tekst </p>
                <p> tekst </p>
                <p> tekst </p>
            </DataContent>
        </ContentItem>
    </NewsComponent>
    <NewsComponent>
        <Role FormalName="Preview"/>
        <ContentItem Href="images/213.jpg">
            <Media-Type FormalName="Photo"/>
            <Characteristics>
                <Property FormalName="Width" Value="384"/>
                <Property FormalName="Height" Value="256"/>
            </Characteristics>
        </ContentItem>
    </NewsComponent>
</NewsComponent>
```

Nye elementer lagt til i ny xml fil:
- *"News component"*

NewsComponent dette kan regnes som å være den viktigste NewsML bestanddelen, og innholder:

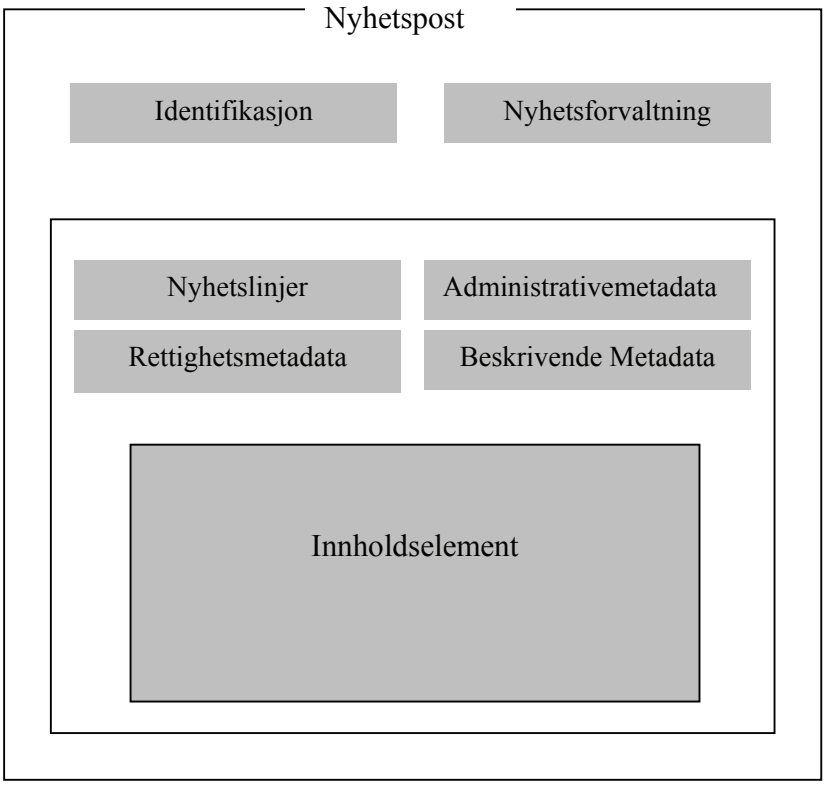
- "*Newslines*": Tekstlig informasjon, som er mulig å publisere direkte.
  - "*NewsLines/HeadLine*": nyhetspostens tittel
  - "*NewsLines/DateLine*": opprinnelsen til informasjonen,
  - "*NewsLines/CopyrightLine*" : NAAF copyright ,
- Administrative metadata
  - "*AdministrativeMetadata/Provider/Party/@FormalName*": tilbyder.
  - "*AdministrativeMetadata/Creator/Party/@FormalName*": Navn på forfatter.
- "*Caption*"
  - "*ContentItem/MediaType/@FormalName*": komponentens type (Text),
  - "*ContentItem/Format/@FormalName*": komponentens format (bcNITF2.5),
  - "*ContentItem/DataContent/*": overskriftens innhold.
- "*Picture*"
  - "*NewsComponent/Role/@FormalName*": komponentens rolle,
  - "*ContentItem/@Href*": innholds filens URL,
  - "*ContentItem/MediaType/@FormalName*": komtonentens media type,
  - "*ContentItem/Characteristics/Property[@FormalName="Width"]/@Value*" : image høyde,
  - "*ContentItem/Characteristics/Property[@FormalName="Height"]/@Value*": image bredde.

5.5 Vurdering av NewsML

NewsML fremstår som både veldig enkelt samtidig som det er komplisert. Det tar ikke hensyn til om man skal lagre tekst, lyd, bilder eller andre medieformat. Alt lagres som et nyhetsobjekt, noe som gjør at man får et begrenset antall metadata å forholde seg til.

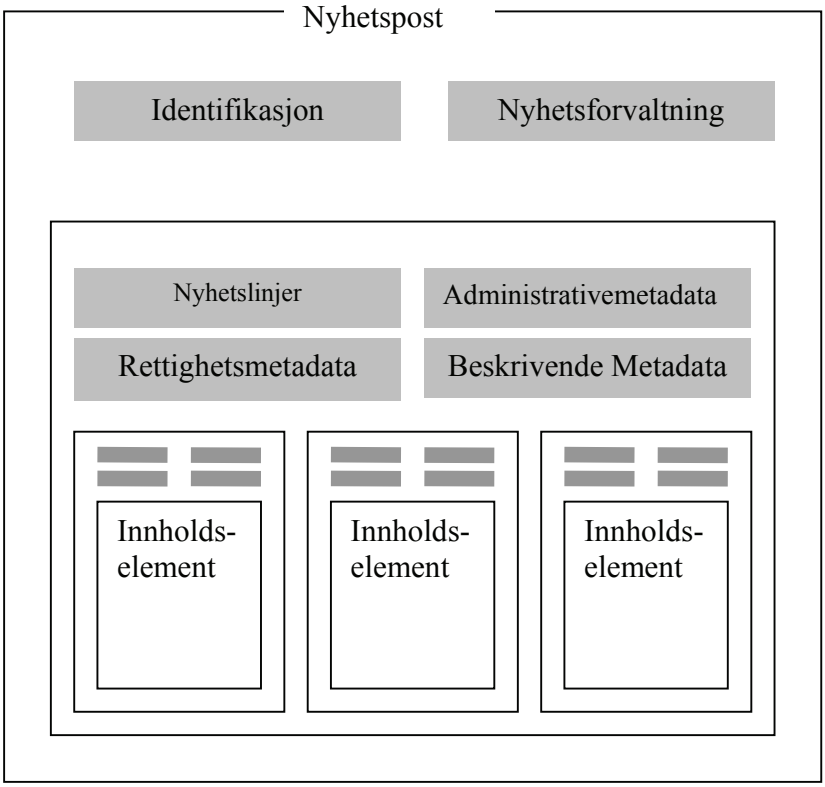
NewsML har en kompleks XML struktur som gjør at alle nyhetsobjekter i for eksempel en avis side med tekst, film og bilder, vil bli lagret i samme filen. Dette effektiviserer også gjenfinning i forhold til for eksempel Adresseavisens sitt system. Alle nyhetsobjekter kan finnes igjen med de samme matadata. Fordelen med at alle nyhetsobjektene som tilhører samme nyhetsartikkel ligger i samme fil, gjør at man slipper å gjøre oppslag i ulike databaser. XML formatet gjør også at det å støtte mange ulike format og medieenheter blir veldig enkelt. Man trenger ikke lage ulike artikler for de ulike mediene.

På sitt enkleste, kan et NewsML-dokument innholde kun metadata, men for at det skal bli en artikkel av noe slag må de i alle fall ha et innholdselement. Dette kan for eksempel være en tekst, lyd eller en film. Den vil kunne skisseres slik:



Figur 7 En enkel nyhetspost. Grå felt angir faktisk innhold.

En mer kompleks artikkel i NewsML, med for eksempel nyhetsobjeker i ulikt språk og tilpasset ulike medier, kan illustreres slik:



Figur 8 En nyhetspost, men mange innholdselementer[39].

NewsML er ikke rigid på hva slags metadata man skal bruke, dette gjør at standarden er åpen nok til at hver enkelt mediebedrift kan tilpasse formatet til sine behov. Samtidig er det enkelt å motta artikler som er skrevet i NewsML fra for eksempel et nyhetsbyrå. NewsML har støtte for lagring av ulike versjoner og markering av oppdateringer i samme dokument, dette gjør at det er mulig å bruke et NewsML-dokument gjennom alle de ulike prosessene en artikkel skal gjennom.

NewsML er derimot ikke et desktop publisher program som støtter skriveprosessene og bearbeidingene i en avis. Det har heller ingen støtte for lagring av fysiske objekter som negativ filmer og interne bøker. Dette er ikke like viktig i en avis der all produksjonen er eller snart vil være digital, men for eksempel tv-stasjoner baserer seg ennå på fysisk film i sin produksjon.

I forhold til Cleverdons kriterier for evaluering av et informasjonsgjenfinningssystem er det spesielt ”recall” og ”effort” som endrer seg i forhold til Adresseavisens sitt system. NewsML samler alle nyhetsobjektene i en database med et metadataformat, noe som gjør at sjansene for å få en god ”recall” på søket øker. Dette har en sammenheng med at det krever mindre av brukeren å velge rett database og skrive inn rett spørring til det aktuelle

systemet. Tiden det tar å utføre et søk kan også gå ned, da brukeren slipper å slå opp i flere databaser. Ellers vil et system basert på NewsML vil fremdeles basere seg på bruk av metadata og vil derfor ikke gi de store forskjellene i forhold til presisjon på søket.

Den største utfordringen til NewsML er nok å nyttiggjøre alle fordelene i praksis. Ved å støtte ulike medietyper, språk og mange ulike nyhetsobjekter, blir et NewsML-dokument relativt komplekst. Et godt dataprogram kan gjøre mye for å forenkle en slik kompleksitet, men å nyttiggjøre fordelene fult ut må også journalistene forstå strukturen i forhold til hvordan en nyhetspost kan inneholde ulike nyhetskomponenter som igjen kan innholde ulike nyhetskomponenter.

Kapittel 6 Mapping av Adresseavisen og NewsML

En artikkel i Adresseavisen vil normalt bli publiserte i både papir avisen og i internettavisen. Majoriteten av artikkelens data er felles i begge presentasjonene, men det er enkelte viktige skiller. En artikkel består ofte av en ”tittel”, ”undertittel”, ”ingress”, ”bildetitler”, ”bildetekst” og ”vignett” i tillegg til selve teksten. Kapittel 5 viser hvordan de ulike metadatapostene en slik artikkel kan ha, med bilder og vedlegg knyttet til seg.

En slik standard artikkel blir i Adresseavisen lagret i FotoStation, internettdatabasen og sift databasen (stoff arkivet og filmarkivet). I dette kapittelet synliggjøres hvordan de samme metadatapostene kan legges inn i NewsML. Hensikten er å vise og forklare hvordan de ulike elementene i NewsML fungerer. Dette er likevel ikke en full versjon av NewsML. For å gjøre dette mer oversiktlig er enkelte attributter og entiteter som i NewsML er mulig å ta med, utelatt. Dette finnes i NewsML DTD som finnes i vedlegg A, med nærmere spesifisering i den funksjonelle spesifikasjonen som finnes i vedlegg B.

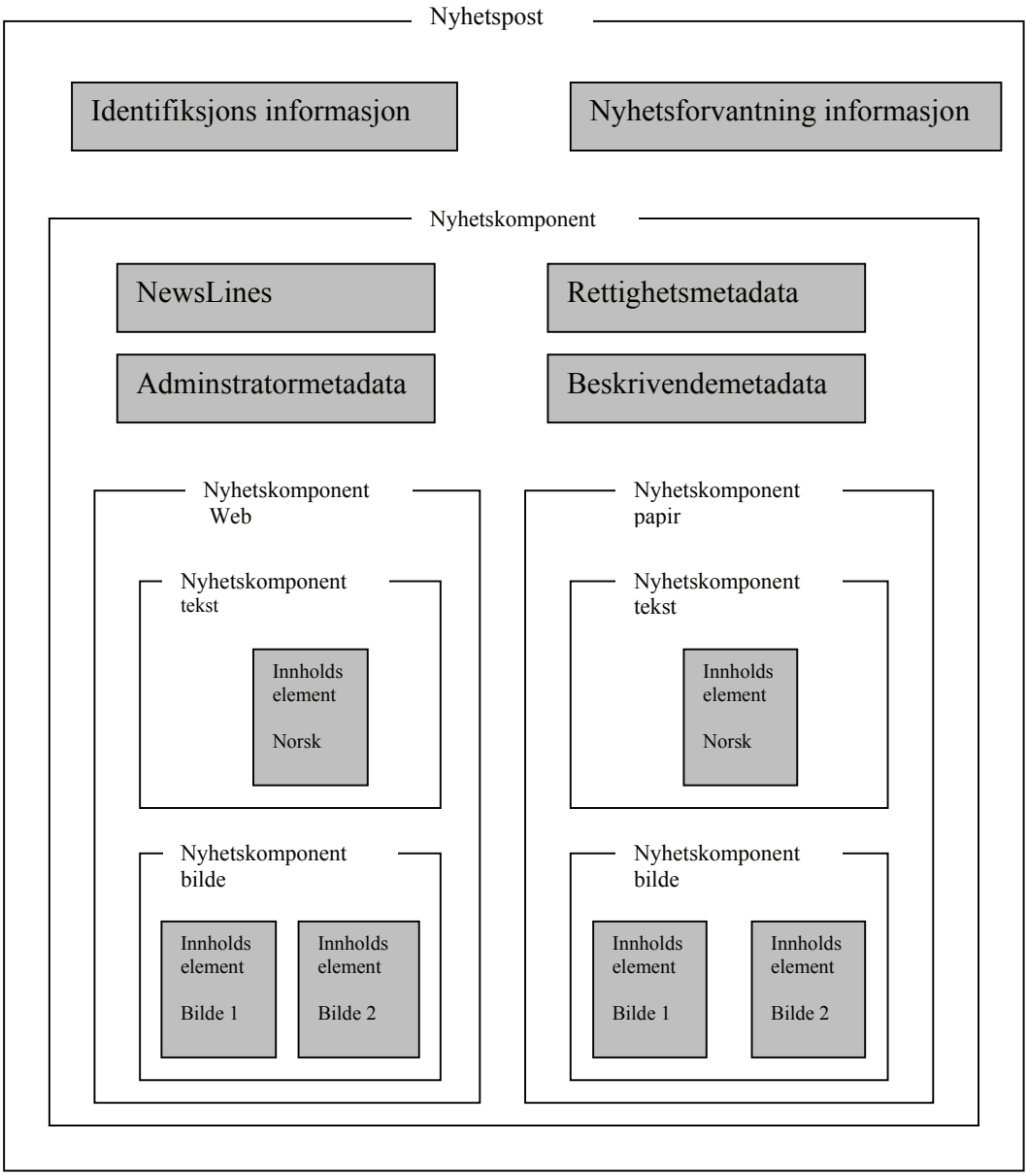
6.1 Ulike løsninger

Den enkleste måten å lage en avisartikkel i NewsML er å ikke skille mellom presentasjonsformene og lagre alle de ulike nyhetsobjektene inn i ett felles nyhetskomponent. Dette vil gjøre det veldig enkelt å lage ved at en alltid bare har en nyhetskomponent å forholde seg til. Samtidig vil en mange av fordelene med NewsML ved at en ikke kan trekke ut felles metadata for de ulike nyhetsobjektene og lagre disse i utenfor i ett felles nyhetskomponent.

En måte å ”overføre” en slik artikkel til NewsML på, er å skille papir og web i hvert sin nyhetskomponent, og igjen skille mellom innhold i form av tekst og i form av bilde. Fordelen med dette er at en skal skrive inn all metadataen som tilhører artikkelen som helhet inn i nyhetsposten eller den felles nyhetskomponenten, på denne måten slipper en unna dobbeltlagring. I tillegg til at den for eksempel på web ønsker å for eksempel ha en nyhetskomponent til som spesifiserer linker eller tekst som skal være i siderammen når artikkelen vises. Et annet alternativ er å snu på det, og lagre all tekst sammen i ett nyhetskomponent, og alle bildene i et annet nyhetskomponent dette vil gi andre felles metadatafelt enn varianten over.

Begge variantene blir forskjellig fra det Adresseavisen har i dag. Fordelen med å sette objekter som skal sammen i inn i et medium er at det synliggjør forskjellen journalistene. For eksempel kan tekst som er laget for web være lite relevant for andre publiseringsmedium, da ulike visningsmedium har forskjellig måte å presentere nyhetsstoff på. En ser tydelig forskjell hvis en sammenligner samme artikkel papirversjon og på nett, både når det gjelder lengde og mengden bilder. Slik Adresseavisen var strukturert da jeg var der hadde de en egen internettredaksjon som redigerte stoffet som skulle ut i nettavisen. På bakgrunn av dette har jeg - valgt å ta utgangspunkt i denne modellen der jeg skiller internett og papir utgaven av avisen og videre forklare hvordan det kan gjøres. Jeg velger også å se bort i fra Catalog, TopicSet og NewsEnvelope elementene som kan standard genereres for alle artiklene. Figuren under viser en

illustrasjon over hvilke hoveddeler en deler en slik ”modell” inn i, og jeg vil videre beskrive de enkelte feltene.



Figur 9 Mulig NewsML-struktur i Adresseavisen. Grå felter viser faktisk innhold.

6.2 Felles nyhetspost - newsItem

Hvis en bruker den strukturen som en ser i figuren over, vil hver post ha en identifikasjons del og nyhetsforvaltningsinformasjon del som er felles for hele artikkelen. Resten av nyhetsposten er nyhetskomponenter som inneholder tekst og bilder som blir brukt i de to ulike presentasjonsmedia som er valgt; papir og internett.

De to delene som nyhetsposten har felles for alle nyhetskomponentene er identifikasjons og nyhetsforvaltningsinformasjonsdel, disse to har fokus på gjenfinning og administrasjon av nyhetsposten, mens selve artikkelen ligger i nyhetskomponentene. En nyhetspost vil representere informasjon om en hendelse eller et synspunkt og lignende, en skal ikke blande ulike artikler inn i et newsItem.

Til selve nyhetsposten kan en knytte følgende informasjon;

Navn	Format	Beskrivelse
Comment	En string	Frivillig kommentar felt
Catalog	Resouce: Uniform Resource Name (URN) eller Uniform Resource Locators (URL). TopicUse: Herf peker.	Elementet kan innholde Resouce og/eller TopicUse element. Resouce elementene identifiserer en ekstern ressurs. Denne ressursen kan fungere som et standardvokabular for hele eller deler av hovedelementet. TopicUse elementet innholder et emne og indikerer hvor i NewsML-dokumentet ulike "topic" er brukt.
Identification	Innholder følgende underelementer.	
	Navn	Beskrivelse
	NewsIdentifier	Som beskrevet i punkt 6.1.1
	NameLabel	Frivillig element som innholder en tekststreng som gjør det mulig for mennesker å identifisere nyhetsposten. Trenger ikke å være unik.
	DateLabel	Frivillig element som innholder en tekststreng med dato, ev dato og tid, som gjør det mulig for mennesker å identifisere nyhetsposten. Trenger ikke å være unik.
	Labell	Frivillig element som innholder en tekststreng leselig for mennesker. Opp til hver enkelt hva den skal innholde.
NewsManagement		Beskrives i punkt 6.1.2
Valg av en av under elementene		
NewsComponent		Beskrives i punkt 6.1.3
Update		En oppdatering av ett eksisterende nyhetspost. Dette kan være et bilag, en erstatning eller en sletting. Kan innholde ett av underelementene; InsertBefore, InsertAfter, Replace eller Delete
TopicSet		Innholder et eller flere Topic elementer som refererer til reelle forekomster (topics). Dette kan være personer, steder, selskap ol.

Tabell nr. 7*"Felles nyhetspost"*

Et samleelement som skal innholde all informasjon om en spesifikk artikkel. Innholder blant annet et identifikasjonselement, og et nyhetsforvaltningselement, i tillegg til at en har mulighet til å legge inn enten en nyhetskomponent element, et update-element eller et topicSet-element.

6.2.1 Identifikasjonsinformasjon – NewsIdentifier

Som navnet sier er dette et element som identifiserer en nyhetspost.
Identifikasjonselementene er til for at en til enhver tid skal kunne identifisere hver enkelt nyhetspost uavhengig hvor i arbeidsprosessen den befinner seg. Dette er viktig ikke bare for å ha muligheten til å finne igjen en spesifikk nyhetspost, men også å entydig og enkelt kunne avgjøre hvilken nyhetspost som er den som sist ble redigert. Alle nyhetsposter skal derfor ha knyttet til seg et newsIdentifier elementet som innholder en globalt unik identifikator. Dette får vi ved å kombinere en tilbyderidentifikasjon, dato, en identifikator på nyhetsposten og en revisjons id.

Navn	Format	Beskrivelse
ProviderId	internettdomene navn eller URN	Elementet skal innholde et domene navn som er eid av Adresseavisen ved den aktuelle datoen angitt i dato-id elementet, eller det kan være et navn Adresseavisen henter fra et kontrollert vokabular identifisert ved en URN, som blir spesifisert i vokabular attributtet.
DateId	ISO 8601 Basic Format ÅÅÅÅMMDD (år, måned, dato).	Elementet innholder dato. Siden Dato-id er en del formelle identifikasjonen av en nyhetspost, må denne ”id”en forbli den samme gjennom alle revisjonene av den aktuelle nyhetsposten. Den vil ikke representere datoen for utgivelse av nåværende revisjon
NewsItemId	Egendefinert identifikator, må være lokalt unik.	Elementet skal innholde en identifikator for en nyhetspost. Kombinasjonen av nyhetspost-id og dato-id må være unik blant nyhetsposter i Adresseavisen sitt system. Det kan være et navn, meningsfylt for mennesker. En kan også tilordne nyhetspost-id til et kontrollert vokabular
RevisionId	Et positivt heltall	Elementet forteller hvilket revisjonsnummer en versjon av nyhetsposten har. Den nyeste instansen av en nyhetspost må alltid ha det høyeste revisjons-id’en. Hvis en nyhetspost innholder et eller flere oppdaterte element må en sette et ”update” attributt til ”U”. Hvis nyhetsposten kun består av nyhets forvaltningsdata må dette attributtet settes til ”A”. Hvis ikke noe av dette er tilfelle settes attributtet til ”N”

Valgfrie felter (uformelle identifikatorer)		
NameLabel	En string	Kan brukes for å identifisere nyhetsposten for mennesker. Som element navnet sier er dette et navn. Formen til navnet bestemmes av Adresseavisen selv. Kan ikke lages på flere språk
DateLabel	En string	Dato på det formatet som er bekvemt for brukeren
Labell	En string	Frivillig og menneskeleselig felt som består av "LabelType" og "LabelText" sub-elementene. Der en har et kontrollert vokabular for labell-type og fri tekst i "LabelText". Kan brukes til å beskrive ting som en ønsker at skal være lett leselig for mennesker, og som ikke dekkes av de øvrige punktene
Alternativt		
PublicIdentifier	URN	urn:newsml:{ProviderId}:{DateId}: {NewsItemId}:{RevisionId} {RevisionId@Update} Som beskrevet over.

Tabell nr. 8 ***"Identifikasjon av en nyhetspost"***

Et newsIdentifier elementet innholder en globalt unik identifikator. Dette får vi ved å kombinere en tilbyderidentifikasjon, dato, en identifikator på nyhetsposten og en revisjons id, kan alternativt formateres som en URN. En kan også knytte til 3 frivillige elementer.

6.2.2 Nyhetsforvaltning – newsManagement

Nyhetsforvaltningselementet innholder informasjon om nyhetspostens type, historie og status, samt hvordan forhold denne nyhetsposten har til andre nyhetsposter, og/eller spesielle instruksjoner til mottaker av nyhetsposten.

Navn	Format	Beskrivelse
NewsItemType	Egendefinert kontrollert vokabular	Elementet forteller hvilke type en nyhetspost er av. For eksempel; ”News”, ”data”, ”Advisitory”
FirstCreated	ISO 8601 Basic Format	Elementet forteller dato, og frivillig tiden som nyhetsposten først ble laget
ThisRevisionCreated	ISO 8601 Basic Format	Elementet angir revisjons dato for denne spesifikke nyhetsposten
Status	Egendefinert kontrollert vokabular. En kan bruke ”IPTC TopicSet. “	Angir nyhetspostens anvenlighet.
Urgency	Egendefinert kontrollert vokabular	Elementet forteller hvor viktig nyhetsposten anses for å være
Frivillig		
StatusWillChange	ISO 8601 Basic Format	Elementet angir når nyhetsposten automatisk kommer til å bli endret. Her angir en under elementene, ”FutureStatus” og ”DateAndTime”.
RevisionHistory	En eller flere stringer	Elementet angir en peker til en fil som innholder revisjonshistorien til nyhetsposten, angis ved et Href attributt. Utgiver kan selv velge syntaks og struktur på denne revisjonshistorie filen.
DerivedFrom	En eller flere stringer	Repeterbart element som gir en peker til den nyhetsposten den aktuelle er derivert fra.
AssociatedWith	En eller flere stringer	Repeterbart element som gir en peker til den nyhetsposten den aktuelle er assosiert med. Dette kan for eksempel være bilder, eller andre artikler om samme tema.
Instruction	Egen definert kontrollert vokabular	Repeterbart element som kan inneholde instrukser fra nyhets utgiver til det som mottar nyhetsposten.
Property	En eller flere stringer	Elementet kan brukes til å angi verdien for spesifikke egenskaper ved ”contentItem”, ”topic”, NewsComponent” og ”newsItem”.

Tabell nr. 9 ***”Nyhetsforvalning”***

Elementene gir informasjon som er relevant for forvaltning av en nyhetspost; dato for opprettelse, dato for siste modifisering, og nyhetspostens status. I tillegg kan en også legge til de frivillige feltene som forteller om viktigheten til informasjonen, om arv og

assosiasjoner mellom nyhetsposter, og/eller spesielle instruksjoner myntet på mottaker av informasjonen.

6.2.3 Felles nyhetskomponent - NewsComponent.

Nyhetsposten har en felles nyhetskomponent som omslutter de andre nyhetskomponentene, som en ser av figur 9. En nyhetspost kan betegnes som et samleobjekt for enten innholdselementer, andre nyhetskomponenter, nye nyhetsposter, eller nyhetspostreferanser. Brukes for å identifisere nyhetsobjektene i relasjon til andre, og tilegne administrative-, beskrivende- og rettighetsmetadata til objektet. Innholder også mulighet til å legge inn nyhetslinjer. Innholder ingen global unik identifikator så dette elementet må være nøstet inn i en nyhetspost.

Navn	Format	Beskrivelse	Default verdi
Attributter			
Essential	Yes | No	Attributtet indikerer om utgiver anser denne nyhetskomponentet til å være essensiell i forhold til konteksten der den står.	No
EquivalentsList	Yes | No	Attributtet indikerer om de elementene som nyhetskomponenten består av er alternative objekter. For eksempel når en artikkel er tilgjenglig på ulike typer språk	No
Navn	**Format**	**Beskrivelse**	**Default verdi**
Comment	En string	Repeterbart elementet gir mulighet for å skrive en kommentar	
Catalog		Som beskrevet over.	
TopicSet		Som beskrevet over.	
Role	En string	Elementet spesifiserer rollen en nyhetskomponent spiller i den nyhetskomponenten den er en del av	
BasisForChoice	Xpath mønster eller element-type navn	Hvis attributtetet ”equivalentsList” er satt til ”yes”, angir en de ulike komponentene her.	
NewsLines		Elementet angi karakteristiske egenskaper ved nyhetsobjektet. Disse er ment å tilføre en representasjon av metadata som er lett tilgjenglig for mennesker.	
	Har følgende underelementer (alle frivillige og repeterbare)		
	Navn	**Beskrivelse**	
	Headline	Elementet angir synlige overskriften.	
	SubHeadLine	Elementet angir synlige ”under-overskriften”	
	Byline	Elementet angir i naturlig språk informasjon om journalist/forfatter.	
	Dateline	Elementet angir i naturlig språk informasjon om dato og/eller stedet for når nyhetskomponenten ble laget.	
	Creditline	Elementet angir i naturlig språk kreditteringsinformasjon.	
	CopyrightLine	Elementet angir i naturlig språk informasjon om copyright. Dette forteller hvem som eier objektet.	
	RightsLine	Elementet angir den visbare versjonen av	

		rettighetsinformasjonen. Altså om hvem som har lov å bruke objektet og hvordan det da skal brukes.
	SeriesLine	Elementet angir den visbare informasjonen om objektets plass i en ev. serie.
	SlugLine	Elementet angir en streng av tekst (ev. med en hyperlink) som brukes til den visbare ”slug-line”. Hva som legges i ”slug-line” er opptil den enkelte redaksjon.
	KeywordLine	Elementet angir den eller de visbare nøkkelordene som er relevant for nyhetsobjektet.
	NewsLine	Elementet må innholde underelementene: - ”NewsLineType” element som er en bruker definert nyhetslinjetype, der verdiene hentes fra et kontrollert vokabular. - ”NewsLineText” elementer, som innholder tekst av den definerte ”NewsLineType”. Finnes det flere instanser av dette elementet bør de tilføres xml:lang attributtet som indikerer hvilket språk det er skrevet på.
Metadata		Elementet kan innholde underelementene; AdminMetadata, DescriptiveMetadata og RightsMetadata. Disse er beskrevet i punkt 6.1.3.1. til 6.1.3.3.
Valg	Så følges et repeterbart felt der man kan velge en av følgene under elementer NewsItem, NewsItemRef, NewsComponent eller et ContentItem	
	Navn	**Beskrivelse**
	NewsItem	Som beskrevet i punkt 6.1
	NewsItemRef	Elementet innholder en peker til et eksternt NewsItem element som kan erstatte NewsItemRef elementet.
	NewsComponent	Som dette hovedelementet, åpner for rekusjon.
	ContentItem	Elementet innholder et nyhetsobjekt, eller en peker til et, som innholder et dataobjekt med meningsbærende innhold, beregnet for mennesker. (f.eks. tekst, images, video, audio etc).

Tabell nr. 10 ***”Nyhetskomponent”***

Elementet identifiserer nyhetsobjekters relasjon til andre, og innholder enten et innholdselement, andre nyhetskomponenter, nye nyhetsposter, eller nyhetspostreferanser. Innholder også nyhetslinjer, og tilegner administrative, beskrivende og rettighets metadata til objektet.

6.2.3.1 Administrative metadata

Denne typen metadata forteller om opprinnelsen til nyhetskomponenten. To nyhetskomponenter i samme nyhetsdokument vil gjerne ha flere av de samme administrative metadata til felles, som kilde, tilbyder, forfatter og lignende. Men samarbeids dokument kan likevel gjerne ha ulike for eksempel journalist, fotografer eller redigerere. Hvis en journalist for eksempel har redigert en artikkel for å tilpasse den til webutgaven av avisen vil dette komme fram i administrative metadata under den nyhetskomponenten som innholder tekst tilpasset web. Alle underelementene er frivillige.

Navn	Format	Beskrivelse
Catalog		Som beskrevet i punkt 6.1
FileName		Elementet identifiserer anbefalte eller faktiske filnavn(path) for nyhetsobjektet.
SystemIdentifier	URL	Elementet identifiserer system addressen (som en URL) til hvor posten er å finne.
Provider	En string	Element identifiserer hvem som har utgitt nyhetsobjektet, kan være et individ og/eller en bedrift og/eller en organisasjon. Her kan en også legge inn et kommentarfelt.
Creator		Elementet identifiserer hvem som har laget nyhetsobjektet, kan være et individ og/eller en bedrift og/eller en organisasjon. Her kan en også legge inn et kommentarfelt.
Source		Repeterbart element som identifiserer kilden som har tilført kilde materialet til nyhetsobjektet. Her kan en også legge inn en ”NewsItem” attributt som innholder en URN til en nyhetspost, hvis kilden har en egen nyhetspost
Contributor		Repeterbart element som identifiserer hvem som har modifisert eller forbedret nyhetsobjektet etter at den ble laget. Her kan en også legge inn et kommentarfelt.
Property		Repeterbart element som kan brukes til å beskrive øvrig administrative metadata som ikke blir dekket av de øvrige.

Tabell nr. 11 ***”Administrative metadata”***
Elementet forteller opprinnelsen til nyhetskomponenten.

6.2.3.2 Rettighets metadata

Denne typen metadata innholder informasjon om opphavsrett, og rett til bruk vedrørende nyhetskomponentene. Som ved administrative metadata kan en se at enkelte nyhetskomponenter innholder for eksempel bilder som er hentet fra eksterne kilder, ikke har samme opphavsrett som resten av artikkelen, dette vil da merkes av under rettighetsmetadata for disse bildene. Alle elementene er frivillige.

Navn	Format	Beskrivelse
Catalog		Som beskrevet i punkt 6.1
Copyright		Repeterbart element som innholder underelementene ”comment”(kommentarfelt, frivillig), ”CopyrightHolder” og ”CopyrightDate”, som er elementer som i naturlig språk gir informasjon om hvem copyrighten tilhører og angir dato for den.
UsageRights		Repeterbart element angir informasjon om hvem som innehar rettighetene vedrørende en nyhetskomponent
	Har følgende underelementer (alle frivillige, men ikke repeterbare)	
	Navn	**Beskrivelse**
	UsageType	Elementet gir informasjon i naturlig språk om hvilke type bruk rettighetene retter seg mot.
	Geography	Elementet gir informasjon om hvilke geografiske områder rettighetene gjelder for.
	RightsHolder	Elementet forteller hvem rettighetene tilhører
	Limitations	Elementet angir eventuelle restriksjoner på bruken av innholdet i nyhetskomponenten
	startDate/ EndDate	Elementet angir tidsrommet rettighetene for denne nyhetsposten varer.
Property		Repeterbart element som kan brukes til å beskrive øvrig rettighets metadata som ikke blir dekket av de øvrige.

Tabell nr. 12 ***”Rettighets metadata”***

Elementet innholder metadata som gir informasjon om opphavsrett, og rett til bruk vedrørende nyhetskomponentene.

6.2.3.3 Beskrivende metadata

Denne typen metadata innholder informasjon som beskriver innholdet til en nyhetskomponent. Dette er en subjektiv type metadata i motsetning til de andre som kan settes mer eller mindre automatisk. Alle elementene er repeterbare.

Navn	Format	Beskrivelse
Catalog		Som beskrevet i punkt 6.1
Language	RFC 3066	Repeterbart element som angir språket på innholdet til nyhetsposten. Verdiene hentes fra et kontrollert vokabular.
Genre	Egendefinert kontrollert vokabular.	Repeterbart element som angir hvilken sjanger innholdet til nyhetsposten tilhører.
SubjectCode	Verdiene hentets fra IPTC ”subject Codes”.	Repeterbart element som angir emnet til nyhetsposten
OfInterestTo	Verdiene her er hentet fra et kontrollert vokabular.	Repeterbart element som angir målgruppen for nyhetsposten. Kan ha underelementet ”Relevance”.
DateLineDate	ISO8601 Basic Date Format.	Elementet angir når nyhetsposten ble laget.
Location		Repeterbart element som ved hjelp av et propery element angir lokaliserings informasjon og ved det frivillige HowPresent attributtet angir type lokalisering.
TopicOccurrence		Repeterbart element som angir emnet til nyhetskomponenten. Elementet har et ”howPresent” attributt som indikerer naturen til emnet, og et ”topic” attribut som angir en Duid.
Property		Repeterbart element som kan brukes til å beskrive øvrig beskrivende metadata som ikke blir dekket av de øvrige.

Tabell nr. 13*”Beskrivende metadata”*

Elementet innholder metadata som gir subjektiv informasjon om innholdet til en nyhetskomponent.

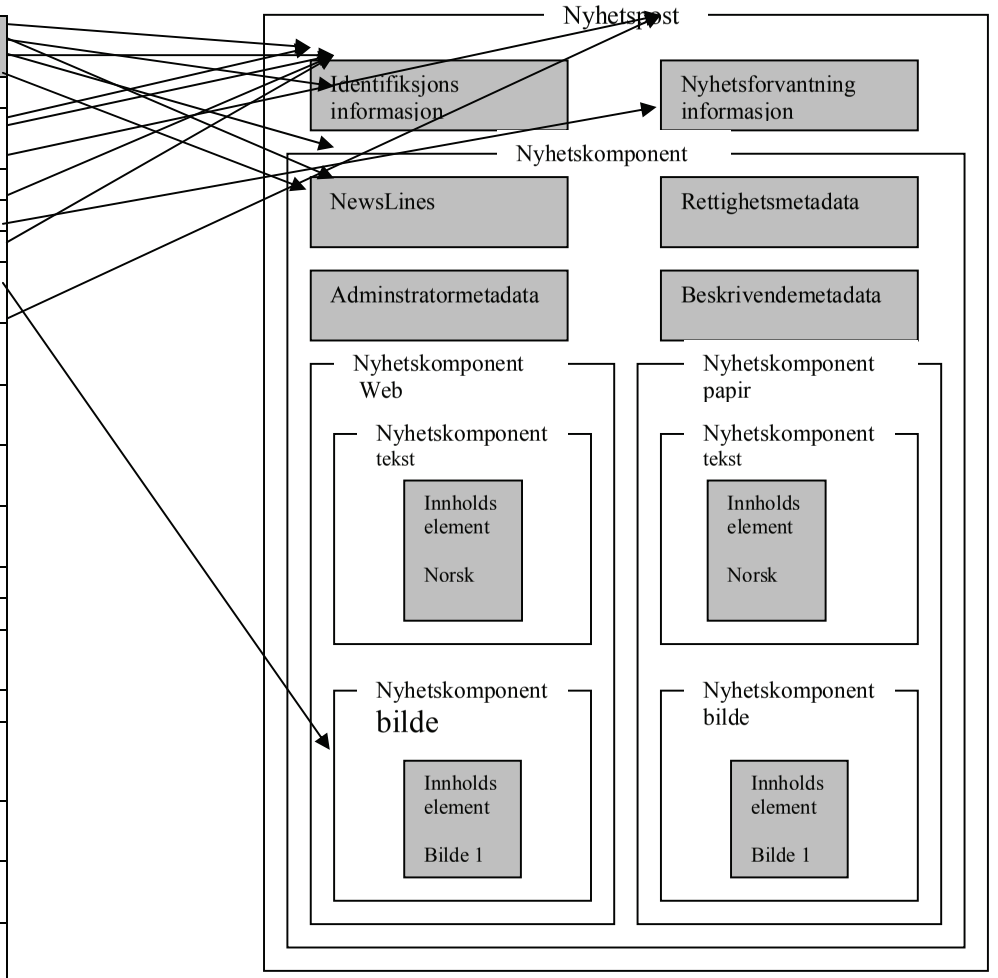
6.3 Adresseavisens og NewsML

Dette er svært annerledes fra slik Adresseavisen lagrer i dag, derfor beskrives de ulike metadatafeltene som avisen har i dag (jf kapittel 4), og deretter vises hvordan det vil kunne se ut i en NewsML-struktur. For å skape oversikt er de metadatafeltene Adresseavisen ikke bruker fjernet, med mindre de er obligatoriske i NewsML.

6.3.1 Indeksering av digitale bilder i FotoStation

Figuren under viser hvordan en kan ta metadatafeltene som finnes i Adresseavisen for lagring av bilder og legge inn i newsML. Vi ser at en del av feltene kan lagres i den felles nyhetsposten og nyhetskomponenten, mens andre må lagres i elementet der bildet ligger. På figuren er det kun trukket piler til en ”bildeboks”, men bilder som tilhører papirutgaven må selvfølgelig også få tilføyd metadata, dette er kun gjort for å få figuren til å se oversiktligere ut. Her vil en se at det blir noe dobbeltlagring, men hvis en sammenligner en nettavis og samme artikkelen i en papirutgave av avisen vil en se at det ganske ofte er forskjellige bilder som blir brukt.

Hoved-gruppe	Metadata	NewsML-element
Nøkkelord.	Nøkkelord	Keywordline
Bildetekst	Gruppe (objekt navn)	Dekkes ikke direkte
	Sak (overskrift)	headline.
	Instruksjoner	Instruction
	Bildestatus	Status
	Fotograf [1]	administrative metadata/creator
	Motiv	Beskrivende metadata/property
Data og status	Fotodato(Dato og tid)	first created
	Utgivelses dato(Dato og tid)	Dekkes ikke direkte
	Brukt dato (Diverse plassering)	Dekkes ikke direkte
	Side (provins/stat)	Dekkes ikke direkte
	Merknad(Diverse)	Comment
Kategorier og nøkkelord	Hva er skannet	beskrivende metadata/property
	Journalist[2]	AdminMetadata/creator.
	Returnert til (opphavsrett)	rettighetsmetadata/ Copyright
	Nøkkel ord (gruppe)	beskrivende metadata/ Genre
	Produkt (kategori)	Beskrivende metadata/property
	Produkt tillegg(kategori).	Beskrivende metadata/property
	Film-nr + Motiv-nr.(Diverse)	Dekkes ikke direkte



Figur 10 Metadataskjema for indeksering av digitale bilder/dokumenter i Fotostation.

[1] Her kan det også legges til et kommentarfelt om man ønsker å spesifiser at det er en fotograf som har tatt bilde

[2] Her kan det også legges til et kommentarfelt om man ønsker ytterligere spesifisering.

6.3.2 Indeksering av artikler for internett

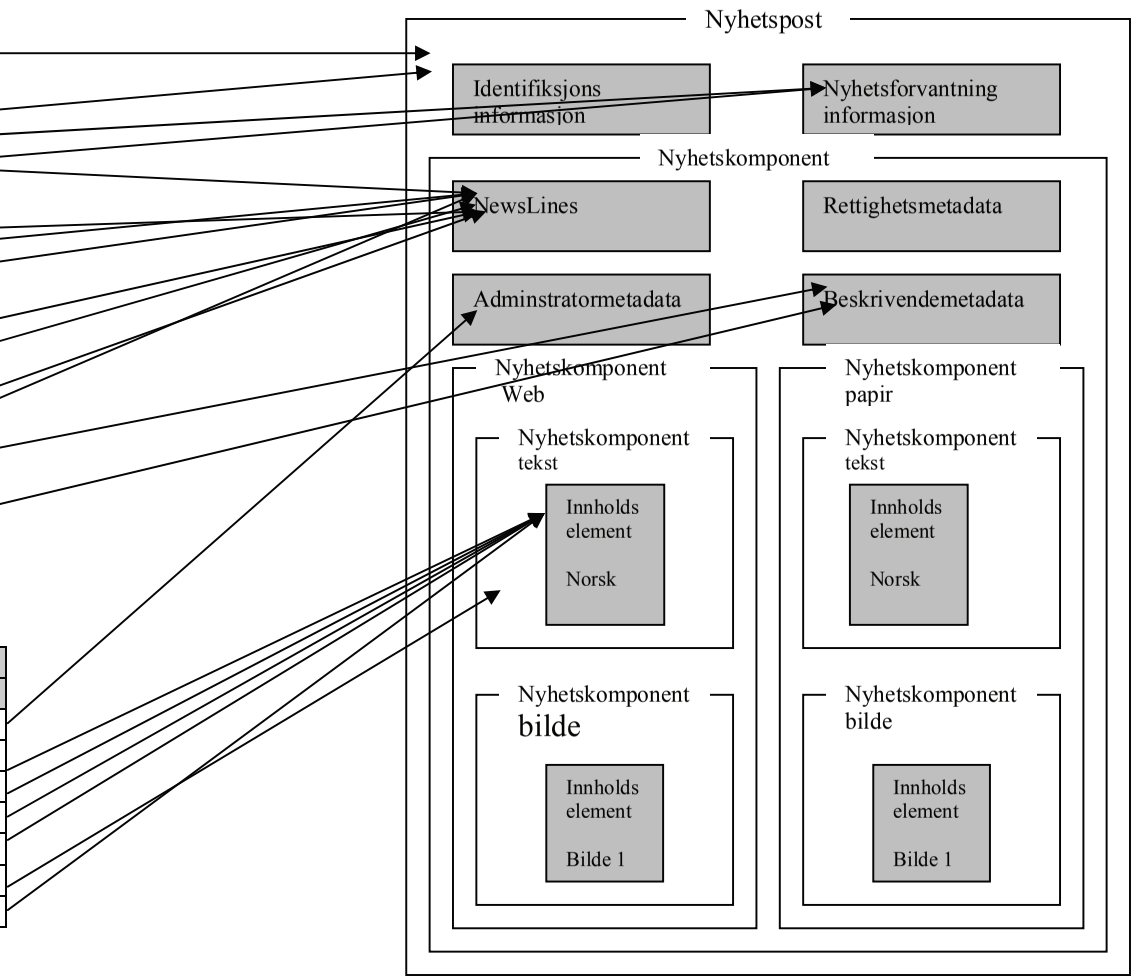
Hvis en ser på figuren her vil en se hvordan NewsML kommer til sin rett. Her kan mesteparten av metadata som skal legges inn trekkes ut fra hovednyhetsposten eller den felles nyhetskomponenten.

Når det gjelder vedlegget til internettartikkelen må det meste lagres lokalt i dette innholdselementet. Dette er fordi metadata her er helt spesifikke for dette vedlegget.

Rekkefølgen på metadatafeltene i tabellen er endret i forhold til kapittel 4, som er utskrift fra Adresseavisen sitt system, dette for å lette lesbarheten i figuren.

Artikler i internettdatabasen	
Metadata navn	**NewsML-element**
Artikkelref	Dokid
Kommentarer	Comment
Serredato	StatusWillChange.
Status	Status
Tittel	Title
Undertittel	Subheadline
Byline	Byline
Ingress	NewsLineText
Brødtekst	Slugline
Forsidetittel	Headline
Emneord	Keywordline
Vedlegg	Slugline, her med hyperlink
Kategori	Genre
Temakategori	SubjectCode
Publisert dato	Dekkes ikke direkte
Malfil	Dekkes ikke direkte
Forsidetekst	Dekkes ikke direkte

Artikkelvedlegg i internettdatabasen.	
Metadatanavn	**NewsML-element**
Filnavn	Filnavn
Vedleggsnummer	Dokid
Vedleggstype	MediaType
Bredde	Characteristics
Høyde	Characteristics
Byline	Byline
Tekst	NewsItem

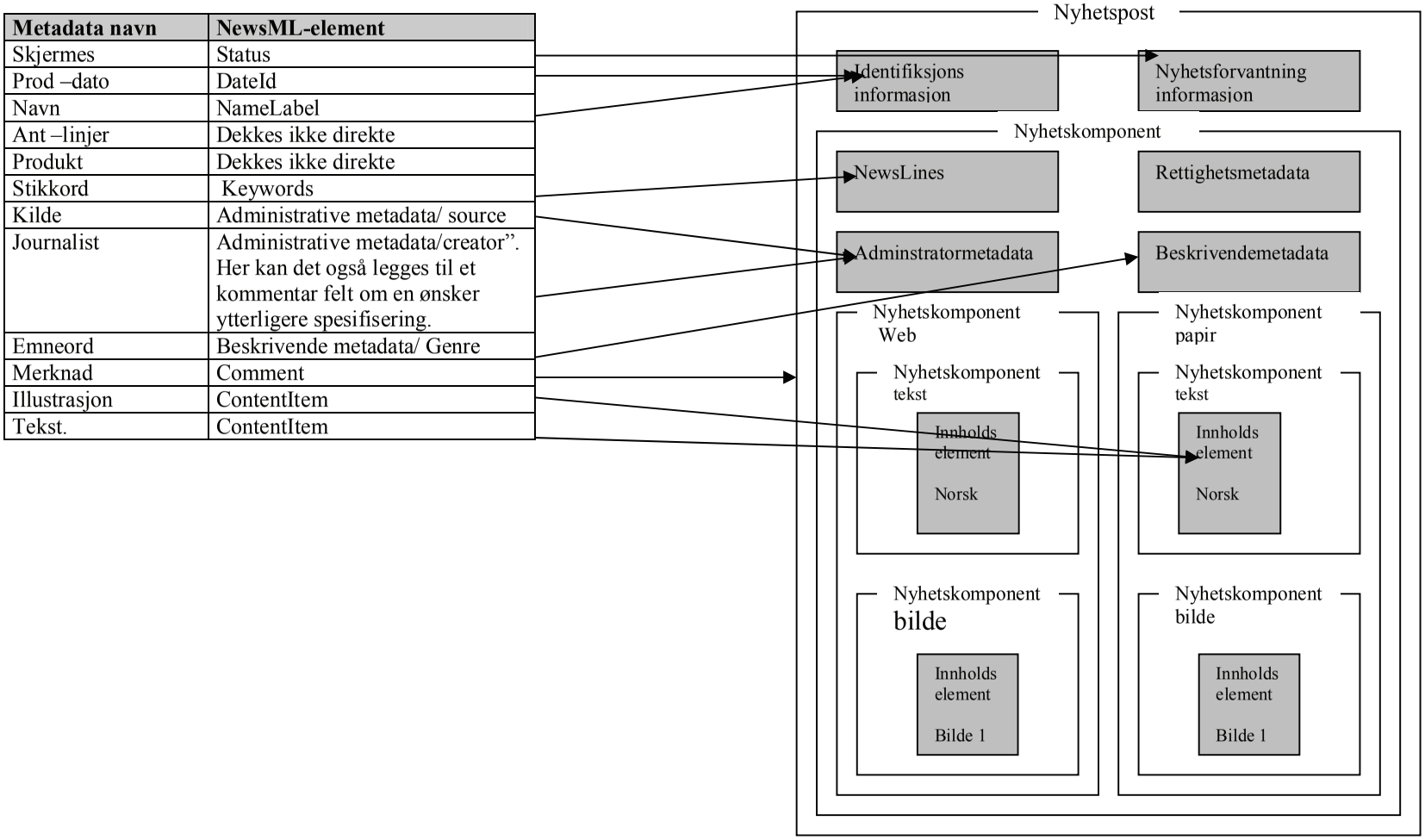


Figur 11 Artikler i internettdatabasen

6.3.3 Indeksering av artikler for papirutgaven

Hvis vi ser på figuren på neste side vil en se at en del av de samme feltene går igjen her, som for artiklene for internettavisen. Logisk sett er ikke dette så rart, og NewsML støtter da at en kan lagre metadata kun et sted.

Metadata navn	NewsML-element
Skjermes	Status
Prod –dato	DateId
Navn	NameLabel
Ant –linjer	Dekkes ikke direkte
Produkt	Dekkes ikke direkte
Stikkord	Keywords
Kilde	Administrative metadata/ source
Journalist	Administrative metadata/creator". Her kan det også legges til et kommentar felt om en ønsker ytterligere spesifisering.
Emneord	Beskrivende metadata/ Genre
Merknad	Comment
Illustrasjon	ContentItem
Tekst.	ContentItem

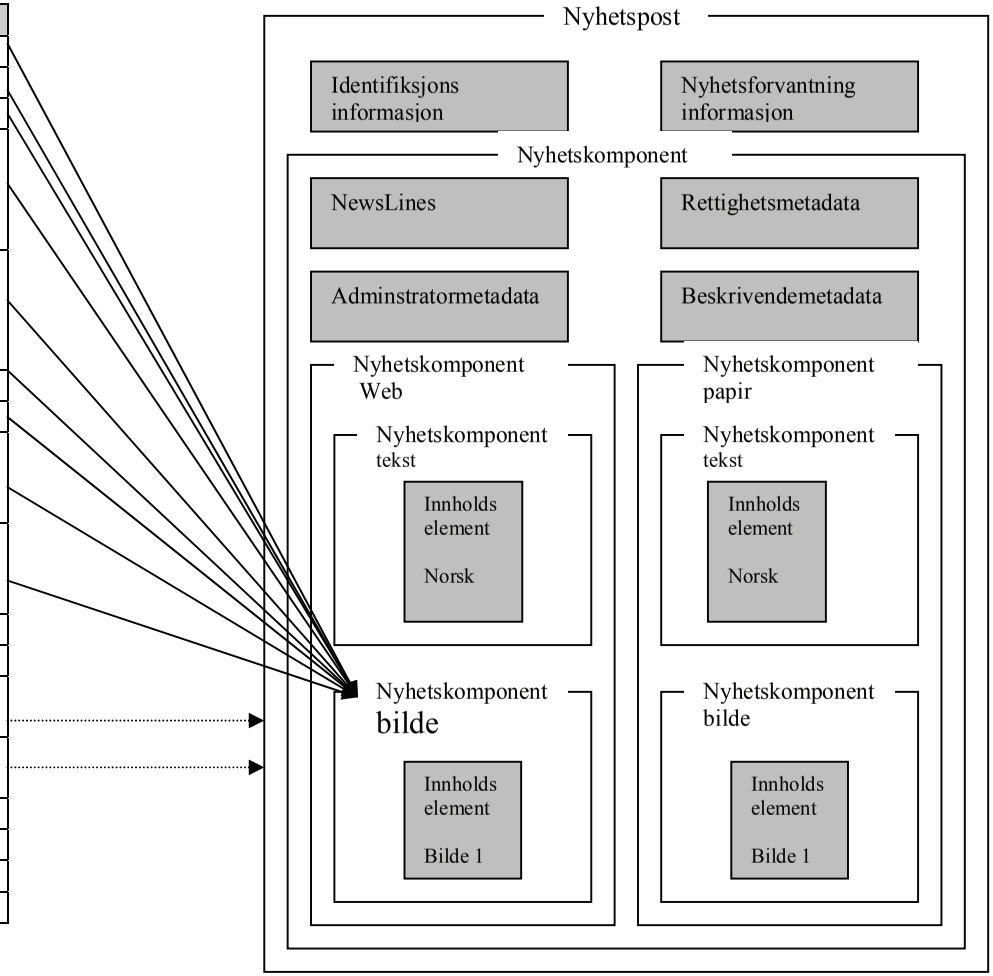


Figur 12 Metadataskjema for indeksering av artikler.

6.3.4 Indeksering av filmarkivet og illustrasjoner

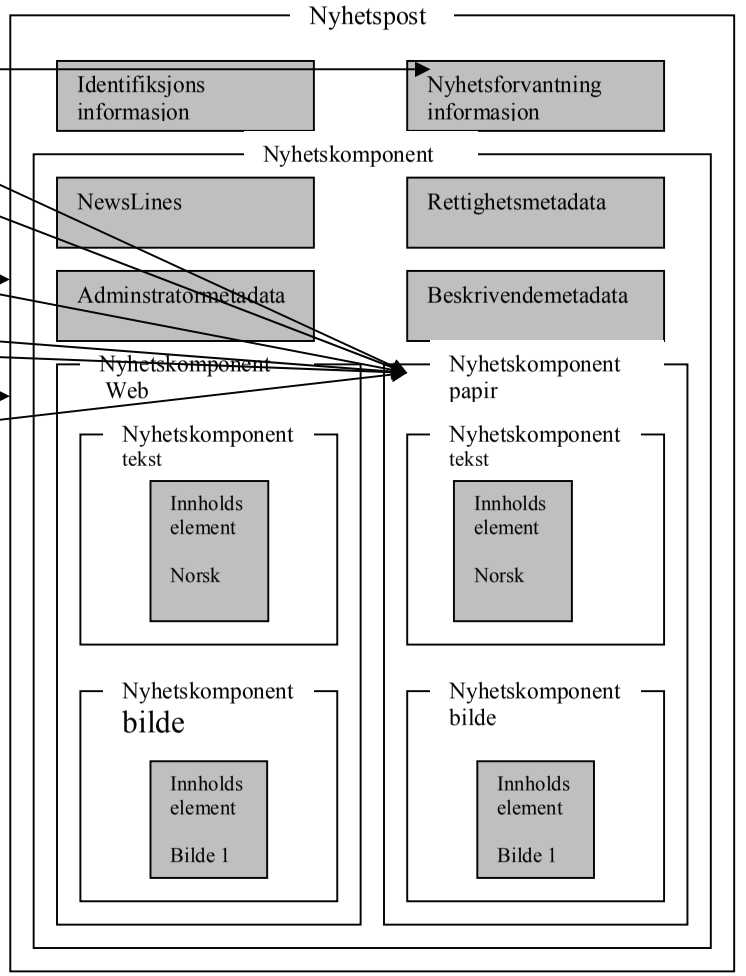
Denne blir litt spesiell siden NewsML ikke direkte håndterer fysiske objekter. Men en kan likevel lage til en slags mapping, ved å la den fysiske filmen bli et eget nyhetskomponent. I forhold til presentasjonen i kapittel 4 er rekkefølgen endret for å gi bedre lesbarhet av figuren.

Metadata Navn	NewsML-element
Jobb-id	Duid
Film-nr	Euid
Film-dato	Nyhetsforvaltning/FirstCreated
Fotograf	Administrative metadata/creator". Her kan det også legges til et kommentar felt om en ønsker å spesifisere fotograf
Journalist	Administrative metadata/creator". Her kan det også legges til et kommentar felt om en ønsker å spesifisere journalist.
Gruppe	Administrative metadata/ source
Merknad	Comment
Produkt	Dekkes ikke direkte men en kan bruke; Beskrivende metadata/property"
Motiv	Dekkes ikke direkte men en kan bruke "Beskrivende metadata/property"
Film-id	Dekkes ikke direkte
Prod-dato	Dekkes ikke direkte
Sak	Vil komme fram fra hele nyhetsposten
Brukt-dato	Vil komme fram fra hele nyhetsposten
Retur	Dekkes ikke direkte
Retur-adr	Dekkes ikke direkte
Bilde-nr	Dekkes ikke direkte
Side	Dekkes ikke direkte



Figur 13 Metadataskjema for indeksering av fysisk film.

Metadata navn	NewsML-element
Prod-dato	FirstCreated
Ill-id	Duid
Illistrasjon	Beskrivende metadata/ genere
Ill-type	Dekkes ikke direkte
Antal	Dette vil komme fram ved at vi har flere innholdselement i samme nyhetsposten
Opphav	Administrative metadata/ creator
Sak	Vil komme fra hele nyhetsposten
Gruppe	Administrative metadata/ source
Merknad	Comment
Retur	Dekkes ikke direkte
Brukt-dato	Vil komme fram fra nyhetsposten
Produkt	Dekkes ikke direkte men en kan bruke Beskrivende metadata/property"
Side	Dekkes ikke direkte



Figur 14 Metadataskjema for indeksering av illustrasjoner.

6.4 Oppsummering

Som vi ser av figurene lar de aller fleste metadatapostene Adresseavisen bruker pr. i dag seg mappe direkte inn i strukturen til NewsML. NewsML har tatt høyde for at ulike aviser bruker ulikt metadataformat og ønsker å beholde dette ved en eventuell overgang til NewsML, det er derfor mulig ved alle hovedelement og legge til egne element som beskriver det som ikke er dekket av de øvrige.

Kapittel 7 Applikasjon med NewsML

For å vise hvordan NewsML virker gjennom hele prosessen i en avis har jeg laget en prototyp. Jeg har valgt å bruke Norges Astma og Allergiforbund (NAAF) ”internettavis” som vil være mitt ”case”. De publiserer noe eget materiale, men de bruker også en del artikler fra NTB eller andre aviser. Gjennom en slik applikasjon, vil jeg vise fordeler og utfordringer ved NewsML som format.

7.1 Systeminformasjon

Før man kan begynne å programmere må man se på hva man egentlig skal lage. Hvem er programmet rettet mot, hva slags tjenester skal det ha og lignende. I dette delkapitelet beskrives dette kort.

7.1.1 Målgruppe

Prototypens formål er å forbedre og organisere NAAF sin nyhetsbase. Den skal primært sørge for laging, og framvisning av nyhets stoff i NAAF sin internettavis. Utvelgelsen av artikler som skal brukes, må til en stor grad gjøres manuelt for å få en redaksjonell vurdering av hva som er interessant. En kan ikke forutsette at alle i brukere i forbundet er vant til å bruke datamaskiner og internett, prototypen må derfor ha et enkelt brukergrensesnitt. Nettavisen skal kunne leses gjennom en helt vanlig nettleser, og brukerne skal ikke trenge å laste ned noen spesielle programmer for å lese avisen.

Et slikt system vil ha 3 typer brukere. Den vanlige sluttbruker som leser artiklene som er lagt ut, leter etter informasjon som er relevant for deres situasjon. Det vil være behov for en redaktør som sorterer innkommet materiale, og velger hva som skal presenteres i nettavisen. Det vil være behov for en systemadministrator som har ansvar for servere, databaser og nettavisen.

- Sluttbrukeren. En vanlig sluttbruker er en person mellom 10 – 70+ år. Felles for disse menneskene er at de søker informasjon om astma, allergi eller eksem. En kan ikke forutsette at disse har noe datakunnskap fra før, derfor må layout være intuitiv og enkelt i bruke. En vanlig bruker som søker informasjon logger seg på www.naaf.no og ønsker å se etter informasjon om for eksempel en medisin, som ble inntatt i Blåresept ordningen i januar 2002. Han går inn i søkesystemet, og utfører et emneord søk. Dette kan enten være fra et kontrollert vokabular som er definert, som for eksempel medisiner, støtteordninger og lignende, eller som fri tekst. Den delen med det kontrollerte vokabularet vil fungere som en katalogtjeneste, der han kan bla seg igjennom for å finne rett kategori der han regner med at det han ønsker å finne ligger. Målet ved et slikt system er at en vanlig sluttbruker ikke skal trenge å sette seg inn i hvordan et slikt søkesystem fungerer, en må gjøre søkebildene så intuitive at de aller fleste brukere skjønner hvordan de skal gjennomføre et godt søk. De menneskene som bruker nettavisen og portalen ønsker hovedsakelig å finne informasjon om blant annet:
    - Hvordan sykdommen fungerer og hvordan den kan behandles

    - Hvor en kan få tak i mat som de kan spise, og hva som kan erstatte matvarer de ikke tåler
    - Oppskrifter
    - Lover og regler som er relevant i forhold til sykdommen, dette inkluderer støtte ordninger for kosttilskudd, støtte til medisiner som ikke er dekket av blå resept ordningen, muligheter for å få tilrettelagt undervisning i skolen, muligheter for å få hjelpemidler gjennom skole og hjelpemiddelsentralen.

- Redaktøren. En bruker som sitter i redaksjonen, er en person som kan mye om sykdommen, mye om støtteordningene som finner, men som ikke nødvendigvis har så mye datakunnskap. Dette gjør at denne applikasjonen må være enkel å bruke. I en slik organisasjon er det heller ikke rom for å ha egne personer som sitter å tilføye metadata, slik adresseavisen har, men alt må gjøres når artiklene legges inn i systemet. Systemet må derfor gjøre det enkelt å legge inn nødvendig informasjon.

- Systemadministrasjon. Systemadministrasjonen vil være de som drifter systemet. Det kan også være ønskelig at den/de som gjør dette også har mulighet til å legge til felt i applikasjonen hvis det er ønskelig. Det er en forutsetning at den som skal vedlikeholde og drifte denne applikasjonen har noe kunnskap om Java, og setter seg inn i NewsML.

7.1.2 Tjenester og samlingstyper

Det er mange slags tjenester en slik applikasjon potensielt kunne ha. Relevant for denne prototypen kan være:

- Online aksess til objekter (semantisk innhold) i samlingene presentert i brukerens nettleser. Alle brukere har tilgang til alle artiklene, uten noen form for tilgangs kontroll. Systemet vil heller ikke kryptere filene som overføres i noen ledd. Dette er det ingen behov for, siden alt informasjon som ligger på sidene er tilgjenglig for alle uten verken betaling eller registrering av brukere.
- Mulighet til å søke etter informasjons ressurser etter ulike kriterier slik som forfatter, type ressurs, dato for innleggelse og lignende. I en slik portal bør vanlige brukere ha tilgang til å søke i systemet gjennom en helt vanlig nettleser.
- Muligheter til å legge inn, og slette filer i systemet.
- Hjelpefiler til alle tjenestene, som enkelt forklarer hvordan tjenesten brukes.

Applikasjonen skal kunne formidle alle typer data. Konkret kan dette være:

- Artikkel samlinger, her vil de faglige sykdomsrelaterte artiklene ligge. Dette vil være et bibliotek der brukerne kan finne faglig god veiledning om hvordan takle ulike problemer med sin sykdom. Her vil det også ligge informasjon om støtteordninger, og hvor og hvordan søke på disse.
- Avis artikler. Her vil dagsaktuelle artiklene fra NTB, og andre aviser ligge. En vil også ha et søkesystem som gjør at en kan søke etter gamle artikler, eller tema, forfatter, personer i artikkelen osv.
- Video av foredrag som er holdt om relevante emner.

- Mulighet for å legge inn spill som kan være en del av opplæring opplegget for de yngste.
- NAAF statutter og regler.

7.1.3 Innhenting og utvelgelse av informasjon

En organisasjon som NAAF har lite midler til å lage mange egenproduserte artikler og baserer, ser derfor noe på gjenbruk av artikler fra NTB eller andre aviser. Et system som skal støtte gjenbruk av artikler, bør kunne ta i mot den elektroniske konvolutten som brukes av NTB, Information Interchange Modell (IIM)[40], med dens tilhørende tekst i IPTC 7901 eller NIFT format, se kapittel 3. En skal også kunne legge inn stoff manuelt, altså lage den fra bunnen av. Videre kan man tenke seg et ideelt system som "overvåker" NTB og aviser, henter ut aktuelle artikler og presenterer de i et eget system for redaksjonell godkjennelse, og ved et tastetrykk legges de i den offisielle databasen som styrer visningen på internettavisen. Et slikt system bør i tillegg ha en enkel mulighet til å legge inn ny artikler og det bør være greit å søke i et slikt system.

Det finns et system som overvåker og henter informasjon hos andre kilder. RSS, *Real Simple Syndication*, et enkelt system for å la ulike informasjonskilder levere sine varer i standardisert XML-format. En RSS-kanal (også kalt «feed») kan for eksempel bestå av nyhetsoverskrifter og ingresser, aktuelle tilbud fra forhandlere, eller annen regelmessig oppdatert informasjon[41]. Artikler som ligger på internettaviser innholder ofte lite eller ingen metadata, og derfor vanskelig å legge rett i en NewsML-database.

Det er to måter å løse dette på slik jeg ser det. Den enkleste måten rent teknisk, er å gjøre arbeidet manuell ved å la personer legge til metadata. Dette er et tidkrevende arbeid, da personen først må få god oversikt over artikkelen som skal beskrives. Den andre løsningen jeg ser på et slikt problem er å lage et system som leser igjennom artikkelen og ved hjelp av ulike thesauri plukke ut relevante metadata fra teksten, og legge de til en NewsML-fil. Det er vanskelig å lage et slikt system som fungerer godt, og det ligger utenfor denne oppgaven.

7.1.4 Avgrensing i prototypen

Å utvikle et komplett system som tar for seg alle deler av NewsML og støtter alle systemkravene skissert over er en stor oppgave. God brukerdesign er meget tidkrevende, og det forutsetter at du har god kunnskap til brukeren, både sluttbruker og kunde (her NAAF). Jeg har derfor valgt å fokusere på følgende punkter i min prototyp:

- Målgruppe: Prototypen tar kun for seg administrasjonsdelen av systemet som er beskrevet. Men hvis en har laget et godt brukergrensesnitt som en ønsker å bruke, kan en gjenbrukekoden i administrasjonsdelen for å få tilgang til dataene. Dette kommer jeg noe tilbake til senere.
- Tjenester og samlingtype. Administrasjonsdelen i systemet vil ha mulighet til å søke etter informasjons ressurser etter ulike kriterier slik som forfatter, type ressurs, dato for innleggelse og lignende og muligheter til å legge inn, endre og slette filer i systemet.
- Innhenting og utvelgelse av informasjon: Også her er det lagt enkelte begrensninger, prototypen kan ta inn eksisterende xml filer, som for eksempel

kommer fra nyhetsbyrå som bruker NewsML-formatet. I prototypen kan man lage nye NewsML-filer, og søke blant filene i databasen. Hvis en ønsker å lagre en nyhetsfil som en henter for eksempel fra NTB, må en lage denne som en ny artikkel for å få med alle metadata.

7.2 Teknologi i implementasjonen

Jeg har valgt å bruke programmeringsspråket Java til å implementere min prototyp. Alle komponenter som er hentet utenifra og alle egen lagde komponenter er skrevet i Java[42]. Valget falt på Java da dette er et programmeringsspråk som er mye brukt på ulike systemer i datamarkede ellers, og som jeg ønsket å lære meg. XML[43] filene transformeres gjennom XSLT(Extensible Stylesheet Language Transformations) med en XSL (Extensible Stylesheet Language) fil til HTML som kan vise artiklene gjennom en vanlig nettleser[44].

Som lagringsmedium for NewsML-filene har jeg valgt å bruke XML databasen Exist[45]. Det er mange xmldatabaser å velge mellom. Mitt valg kom på Exist, da dette er en er fullt funksjonell, Java basert database som støtter fulltekst søk, XQuery med mer, og den er gratis[46]. XML database er brukt også i større systemer, uten at det her er gjort en vurdering av type database for et stordriftsystem. En organisasjon som NAAF vil sikker kunne ta i drift prototypen med mindre endringer.

Jeg har også brukt en Javabasert relasjonsdatabase, Derby, til å lagre objektene som tilhører NewsML dokumentene. Derby er en liten database, implementert i Java, som er gratis, levert av Apache[47]. Den fungerer godt i denne applikasjonen. Å eventuelt skifte database er enkelt hvis en først har en database tilgjenglig. Alt som må skiftes er noen få linjer med driver og oppkoblings informasjon i koden som ligger i en egen fil. Slik databasen i prototypen er satt opp er Derby basen i seg selv ikke søkbar, men XML-filene som returneres fra et søk i Exist databasen vil innholde en referanse til Derby databasen.

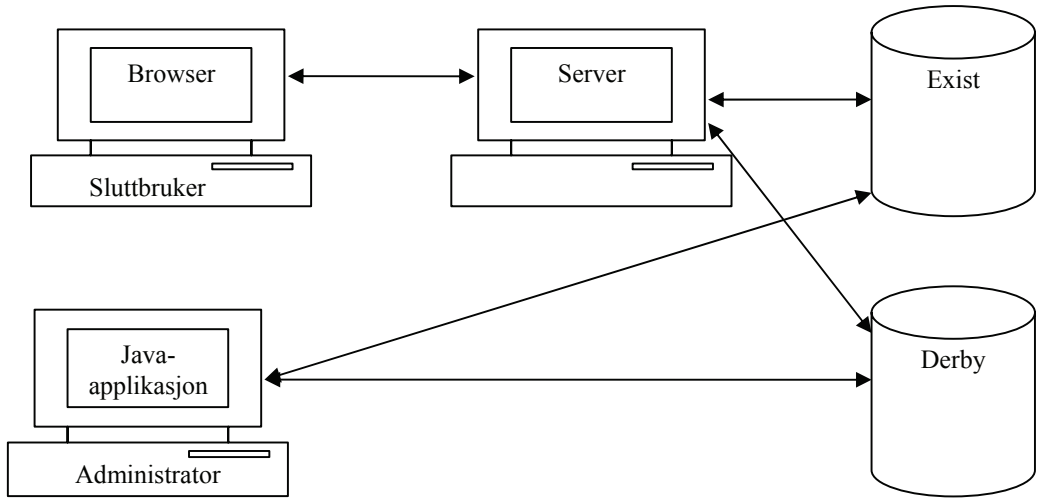
Både Oracle og IBM DB2[48] har støtte for hierarkisk datalagring, som XML er, i sine relasjonsdatabaser. Ved bruk av en av disse basene vil en database være nok. Har man ikke tilgang til en slik database er det også mulig å lagre innholdselement i Exist databasen, ved hjelp av blobs (”binary large objekt”), eller uspesifisert data håndtert på bit nivå).

Jeg har ikke prioritert å bruke tid på datagrunnlaget i denne prototypen. NewsML-filene som ligger i databasen er fiktive, og kun brukt til testing. Begrunnelsen for denne prioriteringen er at det skal en stor samling filer før man får en god gjenfinningsrate, med mindre man vet hva som ligger der. Dette gjelder også for den relativt begrensede datasamlingen jeg har valg at mitt system skal brukes til.

7.3 Arkitektur

Systemet har en klassisk trelags arkitektur, med klient, server og database. Sluttbrukeren får tilgang til systemet gjennom nettleser, mens administrator kan legge inn filer i en Java applikasjon. Serveren overfører data fra databasene til sluttbrukens nettleser. Administrasjonsklienten kommuniserer selv med databasene, en XML database der

NewsML-filene ligger lagret, som gir søketilgang og en relasjonsdatabase som lagrer innholdselementene (bilder, film, tekst med mer).



Figur 15 Arkitektur for prototypen

7.4 Prototypen

I prototypen er alle NewsML-elementene er implementert. I grensesnittet blir kun de NewsML-elementene som er ansett som viktige for prototypens brukergruppe, og strukturen for NewsML-dokumentet tatt med. Dette er gjort gjennom en avveining av de ulike elementene i forhold til nytte og i forhold til å prøve å holde applikasjonen brukervennlig. Ønsker man å koble opp flere NewsML-element til brukergrensesnittet, vil man kunne bruke de ferdige klassene.

7.4.1 Starte programmet

Når du åpner programmet vises dette skjermbilde.

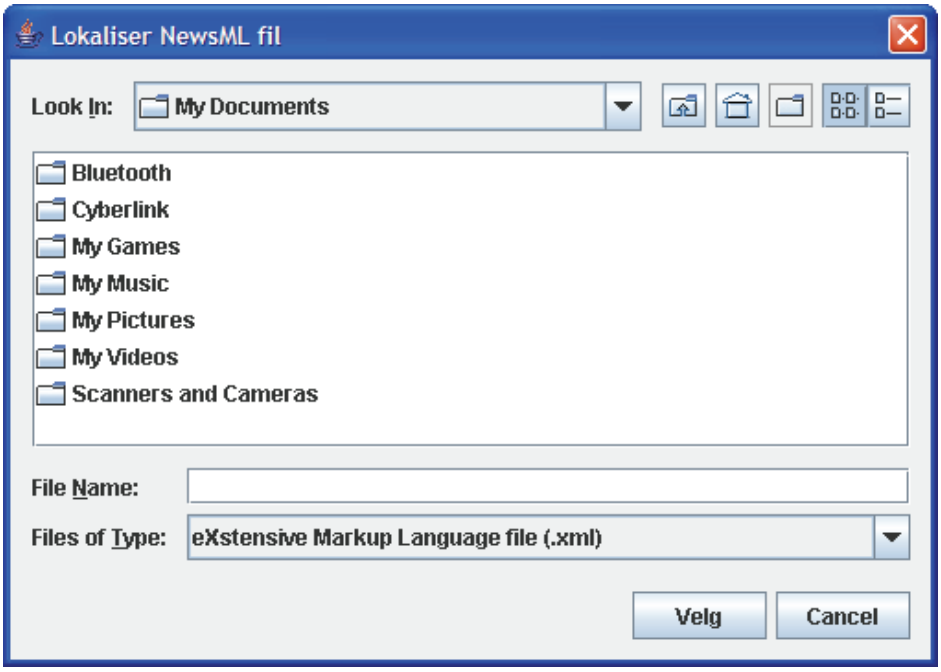
Vi ser her at vi har tre ark å velge mellom.

- Legg inn artikkel
- Lag ny artikkel
- Søk

Figur 16 Prototypen, start skjermbilde.

7.4.2 Legg inn artikkel

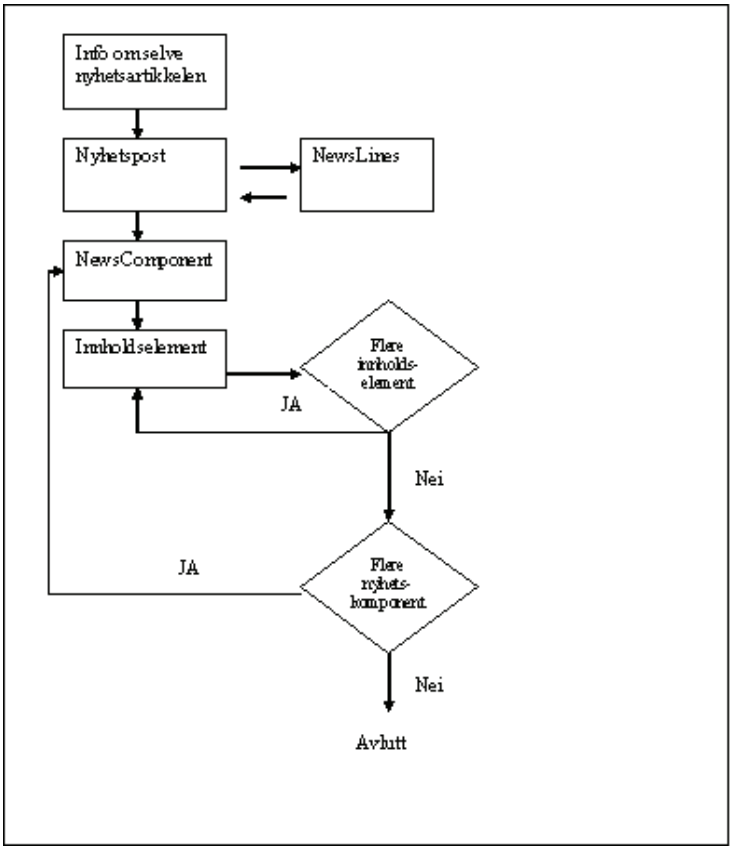
Her er det tenkt at hvis en alt har NewsML-dokument, som man for eksempel har hentet ned fra et nyhetsbyrå, som skal inn i samlingen, så kan en laste den inn her. Den kan også ta imot andre xml dokument også, men disse vil ikke bli like godt søkbare senere da de mangler mye informasjon. Når en trykker på ”*browse*”, får en opp filsystemet. Den lokale ”mine dokumenter” mappen er satt som standard startpunkt.



Figur 17. Prototypen. Velg NewsML-fil.

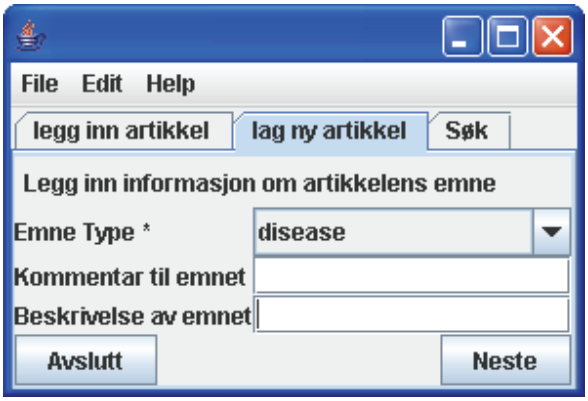
7.4.3 Lag ny artikkel

Her ligger selve kjernen i systemet. Det er her du legger inn nye artikler. Gjennom hele applikasjonen vil elementer merket med * være obligatoriske og disse må være fylt ut før en får gå videre. Gangen i systemet, og elementene er satt opp slik:



Figur 18 Prototypen. Illustrasjon over gangen i ”legg inn ny artikkel”.

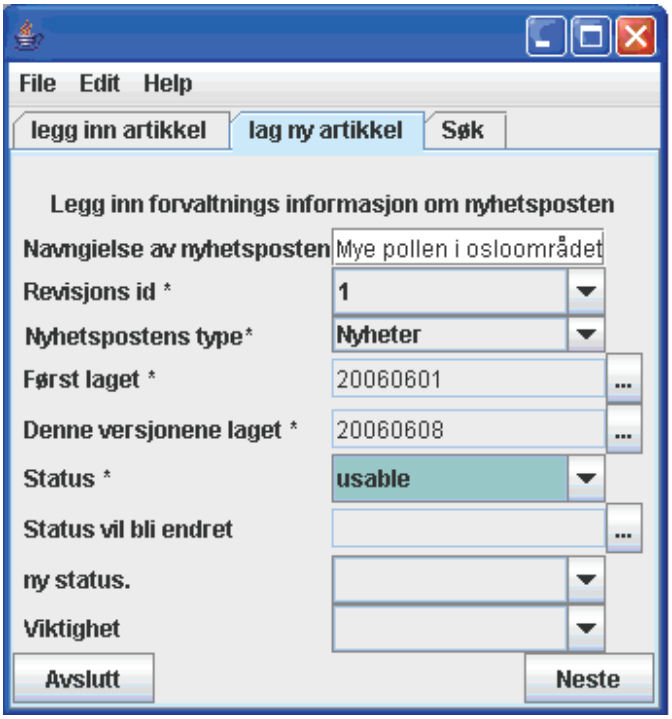
Det første skjermbildet ber deg om å legge inn informasjon om emnet til artikkelen.



Figur 19 Prototypen, emneinformasjon

- Emne Type: er en instans av ”*topicType*” i Topic elementet, som forteller hvilken type emnet er av. Her finnes det en rullgardin meny der brukeren kan velge mellom ulike alternativer. Hva slags verdier man velger her står en i NewsML helt fritt til å velge, men jeg har valgt å bruke IPTC sine standardiserte emnekoder (”*subject codes*”), som beskrives i punkt 3.2.1. Jeg har valgt å ikke endre på disse, da jeg ønsker størst mulig fleksibilitet på applikasjonen og gjøre det mulig å kunne ta imot alle typer NewsML-dokumenter. Jeg kunne valgt å ha utvidet vokabularet, men i forbindelse med denne applikasjon ser jeg ingen behov for dette da IPTC sitt vokabular er meget utfyllende. I selve implementeringen har jeg trukket ut de som er mest interessante for domenet som skal produseres. Implementeringen vil allikevel kunne benytte seg fult ut av hele IPTC sine emnekoder, men når en skal opprette en ny artikkel er det de elementene jeg har ansett som sentrale som brukeren har å velge mellom.
- Kommentar til emnet: er en instans av ”*comment*” elementet i ”*Topic*”. Her kan bruker legge til tilleggs informasjon i naturlig språk.
- Beskrivelse av emnet: er en instans av ”*description*” elementet i ”*Topic*”. Her kan brukeren beskrive emnet ytterligere.

Ved å trykke på ”neste” går en videre i innleggingen. Da kommer en til dette skjermbildet:



Figur 20 Prototypen, legg inn forvatningsinformasjon

Her legger en til informasjon som etter hvert skal bygge opp elementene ”*identification*” og ”*NewsManagement*”.

Navngivelse av nyhetspost: ”*nameLabel”* - En streng som kan identifisere nyhetspostens navn for mennesker. Kan ikke lages på flere språk.

Revisjons ID: ”*revisionId*” - Elementet skal innholde et positivt heltall som forteller hvilket revisjonsnummer en nyhetspost har. Den nyeste instansen av en nyhetspost må alltid ha det høyeste revisjons-id’en.

Nyhetspostens type: ”*newsItemType*” - Angir hvilke type en nyhetspost er av. Verdiene er hentet fra et kontrollert vokabular. Dette bestemmer redaktøren selv, og er i prototypen satt til følgene verdier:

- Nyheter
- Medisiner
- Astma
- Allergi
- Mat
- Pasientrettigheter
- Behandling/rehabilitering

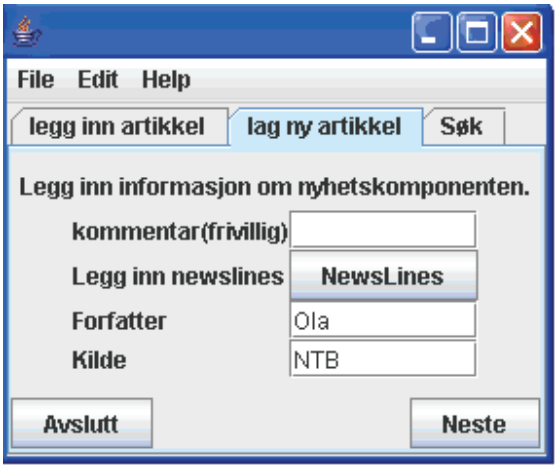
Først laget: ”*firstCreated*” - Angir dato, og frivillig tiden som nyhetsposten først ble laget. Formateres automatisk til et ISO 8601 Basic Format, som NewsML krever. Denne versjonen laget: ”*thisRevisionCreated*” - Angir revisjons dato for denne spesifikke nyhetsposten. Formateres automatisk til et ISO 8601 Basic Format, som NewsML krever.

Status: status - Angir nyhetspostens anvendelighet. Verdiene i dette elementet er hentet fra et kontrollert vokabular. IPTC har definert et ”*topicset*” som en kan bruke. Dette består av følgende status verdier: ”*usable*”, - som forteller at nyhetsposten er uten restriksjoner. ”*Embargoed*”, - som er at nyhetsposten og dens innhold er stengt for offentliggjøring inntil den blir godkjent for dette av utgiver, ”*Withheld*”, - nyhetsposten eller dens innhold er ikke klar for utgivelse. ”*canceled*”, - Verken nyhetsposten eller dens innhold skal under noen omstendigheter offentliggjøres.

Ny status og Status vil bli endret: lager sammen ”*futureStatus*”. Status elementet angir når nyhetsposten automatisk kommer til å bli endret. Her angir en under elementene, ”*FutureStatus*”, som angir ny status og ”*DateAndTime*”, som angir dato for endringen. Elementet er frivillig.

Viktighet: ”*urgency*” - Angir hvor viktig nyhetsposten anses for å være. Verdiene skal hentes fra et kontrollert vokabular. Her har jeg bare for enkelhets skyld satt opp verdiene 1 til 5 der 1 regnes som mest viktig.

Ved å trykke på ”neste” igjen, går brukeren egentlig fra å gi informasjon om ”*newsItem*” til en ”*newsComponent*”. Men siden systemet er lagt opp slik at brukeren ikke skal kunne noe om NewsML og jeg har derfor valgt å ikke markere det i noe annet enn i en endring i ledeteksten. Informasjonen i dette skjermbilde brukes til å lage AdminstrativeMetadata, som er en del av ”*NewsComponent*”



Figur 21 Prototypen, innlegging av informasjon om nyhetskomponenten

Kommentarfelt: ”*comment*” - Kommentar element om en har noe ekstra å si.

Legg til ”*newslines*”: Åpner eget skjermbilde, se under.

Forfatter: ”*creator*” - String som identifiserer hvem som har laget nyhetsobjektet, kan være et individ og/eller en bedrift og/eller en organisasjon. Elementet er frivillig, og kan kun forekomme 1 gang.

Kilde: ”*source*” - URL som identifiserer kilden som har tilført kilde materialet til nyhetsobjektet.

Ved å trykke på ”*NewsLines*” får en dette skjermbildet, der en legger inn standard metadata til en artikkel, og som danner ”*NewsLines*” elementet som også ligger under ”*NewsComponent*” elementet

Figur 22 Prototypen, legg til nyhetslinjer

”*headLine*” - angir synlige overskriften.

”*subHeadLine*” - angir synlige ”under-overskrifter”.

”*byLine*” - angir i naturlig språk informasjon om journalist/forfatter.

”*creditLine*” - angir i naturlig språk kreditt informasjon.

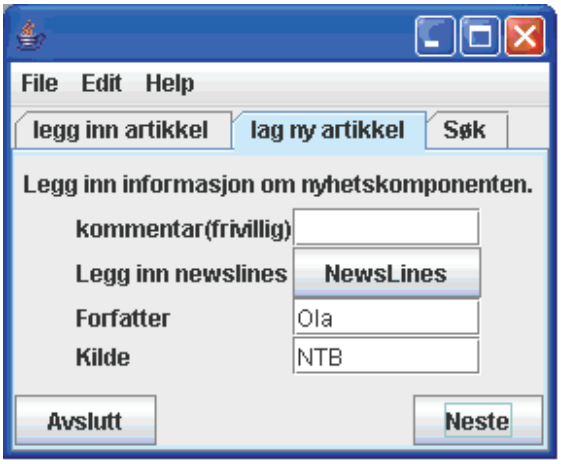
”*rightsLine*” - angir den visbare versjonen av rettighetsinformasjonen. Altså om hvem som har lov å bruke objektet og hvordan det da skal brukes.

”*seriesLine*” - angir den visbare informasjonen om objektets plass i en ev. serie.

”*slugLine*” - angir en streng av tekst (ev. med en hyperlink) som brukes til den visbare ”slug-line”. Hva som legges i ”slug-line” er opptil den enkelte redaksjon.

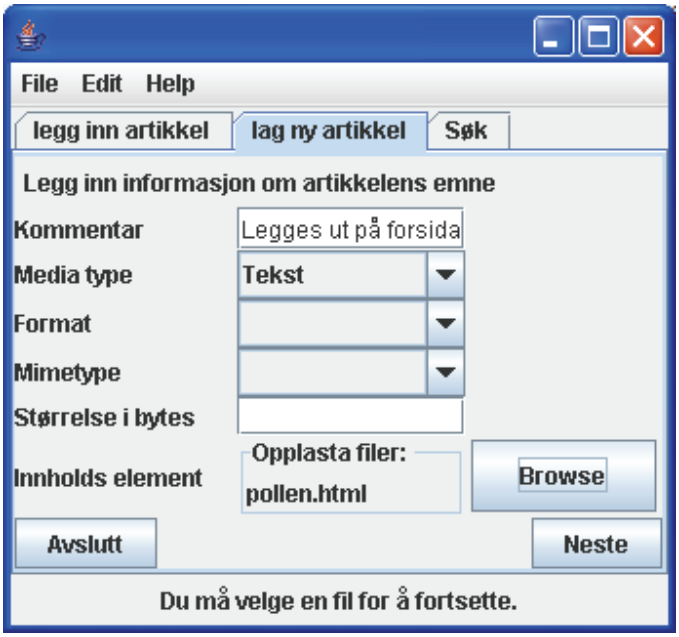
”*keywordLine*” - angir den eller de visbare nøkkelordene som er relevant for nyhetsobjektet.

Når en trykker på ”neste” her kommer en tilbake til det skjermbildet en var på først:



Figur 23 Prototypen, innlegging av informasjon om nyhetskomponenten

Her har man mulighet til å legge inn flere NewsLines om man ønsker det. Når en så igjen trykker på ”neste” kommer en til det siste, men viktige skjermbildet i programmet. Det er her en faktisk gir artikkelen innhold. Dette bygger opp et ”*ContentItem*” som igjen går inn under ”*NewsComponent*” elementet.



Figur 24 Prototypen, legg til innholdsfil.

Kommentar: ”*comment*” - kommentar element.

Media type: ”*typemediaTypeFN*” - beskriver hvilken medietype innholdet er av.

Format: ”*formatFormalName”* - beskriver hvilket format den aktuelle innholdsposten er av.

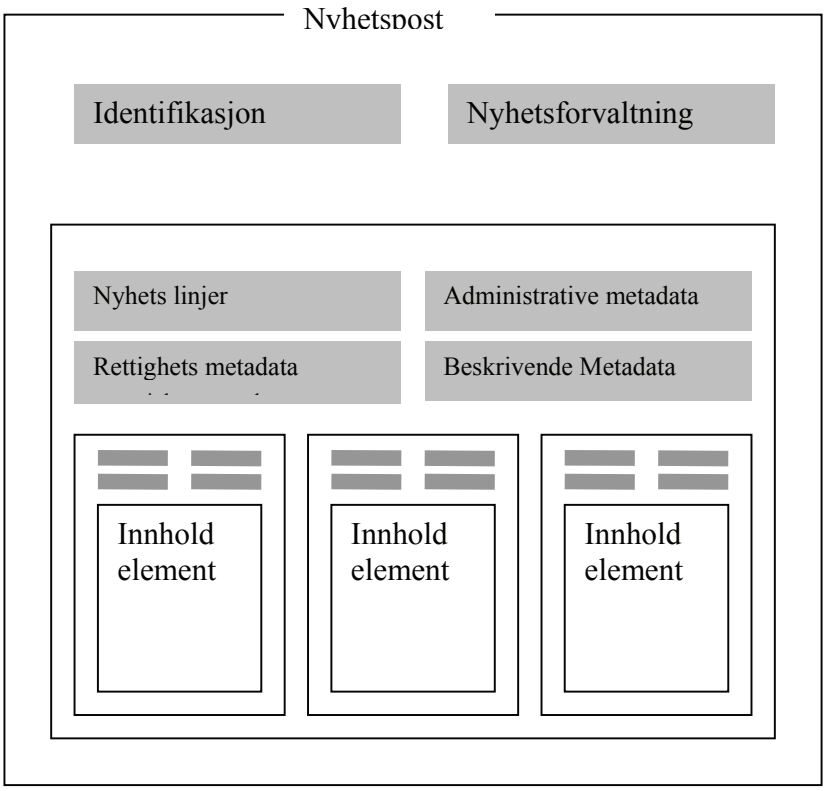
Mimetype: ”*mimeTypeFormalName*” - beskriver hvilken mimetype den aktuelle innholdsposten er av.

Størrelse i bytes: ”*sizeInBytes*” - Angir størrelsen på filen.

Innholds element: ”*data*” - Angir selve innholdselementet. Denne kan repeteres så mange ganger som ønskelig.

Figur 25 Protypen, valg om å legge inn flere nyhetskomponenter.

En NewsML nyhetspost ser slik ut:



Figur 26 En nyhetspost i NewsML

7.4.4 Søke i databasen

Det siste valget i prototypen er søk i databasen. Dette kan gjøres på to måter. Enten ved at en vet filnavnet på artikkelen en er ute etter, og skriver det inn i feltet bak ”søk etter filnavn”, eller ved at en søker i databasen etter metadata. Det gjøres ved at en velger et eller flere søkefelt en ønsker å søke på, ved hjelp av en rullgardin meny, og skriver inn søkeordet en ønsker. Resultatet får en opp som en liste. Ved å velge ”vis” som er ved siden av hvert enkelt resultat streng, vil en få vist artikkelen i en enkel html side.

Figur 27 Prototypen, søk.

Kapittel 8 Evaluering og veien videre

I denne hovedfagsoppgaven har jeg sett på ulike formater som kan brukes for lagring av data i en avis, videre har jeg beskrevet det systemet Adresseavisen har, men hovedfokus har vært på NewsML. For å vise tydeligere forskjellen på måten Adresseavisen i dag produserer sin avis og hvordan det kunne vært med NewsML har jeg valgt å illustrere med et eksempel. Jeg har tatt utgangspunkt i en artikkel med et bilde til papirutgaven, mens i artikkelen til internettutgaven er teksten litt forandret og artikkelen inneholder i tillegg to bilder og fire linker.

Jeg har valgt å se bort fra redigeringsfasen der journalist og ansvarlig utgiver/redaktør sender artikkelen fram og tilbake for finpussing, da dette bør skje i en skriveprosess uansett hva slags system en bruker. Jeg begynner å se på artikkelens gang først når den er ferdig redigert og klar for trykk/utlegging på nettavis. Det sentrale i denne sammenheng er gjenfinning av data som vises avslutningsvis.

8.1 Artikkel i Adresseavisen

Når artikkelen med bilder er ferdig og klar til trykk vil den i adresseavisen bli lagt i CCI systemet. Derifra sendes internett versjonen av artikkelen, med bildene og tilhørende metadata til internettdatabasen. Artikkelen som skal i papirutgaven av avisen går til trykk, og selve artikkelen blir sendt til stoffarkivet i SIFT. Hvis bildene er tatt med analogt kamera blir negativene lagt i filmarkivet under SIFT, mens den digitale utgaven av bildene som blir brukt i artikkelen blir sendt til Fotostation for lagring der.

De ulike delene til denne artikkelen blir altså lagret i 4 ulik databaser:

- Internettdatabasen
- SIFT databasen – Stoffarkivet
- SIFT databasen – Filmarkivet
- Fotostation

8.1.1Søk

For å finne igjen alle lagrede objekt i forbindelse med den aktuelle artikkelen må en søke i ulike databaser. Ved å søke på tittelen i internettdatabasen vil en få opp teksten til artikkelen skrevet for internett, og tilhørende bilder. Bildene som var i papirutgaven av artikkelen finnes i fotostation, her kan en for eksempel søke på dato for utgivelse kombinert med fotograf eller motiv. Når en finner rett bilde, vil også SIFT film-id referansen til negativen komme opp. Om en ønsker å finne negativene til bildene, kan en bruke dette nummeret til å søke i SIFT filmarkivet for å finne film-nr som indikerer hvor en kan finne de fysiske negativene. Det som gjenstår da er å finne teksten som ble brukt i avisen. Dette gjøres mot SIFT databasen, de ulike søkemulighetene er beskrevet i kapittelet om Adresseavisen. Problemet med søk i SIFT databasen oppstår når du skal lete etter en gammel artikkel, som du ikke husker når ble utgitt. SIFT databasen er organisert etter årstall så hvis du skal søke etter en artikkel fra 2000 må du først åpne ”mappen” for 2000 før du gjør søket ditt. Det er mulig å søke på flere årstall samtidig, men hver enkelt mappe må åpnes for seg først, - før en gjør søket.

8.1.2 Vurdering av metadataformatet til Adresseavisen

Min vurdering av metadataformatet til Adresseavisen er at det er virker unødig tungvint; journalistene skriver metadata til sine artikler, disse nyttiggjøres i liten grad. Når en artikkel kommer inn i systemet indekseres den av ulike mennesker og legges i ulike databasesystemer. Det at artiklene lagres i fire forskjellige databaser vil nok gjøre gjenfinningen vanskeligere enn om en kunne søkt i en felles database. Alle databasene har i tillegg ulike indekseringsformater, der samme post kan benevnes forskjellig. Jeg vil anta at dette lett skaper forvirring hos brukerne av systemet, noe som gjør at terskelen for å sette seg inn i systemene som eksisterer er større enn hva den kunne ha vært.

En ser også at de i forbindelse med illustrasjoner og film at de ikke indekserer motiv, eller hva filmen/illustrasjoner dreier seg om. På bakgrunn av dette er min vurdering at behovet for gjenbruk lite ivaretatt.

8.2 Artikkelen lagret ved hjelp av NewsML-formatet

Ved problematikken med ulike metadataformat for ulike nyhetsobjekter kommer en av de store gevinstene til NewsML til syne. Siden NewsML ikke tar hensyn til hva slags format som lagres eller hvor det skal brukes vil alle elementene her lagres som ulike objekter i en enkelt database koblet sammen med en fil som inneholder all metadata. Teknisk sett blir NewsML-filen og nyhetsobjektene kanskje lagret i to ulike databaser avhengig av hva slags system det blir implementert i, men dette er ikke noe brukeren trenger å ta hensyn til, da NewsML-filen inneholder lenker til databasen med objektene så brukeren vil aldri se at det er to fysisk atskilte databaser. Noen databaser har som nevnt i kapittel 7 mulighet til å lagre data både etter relasjons og hierarkisk modell.

8.2.1 Søk

For å finne hele artikkelen eller elementer fra en artikkel som er lagret i NewsML-format trenger en kun å søke i en database. Resultatet i denne databasen viser alle aktuelle objekter. Dersom man for eksempel søker på tittelen i artikkelen, vil du få opp teksten til internettartikkelen, teksten til papirutgaven, og alle bildene.

8.2.3 Fordeler med NewsML

En annen gevinst ved NewsML er at den omfavner hele prosessen ved å lage elektroniske nyheter. NewsML-dokumentet for en artikkel kan lages av journalisten som lager artikkelen, den kan videre brukes når en artikkel skal overførers fra et nyhetsbyrå til en kunde, og den kan videre brukes når avisen som mottar dokumentet skal vise den på nett, eller sende den til desken.

XML teknologien gjør det mulig på en enkel måte å konvertere alle data i en NewsML-fil til ulike typer dokumenter for eksempel PDF. Dette gjør igjen at det er mulig å lage visningsfiler for ulike visningsenheter som for eksempel websider, mobiltelefoner eller andre håndholdte enheter. I tillegg håndterer NewsML alle nyhetselementer som like objekter, noe som gjør at det enkelt kan håndtere alle mulige multimediatyper og støtte

ulike språk, med samme metadata, noe som gjør det meget fleksibelt og enkelt å søke i. NewsML har i tillegg en åpen standard på den måten at hver enkelt avis kan selv velge metadataformatet de ønsker å bruke, noe som gjør det enklere å innpasse i eksisterende systemer.

8.2.4 Problemer med NewsML

Så langt ser det meget bra ut. Gjennom mitt arbeid med prototypen ble det veldig tydelig at NewsML er komplekst. Det er i og for seg ikke uvanlig for et datasystem. Men for å dra nytte av kompleksiteten må også de som legger inn artiklene i systemet, altså journalistene, forstå oppbygningen av et NewsML-dokument. Det kan gjøres enkelt, enda enklere enn det jeg har gjort i min applikasjon. Da mister en mange av fordelene med NewsML, som for eksempel det å støtte flere språklige versjoner av en og samme artikkel, med samme metadata. IPTC arbeider i dag med NewsML versjon 2.0, som de tar sikte på å lansere på markede i løpet av 2006, den nye versjonen tar sikte på å forenkle bruken av NewsML.

NewsML har også den ulempen at det ikke er i utrakt bruk ennå. Siden verken NTB eller for eksempel svenske TT distribuerer artikler gjennom NewsML vil aviser som Adresseavisen ikke kunne nyttiggjøre seg artikler fra nyhetsbyråene direkte, og må da konvertere og sette sammen NewsML dokumentene selv. Selv om Reuters, og franske AFP distribuerer nyheter i NewsML format, kan likevel disse artiklene ikke brukes direkte på grunnen av språket. Selv om det finnes automatiske oversettere, fungerer ikke de bra nok til at disse kan brukes i en slik sammenheng.

Siden NewsML er designet for elektronisk prosessering av nyheter vil det likevel være behov for andre systemer ved siden av som håndterer for eksempel oppsett av selve nyhetssiden, og det å organisere negativ filmer. Integrert i et slikt system må det være støtte for den redaksjonelle produksjonen av avisen.

8.3 Sammenligning og vurdering

For å støtte gjenbruk av nyhetselementer bør en sørge for en enkel gjenfinning av objekter. I avissammenheng ser en ofte at det er mest bilder, illustrasjoner og lignende som blir gjenbrukt i trykk. Dette støtter både adresseavisen og NewsML formatet, da begge har kun en database å søke i, selv om Adresseavisens metadataformat kunne vært mer beskrivende i forhold til hva bilde/illustrasjoner handler om. Gjenfinning av tekstlig materiale er viktig for at journalister kan skaffe seg bakgrunnsinformasjon om det de skriver om. Å finne igjen fullstendige artikler med bilder, er komplisert i adresseavisen sitt system med søk i ulike databaser med ulikt metadata system. NewsML kan gjøre dette enklere ved at en kan gjøre ett oppslag i en database.

8.3.1 Utvikling av Adresseavisens sitt system

Slik jeg ser det bærer Adresseavisens sitt system preg av å ha blitt utvidet ved innføring av nye teknologi samtidig som man ønsker å fremdeles støtte gammel teknologi og rutiner. Dette gjør utførelsen av fullstendige søk kan ta mer tid enn optimalt, samtidig som systemet innholder noe dobbeltlagring av metadata. Adresseavisen fremstår i dag som en vanlig norsk avis med en fysisk papiravis og en nettavis, - alt på norsk. Om man

ønsker å utvikle et nytt system for Adresseavisen bør man se framover. Om noen år er det ikke utenkelig at Adresseavisen ønsker å utvide markedet sitt til å støtte ulike digitale enheter som kommer, støttevisning av artikler på flere språk, og både film og radio på sine internettsider.

Adresseavisen har ulike valg for å utvikle sitt system videre. Man kan ta i bruk NewsML sin nye måte å tenke på, og behandle alle nyhetselementer som objekter, men lagre disse slik Adresseavisen gjør i dag, med hjelp av en relasjonsdatabase. Dette vil føre til at alle nyhetselementene lagres i samme metadataformat, noe som gjør at man får et system som er enkelt for journalistene å bruke og man støtter gjenfinningen av nyhetsobjekter på en god måte. Et annet alternativ er å utvikle et system som baserer seg på NewsML eller deler av det. Hvis man velger å bruke en forenklet NewsML struktur kan man enkelt utvide denne ved behov. Ved innføring av NewsML ligger utfordringen ved å lage et godt brukergrensesnitt slik at journalistene greier å utnytte fordelene ved å kunne lagre flere versjoner av en artikkel med samme metadata. Men man bør uansett vente og se hva NewsML versjon 2.0 har å bidra med, da denne er ventet ferdig.

Referanser

[1] Oppedal, A. I., **"Alt er metadata". Bruk av metadata i et integrert Brukersystem**, IDI, NTNU, 2000

[2] IPTC, **IPTC Web: Homepage**, http://www.iptc.org. (Sist aksessert:2006)

[3] IPTC, **Press release**, http://www.iptc.org/newsmlprel.htm. (Sist aksessert:20.11.2000)

[4] IPTC, **Who's using NewsML**, http://www.newsml.org/pages/whouse_main.php. (Sist aksessert:2006)

[5] Net-federation, http://www.net-federation.de/newsml/dev_basics.html. (Sist aksessert:20.11.2000)

[6] IPTC, **NewsML Version 1.2 Functional Specification,**, http://www.newsml.org/IPTC/NewsML/1.2/specification/NewsML_1.2-spec-functionalspec_8.html. (Sist aksessert:12.11.2003)

[7] Spek and Spijkervet, **Knowledge Management: Dealing Intelligently with Knowledge**, Knowledge Management Network (Kenniscentrum CIBIT), 1997

[8] Tennant, R., **Metasearching; the problem, promise, principles, possibilities & perils**, http://www.infopeople.org/training/webcasts/02-08-05/metasearching_slides.pdf. (Sist aksessert:23.08.2005)

[9] Marchionini, G., **Information Seeking in Electronic Environments**

[10] Chowdhury, G. G., **Introduction to moderen information retrieval(second edition)**, Facet Publishing, 2004

[11] Storleer, R., **Søkeprosedyre:presisjon og søkestrategi.** 1998

[12] Sonnenreich and Macinta, **Web developer.com guide to search engines**, (Sist

[13] Baeza-Yates, R. and Ribeiro-Neto, B., **Modern information retrieval**, ACM Press, 1999

[14] Huseby, O., **Informasjonssøking**, http://www.hib.no/biblioteket/Informasjonssoking.asp. (Sist aksessert:20.08.2004)

[15] Wikipedia, http://en.wikipedia.org/wiki/Metadata. (Sist

[16] Gilliand-Swetland, **Defining Metadata**, in *Introduction to Metadata: Pathways to Digital Information*, Getty Information Institute, Los Angeles, 1998

[17] "Aalberg, Hegna", **Arkitektur for digitale bibliotek**, 2000

[18] M.Ingeberg, Laget etter opplyninger fra Adresseavisen

[19] FotoWare, **FotoWare - The complete soluution for asset management and digital imaging**, www.fotoware.com. (Sist aksessert:2005)

[20] Adresseavisen, Utskrift fra Adresseavisens database

[21] Sesam, **Sesam**, http://sesam.no/search/?q=&page=/pages/9/index. (Sist aksessert:2006)

[22] NTB, **Velkommen til NTB** 2006

[23] Falck, M., NTB Korrenspondanse ang. NTB sitt utvekslingssystem

[24] Scanpix, **Scanpix Norge**, http://www.scanpix.no/. (Sist aksessert:2006)

[25] Reuters, **Reuters The World's Leading Provider of Financial Information and News**, http://about.reuters.com/newsml/. (Sist aksessert:2006)

[26] IPTC, **IPTC - NewsCodes - List**

[27] IDEAlliance, **PRISM Specification,**
[28] IPTC, **IMM**
[29] IPTC and NAA, **Digital Newsphoto Parameter Record Guideline 1**, http://www.iptc.org/download/download.php?fn=dnprv4.zip. (Sist aksessert:20.05.2001)
[30] IDEAlliance, **Ice Specification** 2005
[31] IPTC, **The IPTC Recommended message format**, http://www.iptc.org/std/IPTC7901/1.0/specification/7901V5.pdf. (Sist aksessert:2006)
[32] IPTC, **NITF: A Solution for Sharing News**, http://nitf.org/. (Sist aksessert:2006)
[33] IPTC, **Tutorial**, http://www.nitf.org/tutorial.php. (Sist aksessert:2006)
[34] XMLNews.org, **XMLNews Technical Overview**, http://www.xmlnews.org/docs/tech-overview.html. (Sist aksessert:2006)
[35] XmlNews.org, **XMLNews Specifications**, http://www.xmlnews.org/XMLNews/. (Sist aksessert:2004)
[36] Meur, l., **NewsML for dummies**, http://xml.coverpages.org/NewsMLForDummies.pdf. (Sist aksessert:01.06.2006)
[37] Rabin, J., **NewsML Function**, http://www.iptc.org/xn-3.htm. (Sist
[38] IPTC, **IPTC Web / NewsCodes**, http://www.iptc.org/NewsCodes/nc_ts-table01.php. (Sist aksessert:02.06.2006)
[39] IPTC, **What is NewsML**, http://www.iptc.org/newsml.htm. (Sist aksessert:20.11.2000)
[40] IIM,
[41] Center, B., **RSS 2.0 Specification**, http://blogs.law.harvard.edu/tech/rss (Sist
[42] Sun, **Java technology**, http://java.sun.com (Sist
[43] W3c, **Extensible Markup Language (XML) 1.0 (Second Edition)**, http://www.w3.org/TR/REC-xml (Sist
[44] W3c, **The OAI Executive:The Extensible Stylesheet Language Family (XSL)**, http://www.w3.org/Style/XSL/ (Sist
[45] eXist, **Open Source native XML database**, http://exist.sourceforge.net/ (Sist
[46] W3c, **W3C XML Query (XQuery)** http://www.w3.org/XML/Query/ (Sist
[47] Apache, **The Apache Derby Project**, http://db.apache.org/derby/ (Sist
[48] Wong, C., **Overview of DB2's XML Capabilities: An introduction to SQL/XML functions in DB2 UDB and the DB2 XML Extender**, http://www-128.ibm.com/developerworks/db2/library/techarticle/dm-0311wong/. (Sist aksessert:2006)