

# **NRs radiodatabase**

**En oppsummering av delprosjekt Radiostrømmer i ChannelS 2003**

Forfattere  
Knut Holmqvist  
Eirik Maus

20/8-2004

**Tittel/Title:** NRs radiodatabase

**Dato/Date:** 20-8

**År/Year:** 2004

**Report nr:** 10-04

**ISBN:** 82-539-0512-2

**Forfatter/Author:** Knut Holmqvist og Eirik Maus

**Sammendrag/Abstract:**

Rapporten beskriver arbeidet med streaming av radiokanalene Radio Nova, RadioRakel og Tellus radio. Strømmene lagres, og det er utarbeidet et sett med rutiner og programmer for håndtering av meta-data.

**Emneord/Keywords:** Metadata, radio, database

**Tilgjengelighet/Availability:** Public

**Prosjektnr./Project no.:** 11010

**Satsningsfelt/Research field:** Multimedia, multikanal

**Antall sider/No. of pages:** 17

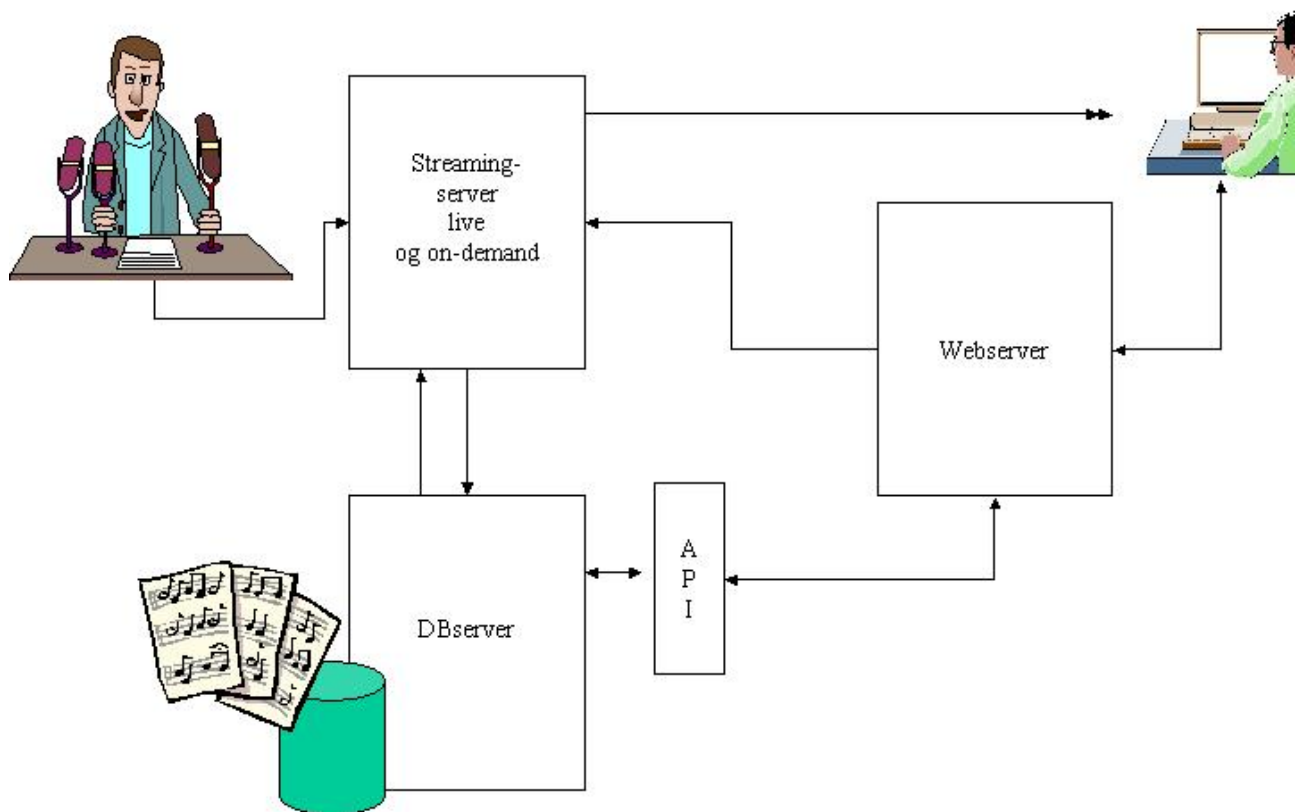
# 1 Tidligere arbeid

De siste årene har Norsk Regnesentral streamet radiofrekvensen 99,3 i Oslo. Frekvensen deles av flere radiokanaler, blant dem Radio Nova (studentradio), radiOrakel (kvinneradio) og Tellus Radio (innvandreradio).

I forbindelse med dette har NR utviklet rutiner, database og serverteknologi til bruk ved streaming av lyd, radio over Internett.

Streamingen foregår ved at et radioapparat tar ned radiosignalet fra frekvensen og sender det inn til en PC. PCen enkoder strømmen til MP3 i fire ulike kvaliteter. Deretter streames signalene. Programvaren som benyttes er IceCast på en Linux server.

Arbeidsflyten er beskrevet i Figur 1



**Figur 1**

Radiosendingen sendes live ut på Internett. Lyttere åpner lydstrømmen i en MP3-spiller, enten direkte eller via en lenke på nærradioens websider.

Sendingene blir også lagret til en databaseserver på NR. Det er laget verktøy (se Figur 2) der nærradioen kan definere lenker til tidligere sendte programmer, f.eks. "Dagens sending på tamil" eller "Bokkafeen med Dag Solstad". Lenken lages ved at man velger program fra lokalmenyen til venstre, skriver inn en tittel og velger dato og tidspunkt for start og stopp. Det genereres en automatisk URL som kan legges inn i systemet. Lenken kan klippes og limes inn i en webside.

Ved å klikke på lenkene kan radiolytteren høre tidligere sendte programmer, eller innslag i disse.

### Programsegmenter i...

Programmer
Velg

Programmer

- Bakrommet
- Barikaderock
- Bingo Beats
- Bionic Women
- Blekk
- Bokcafeen
- Bra Trommis
- DJ Experimenta
- DJ Snau
- Dhanak
- Dirtybags
- Ecko
- Elektronische

### Beskriv nytt programsegment

Program:

Tittel:

Dato:  (År-Måned-Dag)

Reell start:    (Time Minutt Sekund)

Slutt:    (Time Minutt Sekund)

URL:

FM 99.3 live ([ADSL](#), [ISDN](#), [Modem](#))

Figur 2

Det er også laget et verktøy der sluttbrukere kan gå inn og høre på et program, men kun om de kjenner tidspunkt for sendingen. Bruergrensesnittet er vist i Figur 3.

The screenshot shows a Microsoft Internet Explorer browser window. The title bar reads "Radio Streams - Microsoft Internet Explorer". The address bar contains the URL "http://www.01lab.org/public/project/radio/fm993-segment-on-demand.php". The page content features a heading "FM 99,3 MHz Audio Segment On Demand". Below the heading is a form with three dropdown menus for "YY-MM-DD-hhmm" (set to "2004-07-22-0600"), "Start (s)" (set to "01:00:00"), and "Stop (s)" (set to "01:00:00"), followed by a "Listen!" button. Below the form, there is a text block: "You'll need an MP3 player capable of doing streaming audio to listen to these. We recommend these free players:" followed by links for "Windows: WinAmp", "MacOS: iTunes", and "Linux: XMPP".

Figur 3

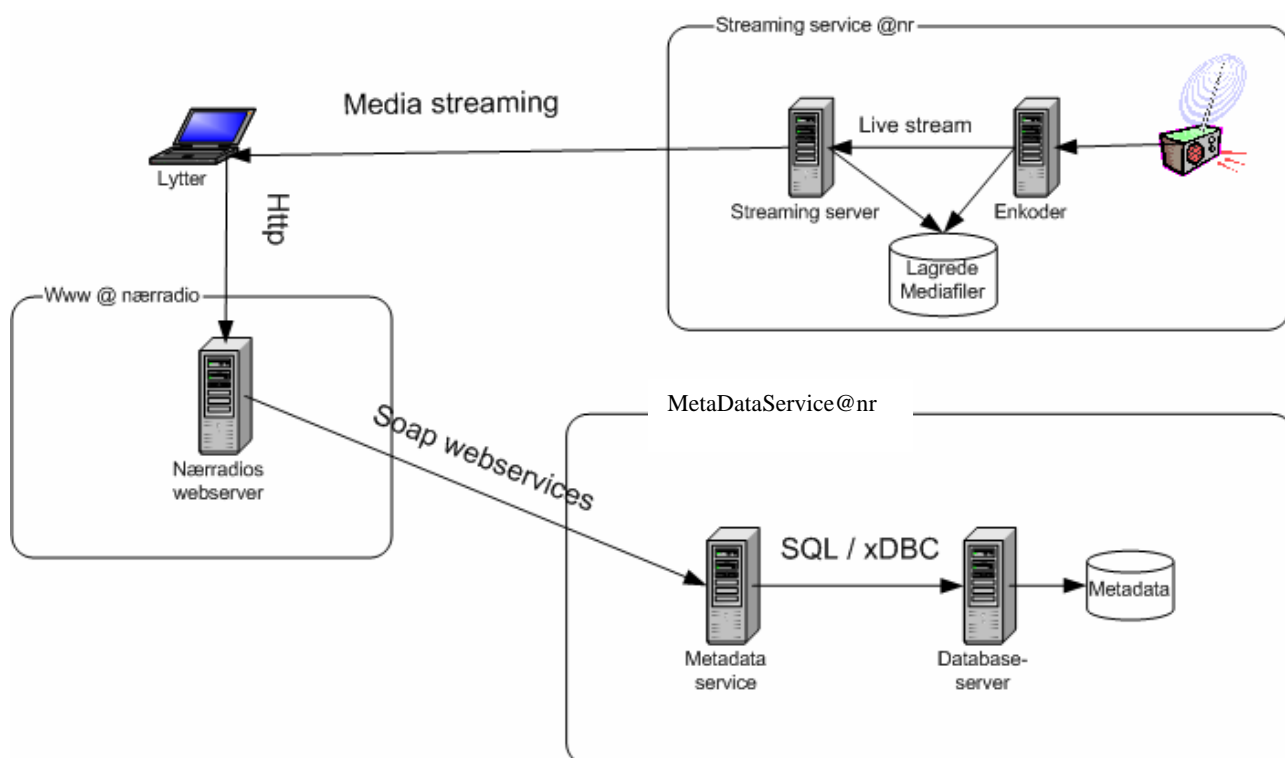
## 2 Radiodatabaseprosjektet 2003

Funksjonaliteten for sending av tidligere programmer, "on-demand", i den løsningen som er beskrevet i kapittelet "Tidligere arbeid" ble vurdert som lite tilfredsstillende i daglig produksjon. Man kan lage lenker til tidligere innslag, men det er vanskelig å sette opp muligheter for lytterne til å søke etter tidligere innslag. Den eneste av nærradioene som benyttet systemet til å lage lenker til tidligere programmer var Tellus Radio, og der var det bare enkelte av redaksjonene som benyttet det.

Det ble også vurdert å gå over til en annen teknologisk plattform. Den siste problemstillingen er oppsummert i NR-rapporten "Overview of sound-streaming".

Figur 4 viser en overordnet arkitektur for den nye løsningen for radiostrømmer. En lytter går inn på en webside og klikker på en lenke eller åpner en URL direkte i en "mp3 spiller". URLen kan være til en live sending, eller peke til en tidligere sending. Hvis URLen er til en tidligere sending, kan den finnes som en direkte lenke, e.g. "Gårdagens nyheter på urdu" som peker til en lydfil, eller via et søkegrensesnitt. Ved søk vil metadata som er lagret om sendingene være til hjelp for å få fram et passende utvalg.

Dette notatet tar i all hovedsak for seg design av funksjonalitet i boksen merket "MetaDataService@nr" i Figur 4. Brukerinteraksjon vil være gjennom websider (JSP eller tilsvarende) fra en webserver. Denne webserveren trenger ikke være plassert eller driftet av NR. I tilfellet med 99,3 har hver av radioene sin egen web-server, men alle knyttes opp mot samme streaming server og metadataserver på NR.



Figur 4

MetaDataServeren skal inngå som en del av NR/Darts arkitektur for streaming, lagring søking og gjenfinning av multimedia. Den er tilpasset for å passe inn i systemet for å streame nærradioer på nett. Metadata-databasen skal i denne omgang brukes til informasjon om radioprogram.

### 3 Metadata

Behovet for metadata har vokst med multimediale data. I tradisjonelle datasystemer er det ofte en enkel sammenheng mellom kodete data og informasjon. Eksempelvis vil tall og bokstaver ofte være lagret i ASCII format. En dekodning fra data lagret i ASCII format til skjerm eller papir og en tolking av tegnene er meget enkel, og vil ofte gi mye informasjon. Hvis det eksempelvis står Nils Ås, Gabelsgate 14, 0123 Oslo, 1600.10.12345 kan dette tolkes som navn, adresse og kontonummer til en person. Man kan også tippe at vedkommende har hatt kontoen siden "Sparebanken Oslo og Akershus" dager.

Lyd og video kodes på en mer kompleks måte. For video vil man først transformere dataene til et fargerom med en luminans- og to krominanskomponenter. Disse vil så, hver for seg, bli omformet via Fourier (DCT) eller Wavelettransformasjon. "Lydsporet" vil få en tilsvarende behandling. Koeffisientene fra transformasjonen vil bli kodet i en av flere codesystemer og det foretas kvantisering før dataene blir lagret. Et forsøk på tolking av disse dataene uten noe tilleggsinformasjon, metadata, vil være en omfattende og rimelig håpløs oppgave. Ikke umulig, men mye tyngre enn for "tradisjonelle data".

Det finnes ulike måter å definere metadata på, fra de mest formelle til de mer luftige. Vi nøyer her med å si at metadata er data som kan gi informasjon om andre data. En slik definisjon er rekursiv, dvs metadata kan ha sine egne metadata.

Metadata må betraktes på (minst) to ulike nivåer. Det første nivået er innhold eller mening, ofte kalt semantikklaget. Her defineres meningen med informasjonen. Eksempelvis definerer EBU/Dublin Core (se under):

#### **Element 5: Publisher**

*Name:* Publisher

*Definition:* An entity responsible for making the resource available

*DC Comment:* Examples of a Publisher include a person, an organization, or a service. Typically, the name of a Publisher should be used to indicate the entity.

*EBU Comment:* The publisher given here is the entity that has made the resource available in digital form.

Det andre nivået er et "teknologilag" der man bestemmer seg for hvordan man skal beskrive, og særlig utveksle, det innholdet man har blitt enig om. Teknologien som er valgt over, formatert tekst, er grei å lese for mennesker. Den tilsvarende definisjonen, skrevet som XML eller som KLV (key-length-value) vil være mindre forståelig for (vanlige) mennesker, men lettere å tolke for programmerere, noe følgende eksempel kan illustrere:

```
<xs:complexType name="Invoice">
  <xs:complexContent>
    <xs:extension base="pt:BusinessObject">
      <xs:sequence>
        <xs:element ref="jdf:JDF"/>
      </xs:sequence>
      <xs:attribute name="Currency" type="xs:NMTOKEN" use="optional"/>
      <xs:attribute name="Expires" type="xs:dateTime" use="required"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Denne oppdelingen i lag er viktig for behandling av metadata. Semantikk kan godt lagres i ulike formater, men må utveksles i den teknologi man er enige om å utveksle i. Eksempelvis vil den informasjonen som trengs for å lage en JDF fil (som benytter XML) være tilgjengelig i en kundes

MIS-løsning (administrasjonssystem), men ikke i XML format. Når en JDF fil skal sendes til en underleverandør må MIS systemet generere en XML fil som så overføres. Hos underleverandøren skal henvendelsen tolkes, enten av en selger eller av et automatisk system. I begge tilfeller skal metadataene, som sidetall, opplag, papirkvalitet osv inn i underleverandørens MIS system, der det behandles på ulike måter. Informasjonen, f.eks 32 sider, vil være den samme, men dataformatet kan variere fra et DB-sjema, til XML, til KLV og til formatert tekst, osv, alt avhengig av hvilke anvendelser man har for metadataene på ulike tidspunkt.

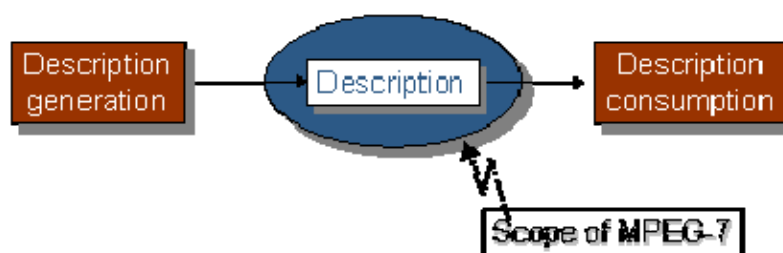
Behovet for standarder for metadata har vært åpenbar lenge, og det er utviklet standarder innen ulike områder. De senere årene har det vært gjort mye for å forene standardene innen ulike bransjer. To eksempler på dette er "Job Definition Language" JDF, innen grafisk produksjon og MXF innen videoproduksjon. JDF benytter XML skjema som teknologi mens MXF benytter KLV.

### **MPEG-7**

MPEG-7 beskriver audiovisuelle data innen en multimedial sammenheng. MPEG-7 er ikke direkte knyttet opp mot andre standarder. Man kan beskrive metadata for eksempelvis et papirbilde eller en analog video.

Følgende figurer er hentet fra "MPEG-7 overview" (ISO/IEC JTC1/SC29/WG11N5525) , og viser sider ved MPEG-7. Den første figuren skal vise at standarden omhandler hvilke semantiske elementer som inngår i standarden, mens det å generere eller å benytte disse ikke er del av standarden.

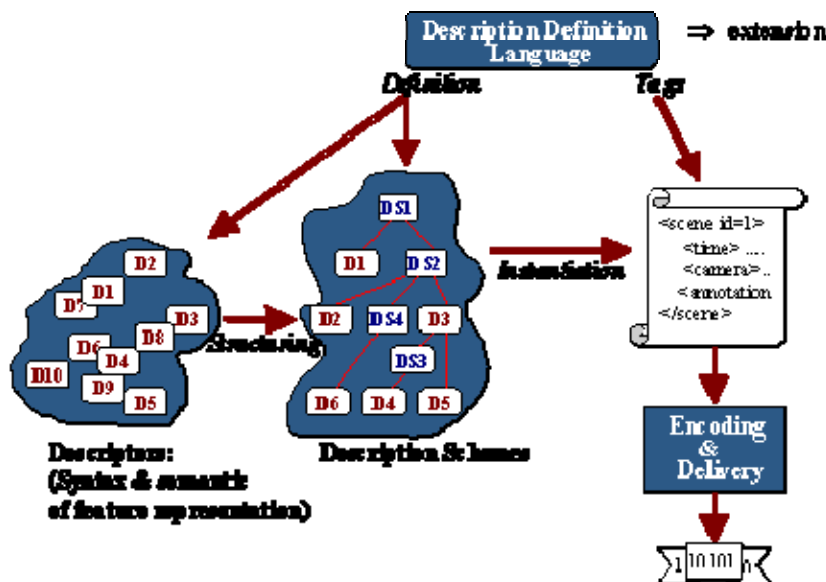
MPEG-7 benytter XML for å beskrive og utveksle metadata.



Figur 5

- Hovedelementene i XML er:  
"Description tools" som kan være enten Descriptors, som definerer både semantikk og syntaks for hvert enkelt metadataelement eller "Description Schemes" som beskriver forholdet mellom "components" som kan være Descriptors eller Descriptor Schemes.
- Et "Description Definition Language" (DDL) som beskriver syntaksen for Description tools.
- "System tools" som støtter produksjon av MPEG-7 filer.

Forholdet mellom dem er vist i Figur 6.



Figur 6

### 3.1 EBU metadatastandarder for radiosendinger

Siden vårt anliggende i denne sammenhengen er radiosendinger, har vi tatt utgangspunkt i EBU's "Core Metadata Set for Radio Archives". Siden denne er basert på en av de første metadatastandardene standardene, Dublin Core, vil vi kort omtale Dublin Core.

De 15 elementene i Dublin Core ble foreslått i 1996 av Dublin Core Metadata community, Dublin, Ohio. The Dublin Core Metadata Element Set (DCMES) kom som et resultat av behovet for bedre søkemuligheter på Internet (WWW). Til å begynne med ble det fokusert på vanlige brukere, men etter hvert har bruken blitt utvidet.

DCMES har et relativt enkelt format som kan utvides med lokale tilpasninger.

De opprinnelige taggene i DCMES var delt opp etter ulike egenskaper:

Content	Intellectual Property	Instantiation or Version
Coverage	Contributor	Date
Description	Creator	Format
Type	Publisher	Identifier
Relation	Rights	Language
Source		
Subject		
Title		

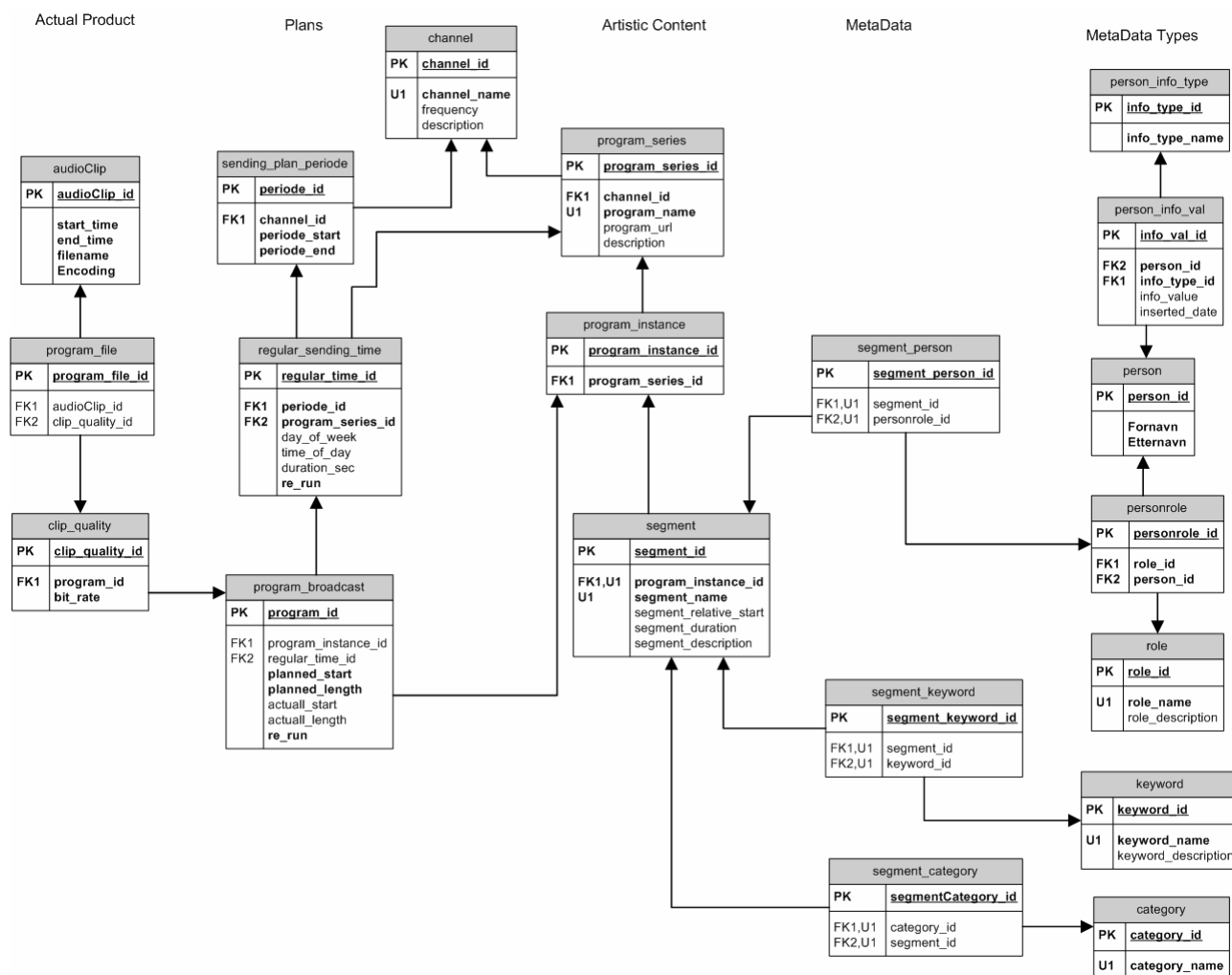
EBU baserer sin standard tidligere standard for Radio metadata på Dublin Core, men med egendefinerte tilpasninger.

### 3.2 Datamodellen

Prosjektet utviklet en datamodell som var mer tilpasset de daglige oppgavene ved en nærradio, der vi forutsatte at databasen eller systemet ville inneholde rutiner for eksport av metadata mot ulike systemer. Modellen benytter navnestruktur fra Dublin Core / EBU. Fra modellen vil man kunne generere metadata for ulike teknologier.



### 3.2.1 Modellen



Figur 7

### 3.2.2 Annen bruk av metadata-databasen

Metadata-databasen bør også være brukbar i andre prosjekter, som f.eks. ny databasestruktur i PIBB. Det viser seg at kategorisering av fil-samlinger og filer, samt indeks-punkt-registrering er nyttig i alle typer mediedatabaser. Mulig at noe av bruker- og brukergrupperfunksjonaliteten også kommer inn her, men også mulig at dette holdes utenfor.

### 3.2.3 Cueing

Vi var inne på dette med cueing (det er til og med en artikkel som ligger på under katalogen "11 - Artikler"). Cueing er jo mest aktuelt for live strømmen hvor man trenger å få signaler sammen med strømmen for å finne ut av når et innslag starter og slutter. Man kan også få en uoverensstemmelse mellom planlagt sendeskjema og cueing-signalet.

## 4 Scenarier

### 4.1.1 Metode: Viewpoints og scenarier

ISO/IEC Reference model of Open Distributed Processing definerer 5 viewpoints som til sammen skal kunne gi et utfyllende bilde av et distribuert systems funksjon og design. De fem viewpointene er:

- *Enterprise viewpoint: A viewpoint on an ODP system and its environment that focuses on the purpose, scope and policies for that system.*
- *Information viewpoint: A viewpoint on an ODP system and its environment that focuses on the semantics of information and information processing.*
- *Computational viewpoint: A viewpoint on an ODP system and its environment, which enables distribution through functional decomposition of the system into objects which interact at interfaces.*
- *Engineering viewpoint: A viewpoint on an ODP system and its environment that focuses on the mechanisms and functions required to support distributed interaction between objects in the system.*
- *Technology viewpoint: A viewpoint on an ODP system and its environment that focuses on the choice of technology in that system.*

Disse viewpoints er ment å gi en dekkende beskrivelse av systemet. I varierende grad har det blitt laget formelle beskrivelsesmetoder og språk for å uttrykke de ulike viewpointene, men det viser seg at det uansett må menneskelige vurderinger til for å sjekke om det er innkonsistenser, ufullstendighet m.v. i spesifikasjonen.

Selv om man i liten grad benytter seg av spesielle språk osv for å beskrive de ulike viewpointene, kan de være nytte til å gi en pekepinn på hvilke beskrivelser man bør ha med for å få en relativt fullstendig spesifikasjon av et system.

Engineering viewpoint er lite interessant når man bruker standardteknologi for å koble sammen systemer (RM-ODP er et ganske gammelt dokument). Det vil dermed bli integrert i de andre, først og fremst i beskrivelse av krav og funksjonalitet (enterprise) og valg av implementeringsteknologi (technology).

Under vil vi presentere en del scenarier for ulike brukergrupper.

### 4.1.2 Scenarier for radiolyttere

#### **Se ukas/dagens sendeplan**

Mål: Få fram liste over hva som skjer i dag / denne uke

- Denne ligger gjerne på webside uten forutgående interaksjon fra bruker

#### **Søk etter program på dato (eller fra dato til dato)**

Mål: Få fram liste om hva som ble sendt i ønsket tidsrom

- Samme eller annet listeforamt som "Se dagens/ukas sendeplan"

#### **Få info om programinstans (x/ fra sendeplan)**

Mål: Få informasjon om en programinstans, med tekstfelter, sendetidspunkter, personer og indekspunkter (med beskrivelse) listet opp.

1. Bruker klikker på programinstans fra sendeplan eller fra resultat av søk
2. Informasjon om programinstans vises

### **Søk etter programmer med person**

Mål: Få liste over innslag med *person*, gjerne gruppert etter programinstans”

- Søkefelt hvor man kan skrive inn navn. (og evt ønskede roller?)

### **Søk i fritekst**

Mål: Få fram liste over programinstanser/innslag annotert med fritekstord

### **Lytt på opptak, fra og med indekspunkt**

Mål: høre på opptak av lyd fra et gitt tidspunkt.

- bruker klikker på lenke som er resultat av noen av søkene over.

## **4.1.3 Scenarier for dem som driver radiostasjon**

Flere av disse scenariene kommer ikke til å bli brukt særlig ofte. ”Opprett ny programserie” skjer veldig sjelden, og kan gjøres for hånd inntil videre.

### **Opprette ny programkategori**

Mål: Få opprettet en ny programkategori for denne radiostasjonen, som kan listes i webside og brukes til å assosiere kommende programserier

### **Opprette ny programserie**

Mål: Få opprettet en ny programserie for denne radiostasjonen, som kan listes i webside og til å assosiere kommende programinstanser.

Uløst:

- Man kan samtidig taste inn antatt varighet for programserien, fast sendetid e.l., og få generert opp programinstanser for kommende uker, måneder, år, tiår, eller dette kan gjøres på et senere tidspunkt

### **Redigere sendeskjema**

Mål: Gjøre endringer på hvilke programinstanser som skal starte når

### **Kopiere forrige periodes sendeskjema**

Mål: Ha noe å starte ut fra når det gjelder å redigere et sendeskjema.

### **Generere sendeskjema**

”Sendeskjema” finnes egentlig ikke. Det er implisitt gitt av alle planlagte (opprettede) programinstanser, og genereres ut i fra de andre tabellene. (Et demoprogram som gjør dette er laget, se under).

## **4.1.4 Scenarier for dem som lager radioprogram**

### **Opprette ny programinstans**

Mål: få opprettet en ny programinstans det kan redigeres på

### **Opprette nytt innslag for programinstans**

Mål: få lagt til et innslag i programinstansen som redigeres.

### **Redigere data for programinstans**

Mål: endre data om programinstans

1. Brukeren velger programinstans fra liste
2. Får data med detaljer om instans opp i skjermbilde. Også "redigere"-knapp/link hvor bruker (innlogget!) har lov til å redigere
3. Klikker på "redigere"
4. Får opp alle data om programinstans i skjema med redigerbare felter
5. trykker Lagre for alle endringer som skal lagres
6. trykker "Ferdig" når ferdig => kommer til liste fra punkt 1.

### **Redigere data om/for/i innslag**

Uløst:

- Skal programinnslagene redigeres som del av en programinstans, eller skal de redigeres separat og være mulig å flytte til ulike programinstanser

### **Slette innslag fra programinstans**

Uløst:

- Skal innslag slettes hvis den ikke er assosiert med en programinstans?

## **4.1.5 Scenarier under og etter sending**

### **Trykk "nå starter det" på indekspunkt**

Mål: Få lagt inn eksakt startpunkt i database for indekspunkt.

Bruker: studiotekniker eller programleder

1. Når innslag starter, trykk OK på knapp ved siden av innslag i programplanen
2. Returnerer til samme side med faktisk startpunkt satt til trykk tidspunktet

## **4.2 Valg av implementeringssteknologi**

MetaDataService baserer seg på standard n-lags arkitektur for tjenerløsninger. Vi har i hovedsak to ulike lag:

- *Metadata tjener* har ansvar for å ta i mot og håndtere forespørsler fra klienter. Tjeneren har et eller flere SOAP grensesnitt. Noen av disse må muligens beskyttes med brukernavn og passord. Et spørsmål vi avgjøre er om SOAP grensesnittet skal være en RPC type (request/response) eller av dokument type (hvor man utveksler XML dokumenter). En blanding er også mulig.

Tjeneren skal også kommunisere med databasen.

Vi må også ta stiling til om vi skal benytte oss av en applikasjonstjener. Fordelen med dette er at vi kan programmere en web applikasjon som pakkes som en WAR fil. Denne kan da benyttes på alle J2EE applikasjonstjenere og servlet containere.

- Mulige containere: Tomcat, JBoss, WebLogic (J2EE løsninger).
- SOAP løsninger: GLUE eller Axis.
- Persistenslag: Castor, Hibernate eller annen JDO løsning.
- DB-caching: Proxool
- Logging: log4j

- *Database* hvor alle metadata ligger lagret. All kommunikasjon med databasen bør gå igjennom applikasjonstjeneren siden den kan cache data. Et poeng med databasen er at den bør ha gode mekanismer for søking (f.eks. fritekstsøk). Det er ikke så viktig at tabellene har ACID egenskaper siden transaksjoner ikke er veldig viktig i applikasjonen.

Database: Microsoft SQLServer.

### **4.3 Demonstrator**

For å teste ut databasemodellen valgte vi å implementere en løsning for å lagre metadata om sendingene til RadioNova. Fremgangsmåten vi brukte var å lage modellen i Visio og å generere databasen direkte fra VISIO i Microsofts SQL server (databasen). For å gjøre det lett å programmere benyttet vi Java open-source løsningen Hibernate for overgangen mellom relasjonsdatabasen og objekter i Java. Ved å bruke Hibernate blir radene i tabellene representert som vanlige Java objekter.

For å ha noe innhold i databasen lagde vi et program som tok sendeplanen til RadioNova og la denne inn i de riktige tabellene i databasen. Med innhold i tabellene genererte vi så sendeplanen.

## 5 Universitetsstudier

### 5.1 Hovedfagsoppgave for Benedikte Lund (Cand. Scient eksamen)

Metadata kan samles og legges inn i databaser på to måter, manuelt eller automatisk. Det er utfordringer forbundet med begge måter. Manuell innlegging er ofte kostbar, og det krever stor disiplin av dem som gjør det. Det skal også være streng struktur i metadataene for at to personer skal beskrive samme medieobjekt på samme måte.

Automatisk generering av metadata vil generelt være en fordel. Eksempler på dette er digitale stillbildekamera som genererer informasjon og legger dem i bildefilen (EXIF header). Dette er opplysninger som brennvidde, lukkerhastighet, avstand til fokusobjekt, dato og tidspunkt, osv. Profesjonelle videokamera gir tilsvarende opplysninger. For alle videoklipp genereres en unik nøkkel, basert på kameratype, tidspunkt osv (SMPTE UMID) samt en del tekniske metadata om ett opptak.

For informasjon om innhold av en video er det vanskeligere å generere metadata automatisk. Når det gjelder å finne semantikk ut i fra bilder kreves det mye mer forskning. For radio, og også en del videosendinger, vil imidlertid i form av tale kunne gi mye metadata. Indeksering av tekst er et område der vi har mye metoder og erfaring. Hvis man kan bruke tale-til-tekst metoder for å lage tekstdokumenter ut fra radiosendinger, kan man benytte kjente metoder for indeksering. En del tale-til-tekst programmer gir også en tidsakse i teksten relativt til talen.

Tale til tekst fortsatt en umoden teknologi. Generelt vil man ha systemer

- 1) Som har et stort ordforråd
- 2) Som man ikke trenger å lære opp
- 3) Der man kan bruke kontinuerlig tale

I de systemene som benyttes i dag kan man, med akseptabelt resultat for diktat, få "to av tre". Dvs, vil man ta tiden med å "lære opp" systemet kan man få et stort ordforråd og kontinuerlig tale. Hvis ikke så kan man få en av de to andre.

Det som anses som et uakseptabelt resultat for diktat; kan imidlertid være akseptabelt for andre formål, f.eks. indeksering. Mens man i diktat ønsker minst 95% treff, kan indeksering muligens klare seg med mindre. I en nyhetssending der "Krigen i Irak" omtales vil "Krig" og "Irak" gjentas mange ganger og det er ikke nødvendig at alle instanser oppfattes.

Benedikte Lund leverte sin hovedoppgave våren 2003 med Knut Holmqvist som faglig veileder. Oppgaven var å undersøke hvordan man kan benytte tale-til-tekst metoder for å lage tekstfiler av radiosendinger, og om disse så kunne indekseres ut i fra nøkkelord i teksten.

Hun benyttet standard tale til tekst teknologi (fra IBM) til å generere tekstfiler. Disse ble så indeksert, og databasen ble oppdatert med nøkkelord om innslagene.

Konklusjonen var at metoden fungerte tilfredsstillende. Datamaterialet var sendinger fra BBC som ble lastet ned fra Internett. Kvaliteten på dette datamaterialet var ikke veldig høy, og i en redaksjon der man har tilgang til originaltaper vil metoden trolig fungere ennå bedre.

Oppgaven kan hentes fra <http://folk.uio.no/idalu/paper.pdf>.

## 5.2 Semesteroppgave INF 102 våren 2003

Bror Gundersen, som var ansatt ved NR våren 2003, skrev sammen med andre studenter (Gundersen et al<sup>1</sup>), en semesteroppgave med utgangspunkt i prosjektet.

De definerte følgende:

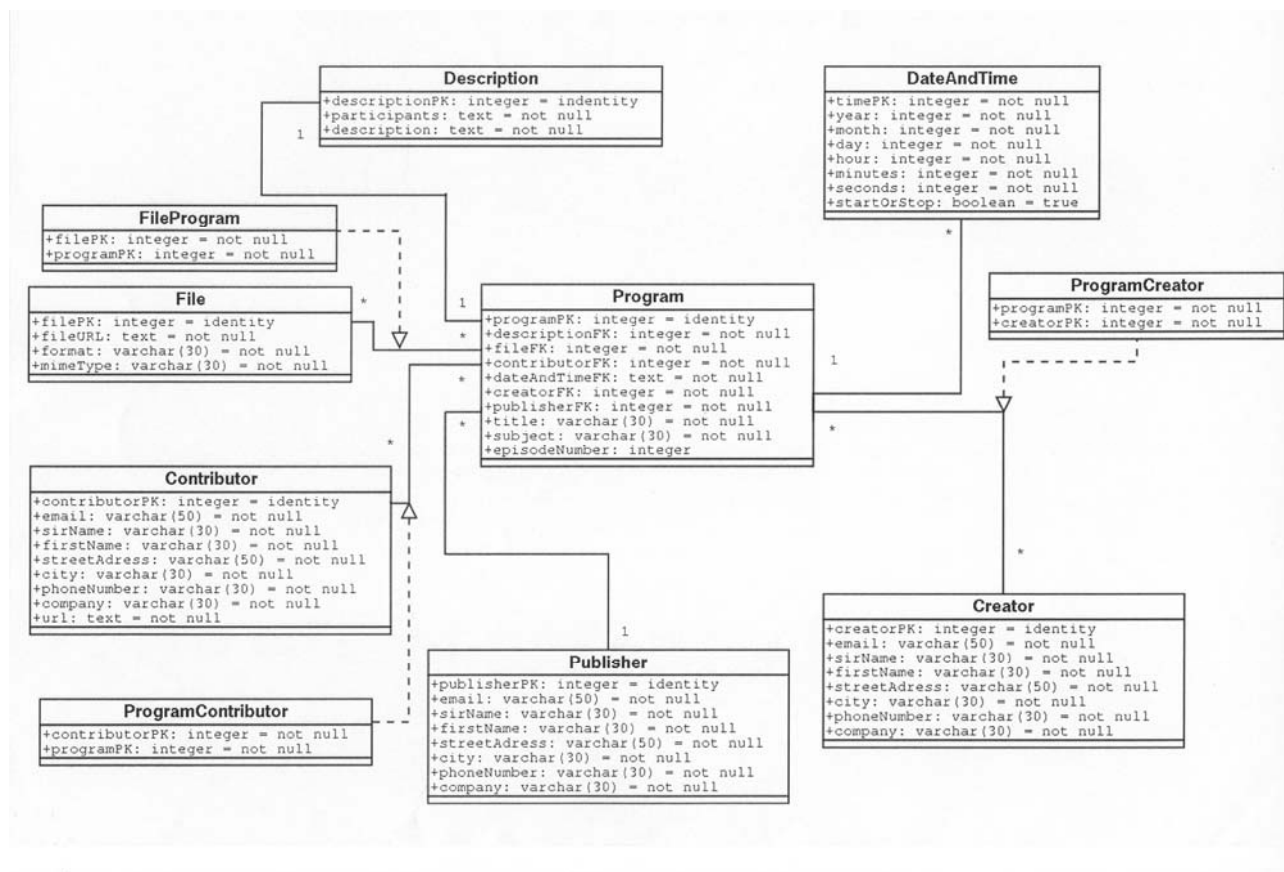
### "Purposes

Our system will:

- Have a more systematic archive, which will help the employees and listeners to find and use necessary data easier than before.
- Make it possible to search for the time of a broadcast and/or keywords related to the broadcast and then stream the appropriate sound file.
- Make it possible to index and annotate broadcasts.
- Make it easier to reuse previously made material.
- Pose the opportunity for the listeners to have "Play-on-Demand" so they can select what to play and when. There will be no time limit for streaming a program.

Our goal is that the radio station will get more fame, have a growth in number of listeners. Hopefully that will lead to wellbeing and possible growth of income to the radio station."

Datamodellen de kom fram til var som følger:



Figur 8

Denne datamodellen tar utgangspunkt i et "Program", og lenker utvalgte metadataelementer fra Dublin Core / EBU opp mot programmet.

<sup>1</sup>Åshild Eldholm, Bror Gundersen, Elizabeth Kerr, Rustam Mehmandarov og Bhavna Pathania, prosjektoppgave INF 102 våren 2002

## 6 Prosjektdeltakere

Følgende personer har deltatt i prosjektet:

- Benedikte Lund (Hovedfagsstudent)
- Bror Gundersen (Student og ansatt)
- Eirik Maus
- Jørn Inge Vestgården
- Knut Holmqvist (Prosjektleder)
- Per Thomas Jahr
- Svetlana Boudko