Forord

Våren 2006 ble det gitt en diplomoppgave ved institutt for telekommunikasjon, Norges tekniske-naturvitenskapelige universitet. Oppgaven er i samarbeid med Adactus AS med daglig leder Peder Drege som veileder og professor Andrew Perkis som faglærer.

Diplomoppgaven gikk ut på at konseptet filmweb.no skulle videreutvikles til å bli en 3G-mobiltjeneste ved hjelp av MPEG-21. Diplomoppgaven er en avsluttende oppgave ved Master i elektronikk.

Jeg ønsker å rette en stor takk til Adactus, og da spesielt til Peder Drege og Thomas Skjølberg for god veiledning. I tillegg vil jeg takke Andrew Perkis for mye motivasjon og god støtte.

12.06.2006

Trude Vareide

Sammendrag

Denne diplomoppgaven beskriver videreutviklingen av konseptet filmweb.no til å bli en 3G-tjeneste basert på ISO-/IEC-standarden MPEG-21. Filmweb.no er en internettjeneste som informerer om kinotilbudet og gjør det enklere å kjøpe kinobilletter. Diplomoppgaven har resultert i et menysystem 3G-Filmweb vist i GUI.

Ut fra dette menysystemet kan det sies at MPEG-21 er et rammeverk som kan gi innholdsleverandører en strukturert og kontrollert måte for multimedieleveranse. Del 2 av MPEG-21, Digital Item Declaration, har blitt benyttet til å implementere multimedieressurser og tilhørende metadata, mens del 10, Digital Item Processing, har blitt benyttet til prosesseringen. Sammen utgjør de et Digital Item som har blitt testet i et Javamiljø.

Programkoden har blitt skrevet i XML, ECMAScript og Java. Digital Item er skrevet i XML siden dette er formateringsspråket til MPEG-21. Prosesseringsverktøyene fra Digital Item Processing er implementert i XML ved hjelp av ECMAScript. Java har blitt benyttet til kjøremiljø.

Innhold

Figurer

Forkortelser

MPEG	Moving Pictures Experts Group
DI	Digital Item
DID	Digital Item Declaration
DIDL	Digital Item Declaration Language
DIA	Digital Item Adaption
DIAC	Digital Item Adaption Configuration
QoS	Quality of Service
DIP	Digital Item Processing
DIM	Digital Item Method
DOM	Document Object Model
DIML	Digital Item Method Language
DIBO	Digital Item Base Operation
DIXO	Digital Item eXtension Operation
XDI	Context Digital Item
CDI	Content Digital Item
W3C	World Wide Web Consortium
XML	Extensible Markup Language
HTML	Hypertext Markup Language
SGML	Standard Generalized Markup Language
3GPP	3rd Generation Partnership Project
NMT	Nordisk Mobil Telefon
GSM	Global System for Mobile Communications
UMTS	Universal Mobile Telecommunications System

Kapittel 1

Innledning

Manuell mobiltelefonitjeneste ble introdusert i Norge i 1966 – en tjeneste som ble forløperen til det automatiske NMT-systemet som kom i 1981. Den digitale arvtakeren, GSM, ble åpnet i 1993. Siden den gang har antall mobiltelefoner eksplodert. 3.generasjons mobilnett, UMTS, ble klart for kommersielt bruk i 2004. UMTS-nettet har høyere overføringshastighet av data enn GSM-nettet. [19],[1]

Tjenester som er spesielle for UMTS-nettet kalles 3G-tjenester og kan for eksempel være videosamtaler eller mobil-TV. Stadig raskere dataoverføring på internett fører til samme forventninger til mobiltjenester.

Dagens mobiltjenester som viser film på mobil benytter forskjellige leverandører til sine løsninger. Eksempler er Apple, Envivio, Harris, Real, Snell og Wilcox, Vidiator og Nextreaming. Disse er de mest aktuelle leverandører av videokodings- og streamingløsninger. Kontrollert overføring og god presentasjon er viktig for mobilapplikasjoner. Mange mobilapplikasjoner fungerer dårlig og gir misførnøyde brukere av tjenesten. Flere har opplevd at bestilte mobiltjenester ikke kommer frem eller de er så dårlig tilpasset mobilskjermen at det er nesten umulig å se innholdet.

Filmweb er en internettjeneste som blant annet benyttes til å markedsføre kinotilbudet. Internettjenesten inneholder informasjon om kinofilm, for eksempel filmomtale, skuespillerinformasjon og filmtrailere. I en spørreundersøkelse gjort for studentene på NTNU ble resultatet at i en tilsvarende tjeneste på mobiltelefon er filmtrailer det man ønsker minst. Det ble da tatt utgangspunkt i websiden for Filmweb og hvilke tjenester man ut fra denne websiden man kunne tenkt seg å ha tilgjengelig på mobilen. Tjenestene som det ble spurt studentene om var kinoprogram, filmomtale, topp10-liste for kino, billettbestilling og filmtrailer. Ut fra disse tjenestene kom filmtrailer dårligst ut. Når man da spurte studentene om grunnen til dette var at de var

skeptiske til kvaliteten man vil få på en film på mobil. Resultatene for spørreundersøkelsen er vedlagt i vedlegg C.1, mens skjemaet for undersøkelsen er vedlagt i vedlegg E.1.

3G-Filmweb er en 3G-tjeneste og har store muligheter for å kunne presentere multimediainnhold. Tjenesten i diplomoppgaven er begrenset til å kunne vise bilde, tekst og film. Oppgaven forutsetter at mobilbrukerens posisjon er kjent og dette legges til grunn for hvilken kinotilknytning som presenteres. Tjenesten vil være en god markedsføring for kinotilbudet da den gir en god brukeropplevelse.

3G-tjenesten benytter UMA som overføringskonsept, mens selve overføringen er basert på MPEG-21. XML, JavaScript og Java ble benyttet som programmeringsspråk og resultatet ble presentert i GUI. Tjenesten er begrenset til å kunne vises i en GUI da dagens mobiltelefoner ikke støtter ECMAScriptmaskinen som benyttes i oppgaven.

UMA er et konsept hvor et hvert innhold skal være tilgjengelig for brukeren uavhengig av termianl og nettverkstilknytning. Hensynet til Quality Of Service i UMA-konseptet er beskrevet i artiklene *UMA enabled environment for mobile media using MPEG-21 client and server technology*, [2], *A QoS-driven UMA viewer in MPEG-21 framework*, [3] , og *On some QoS Aspects of User Profile in Unniversal Multimedia Access*, [27].

MPEG-21 er overordnet MPEG sitt rammeverk for hvordan en skal overføre eller konsumere multimediaressurser. [12] *IEEE Transaction on multimedia, vol 7, No. 3, June 2005* har en introduksjon til MPEG-21 og beskriver de enkelte delene av MPEG-21, [5],[11],[29],[4],[9]. Artikkelen *MPEG-21 enabled client for rich mobile content delivery*, [24], beskriver hvordan man benytter MPEG-21 til å levere multimediainnhold til mobiltelefoner. Standarden av MPEG-21 er utformet av ISO/IEC JTC1/SC29/WG11. [14],[13],[15]

Formateringsspråket XML er MPEG-21 sitt språk. Bøkene *Dynamiske websider*, [6], *Inside XML*, [10], *Java og XML*, [17] og *XML and Java, Developing Web Applications*, [22] var viktig litteratur for å kunne sette seg inn i XML. XMLSpy ble benyttet som XML-editor. XMLSpy følger W3C sin XML-spesifikasjon og ved å implementere i XMLSpy og bruke validatorene der, ble det sikret at XML-koden var i henhold til MPEG-21 sine skjema og W3C sine regler for XML-bruk.

Programmeringsspråket Java ble benyttet til å sette hele systemet sammen. Her ble editoren Eclipse benyttet. Bøkene *Programmering i Java*, [23], *Java, Servlet Programming*, [18], og *Developing JAVA Software*, [26], var nyttige i dette arbeidet.

Tjenesten skulle presenteres i et implementert GUI for PC for å gi en illustrasjon på hvordan den ville sett ut i en mobiltelefon.

Kapittel 2

Teori

2.1 UMA

FIGUR 2.1: Universal Multimedia Access

UMA står for Universal Multimedia Access og er et konsept som beskriver på et overordnet nivå hvordan mulitmedia skal tilpasses. [2],[3],[27] UMA skaffer brukeren en best mulig presentasjon under forskjellige terminal- og nettverksforhold. Målet til UMA er å identifisere eksisterende og nye verktøy som er nødvendig for å kunne tilpasse en multimediaressurs til hvilken som helst multimediakapabel terminal. Når verktøyene er definert og tatt i bruk, kan man i prinsippet "Create once, publish anywhere" en optimal tjenestekvalitet i hvert tilfelle. Orginalinnholdet med høyest mulig kvalitet lagres på serveren og leveres til alle typer terminaler og nettverk gjennom forskjellige overføringsverktøy, nettverksverktøy, kodekverktøy og konfigureringsverktøy. Mange klienter krever imidlertid at den samme informasjonen kan leveres og mottas på ulike måter, i forskjellige filformater og oppløsningsgrader.

2.2 XML - eXtensible Markup Language

XML er et språk definert av World Wide Web Consortium, W3C, og er stammen som setter standarden for Web. [6],[10],[17],[22] XML er et formateringsspråk og et språk for å definere nye formateringsspråk. I XML kan man først definere hvilke strukturer man ønsker å benytte, dernest kan man benytte strukturene. For eksempel kan man lage egne elementer ved å designe et skreddersydd formateringsspråk for eget bruk. På denne måten overgår XML andre formateringsspråk som Hypertext Markup Language, HTML, hvor alle HTML-elementene er definert på forhånd. Det kan derfor sies at XML er et meta-formateringsspråk i og med at man kan lage sitt eget formateringsspråk. Formateringsspråk handler om å beskrive formen av dokumentet, det vil si måten innholdet av dokumentet skal tolkes på.

Både XML og HTML er basert på Standard Generalized Markup Language, SGML. Som navnet antyder er SGML et veldig generelt formateringsspråk, med enorme kapasiteter. På grunn av det store mangfoldet hos SGML er det et veldig vanskelig språk å lære seg. XML er derfor egentlig en "enklere-å-bruke"-del av SGML, mens HTML er en applikasjon av SGML.

For å være lovlige XML-dokumenter må dokumentene oppfylle en del kriterier. Dersom kriteriene er oppfylt sies de å være Well-Formed Documents. Dokumentene skal i tillegg ha en definisjon av hvilke strukturer som skal brukes, og hva de kan inneholde. En slik definisjon kalles en DTD – Document Type Definition. En DTD forteller kun om hvilke elementer dokumentet kan bestå av og den innbyrdes sammenhengen mellom dem.

En alternativ måte å beskrive datastrukturer i XML på er skjema. Skjema er til forskjell fra DTD-er skrevet i XML-kode. Det finnes to ledende skjema, W3C XML Schema og Microsoft XML Schema. I denne diplomoppgaven benyttes det førstnevnte. Skjema er mer effektivt og presist enn DTD. I forhold til DTD har man i XML-skjema muligheter til for eksempel å spesifisere syntaksen, spesifisere den egentlige datatypen av hvert element og kreve unike attributt- eller elementverdier.

2.2.1 Parser

En av de viktigste delene til en XML-applikasjon er XML parser. Den håndterer oppgaven med å gjøre om rådokumentet til noe som gir mening. I tillegg forsikrer den at dokumentet er lovlig. Hvis en DTD eller skjema er referert kan den forsikre at dokumentet er kontrollert i henhold til DTD eller skjema, det vil si at XML-dokument er validert. Resultatet fra et parset XML-dokument er typisk at datastrukturen kan manipuleres og håndteres av andre XML-verktøy eller Java APIer.

Parseren går gjennom dokumentet på en strukturert måte og lar User kontrollere hvilke elementer det skal fokuseres på. XML-elementene kan både ha en attribut som gir viktig informasjon og innhold som gir viktig beskrivende informasjon. Følgende eksempel kan beskrive dette:

<did:Statement mimeType=”text/plain”>Kinoprogram i dag</did:Statement>

Her vil mimeType = ”text/plain” være attributten og fortelle at dette er en tekst, mens ”Kinoprogram i dag” er innholdet til tagen.

I denne oppgaven benyttes som sagt XML-skjema. For at parseren skal benytte ønsket XML-skjema må selve XML-dokumentet være et XML-skjema instanse dokument (xsi). xsi-dokumentet må vite hvor det kan finne skjemafilen, det vil si .xsd, og derfor må følgende namespace inkluderes:

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="urn:mpeg:mpeg21:2002:02-DIDL-NS C:\XML\MPEG-21_skjema\didl.xsd"
```

FIGUR 2.2: Figuren viser at XML-dokumentet inneholder XML-skjema instansedokument (xsi) og skjemalokalisering.

DOM - Document Object Model

Document Object Model har sin bakgrunn fra World Wide Web Consortium, W3C. [16],[7] DOM er en standard for prosessering av XML-dokumenter. En DOM-parser bygger en trestruktur som inneholder hele XML-dokumentet i maskinens minne, i form av et Document-objekt. Etter at dokumentet er parset og en har bygget DOM-strukturen får en tilgang til dokumentet via en peker til DOM-strukturen. En kan manipulere XML-dokumentet ved å bevege seg i DOM-strukturen. Et alternativ til DOM-parser er en SAX-parser. SAX-parsere er raske og egner seg for store dokumenter som ikke kan lagres i minnet på maskinen. Siden XML-dokumentet er lite og DOM-parseren er lettere å bruke og har innebygget funksjonalitet for å bevege seg i XML-dokumentet, ble det valgt å benytte en DOM-parser. Javametoden for DOM-parser er vist i figur 2.3.

XPath

XPath står for XML Path Language og er et språk som er utviklet for å hente ut informasjon som er lagret i XML-dokumenter. [28] XPath henter ut informasjonen ved å spesifisere hvor informasjonen er lagret ved å gi stien til informasjonen. Eksempel på XPath er vist i figur 2.4.

```
public static Document load(String string) throws Exception{
    DocumentBuilderFactoryImpl factory = new DocumentBuilderFactoryImpl();
    factory.setIgnoringElementContentWhitespace(true);
    factory.setNamespaceAware(true);
    DOMParser parser = new DOMParser();

    FileInputStream filein = new FileInputStream(string);
    parser.parse(new InputSource(filein));
    return parser.getDocument();
}
```

FIGUR 2.3: Metode for DOM-parser hentet fra Javakode vedlagt i vedlegg E.1.

```
try {
    NodeList list = Parsing.doXPath(didDocument, "//dip:MethodInfo");

    System.out.println("Fant " + list.getLength() + " DIP-metoder");

    for(int i = 0; i < list.getLength(); i++) {
        Element element = (Element)list.item(i);

        componentList.addElement( element.getParentNode().getParentNode().getParentNode());

    }

} catch (XPathExpressionException e) {
    // TODO Auto-generated catch block
    e.printStackTrace();
}
```

FIGUR 2.4: Eksempel på XPath hentet fra Digital Item.

2.3 ECMAScript

13.juni 2002 foreslo en gruppe med bedrifter ledet av BEA Systems et sett med utvidelser til programmeringsspråk som gir grunnleggende XML-støtte til ECMAScript (ECMA-262). [8] Programmeringsspråkets utvidelser var laget for å frembringe en enkel og velkjent programmeringsmodell for XML. Programmeringsmodellen skulle gjøre lærekurven for XML mindre bratt ved å utnytte eksisterende dyktighet og kunnskap i et av de største utviklermiljøene i verden. Fordelen med denne programmeringsmodellen for XML er redusert kodekompleksitet og tettere forandring. Det benytter velkjente konsepter, metoder og syntaks for manipulering av XMLdata, noe som betyr at programvareutviklerne kan begynne å opprette, navigere og manipulere XML uten særlig behov for ekstra kunnskap.

Gruppen ECMAScript (Ecma TC39-TG1) godtok enstemmig forslaget og etablerte en delgruppe for å standardisere syntaksen og semantikken til et generelt forslag med en tverrgående plattform. De skulle være leverandøren av et nøytralt sett med utvidelser av programmeringsspråk kalt ECMAScript for XML (E4X). Utviklingen av denne standarden startet 8.august 2002. Denne standarden ble utviklet som en utvidelse til ECMAScript, utgave 3, men kan også brukes til andre versjoner av ECMAScript.

Denne standarden legger til naturlige XML datatyper til ECMAScript-språket, utvider semantikk til kjente ECMAScript-operatorer for manipulering av XML data og legger til små sett med nye operatorer for vanlige XML-operasjoner, slik som søking og filtrering. Den legger også til systemstøtte for litterale XML, nampespace, kvalifiserte navn og andre mekanismer for å forenkle utførelsen i XML.

2.4 Java

Programmeringsspråket Java med tilhørende teknologi ble offisielt annonsert 23.mai 1995 av forskningssjefen i Sun Microsystems, John Gage, og Netscape-grunderen Marc Andreessen. [23], [18], [26] Språket ble til etter et utviklingsprosjekt hos Sun som het Green.

Java er et objektorientert programmeringsspråk. Det er laget etter mønster av C++, men er forsøkt gjort enklere å bruke og lære ved å kutte ut det mest avanserte. Det som skiller Java fra C++ er mulighetene man har til å lage applets.

En applet er et lite program som kjøres ved hjelp av en Web-browser, som for eksempel Explorer. En applet kan være en animasjon eller den kan for eksempel spille av lydfiler. I motsetning til animasjon har man også mulighet til interaktivitet.

Alt er klasser i Java, og det finnes utallige forhåndsdefinerte metoder som kan brukes i appletene, enten direkte eller man kan lage sin egen versjon. Disse metodene finnes i Java API. API er et standard programbibliotek som følger med Java-verktøyene. Det defineres som en spesifikasjon som beskriver hvordan en programmerer skal skrive en klient som aksesserer oppførselen og tilstanden til klasser og objekter.

Javaprogrammer kan deles i tre typer: Selvstendige Java-programmer, eller applikasjoner, Java-servleter og Java-appleter.

- Applet - Et Java-program som utføres inne i en nettleser, idet det er en del av et web-dokument.
- Applikasjon - Et frittstående program, som er ment å kjøre på egen hånd uavhengig av for eksempel nettleser.
- Servleter - Javaprogram som håndterer tjenester i en webtjener, slik som å slå opp en adresse i en database.

2.5 MPEG-21

MPEG står for Moving Pictures Experts Group og er et internasjonalt standardorgan. [12] MPEG lager sett av standarder for audio- og videokompresjon.

Det finnes mange elementer for å bygge en infrastruktur for levering og forbruk av multimedia innhold, men det finnes ingen overordnede avspeilninger som beskriver hvordan disse elementene, enten tilstedeværende eller under utvikling, er relatert til hverandre. Hovedoppgaven for ISO/IEC 21000, altså MPEG-21, er å beskrive hvordan disse varierende elementene passer sammen. Hvor det ikke finnes tilsvarende, vil MPEG-21 anbefale hvilke nye standarder som kreves. ISO/IEC JTC 1/SC 29/WG, MPEG, vil så utvikle nye standarder som en tilpasning, mens andre relevante standarder kan være utviklet av andre. Disse spesifikasjonene integreres i rammeverket for multimedia i et samarbeid mellom MPEG og andre aktører.

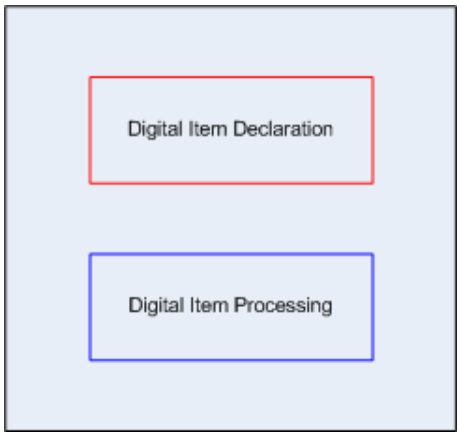
Resultatet er et åpent rammeverk for multimedialevering og -forbruk, med både innholdsleverandører og innholdsforbrukere som fokus. Dette åpne rammeverket skaffer innholdsleverandører og tjenesteforsørgere lik mulighet i et åpent MPEG-21-marked. Dette vil også gi den fordelen for innholdsforbrukeren at det gir dem tilgang til et stort variert innhold på en interoperabel måte.

Visjonen for MPEG-21 er å definere et multimedia rammeverk for å muliggjøre et åpent og forstørret bruk av multimediaressurser på tvers av et stort område av nettverk og enheter brukt av forskjellige aktører. Det er overordnet MPEG sitt forsøk på modellere hvordan en skal overføre eller konsumere multimediaressurser. Et sett av verktøy defineres for å kontrollere multimedialeveranse og disse verktøyene standardiseres av MPEG. Verktøyene i MPEG-21 er deklarert i XML-skjema. Mer om XML-skjema er beskrevet i avsnitt 2.2.

Atomet til MPEG-21 er Digital Item, det er den minste enheten som kan overføres mellom de ulike brukerne. Alt som blir utvekslet mellom to brukere er pakket inn i en XML-pakke, som har sitt eget språk Digital Item Declaration Language og utgjør et Digital Item. Fordelen med dette er det kjente uttrykket ”Instead of many files -> ONE FILE”. Digital Item er et strukturert, digitalt objekt med standard representasjon, identifisering og metadata.

En User i MPEG-21 er uansett eksistens med multimedia rammeverk og inkluderer slik alle delene i verdikjeden som for eksempel skaper, egentlig innehaver, distribuerer, og forbruker av Digital Item. User inkluderer for eksempel individualister, forbrukere, samfunn, organisasjoner, selskaper, forbund og regjering. Figuren under viser Digital Item for XML-dokument vedlagt i vedlegg E.1. Denne Digital Item inneholder kun Digital Item Dec-

laration og Digital Item Procesing, men den kan også inneholde flere deler av rammeverket som for eksempel Digital Item Identification og Digital Item Adaptation. Digital Item Declaration, Digital Item Adaptation og Digital Item Processing vil bli sudert nærmere i oppgaven.



FIGUR 2.5: Modell for Digital Item gjeldende for denne diplomoppgaven.

2.5.1 Del 2: Digital Item Declaration

Del 2 av MPEG-21 definerer strukturen og sammensetningen av et Digital Item. [14],[11] Et Digital Item er som sagt en strukturert enhet for inndeling og overføring av medieressurser og metadata innenfor MPEG-21 sitt rammeverk. I MPEG-21 er ressurs det samme som innhold, for eksempel bilde, film eller audio. Metadata er strukturert data om data og i dette tilfellet vil metadataen beskrive medieressursen. I del 2 av MPEG-21 defineres et Item som en gruppering av sub-item eller komponenter som er bundet til relevante databeskrivelser.

Digital Item Declaration beskriver et sett av abstrakte betegnelser og konsepter for å forme en nyttig modell for et Digital Item. Innenfor denne modellen er et Digital Item den digitale representasjonen av "et arbeid", og er slik den enheten som påvirkes innenfor modellen. Den uttrykker og identifiserer ressursen og metadataen. Videre binder DID sammen individuelle og grupper av ressurser og metadata. Målet til en DID-modell er å være så generell og fleksibel som mulig og å være byggestein for videre nivåer i MPEG-21, nivåer som del 7 og del 10.

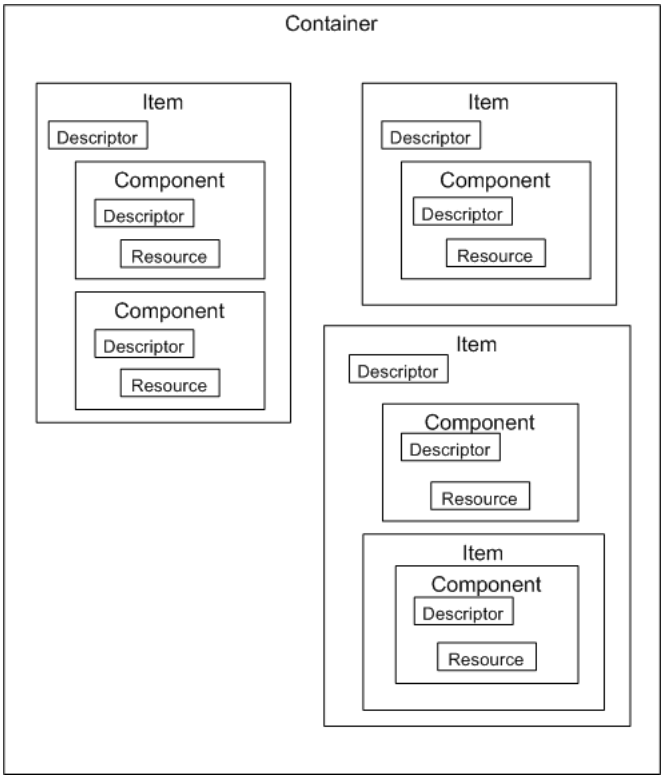
Modellen deles inn i tre seksjoner:

- Modellen: Modellen i dette tilfellet er DID-modellen som beskriver et sett av elementer som skal gi et strukturert Digital Item.
- Representasjonen: Representasjonen er en detaljert beskrivelse av DID-modellen. Den beskriver elementene i DID-modellen.
- Skjema: Skjema definerer den lovlige oppbyggingen av elementene og sammenhengen mellom disse. De implementeres med basis i definerte XML-skjema i XML med Digital Item Declaration Language som språk.

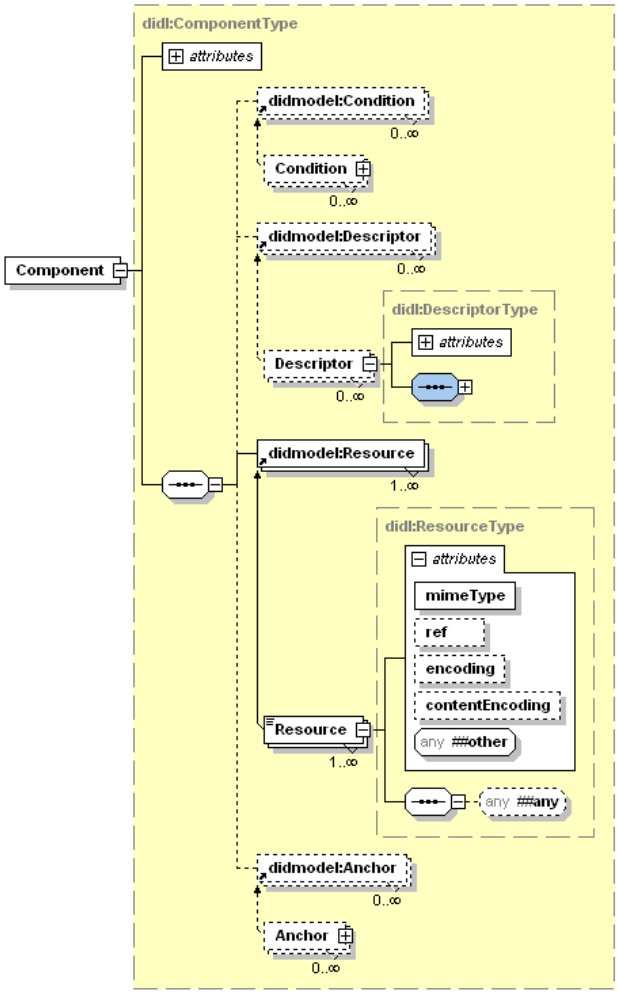
En DID-modell er en abstrakt modell bestående av blokker som er nyttige for å deklarere et DI. Et eksempel på en slik modell er vist i figur 2.6. MPEG-21's del 2 definerer en normativ XML-representasjon av DID-modellen. Denne XML-baserte representasjonen er Digital Item Declaration Language, DIDL.

Eksempel på validert XML-dokument.

Figur 2.7 viser hvordan DID-modellen har definert hvordan Component skal bygges opp. Ved å ikke følge denne modellen får man ikke validert XML-dokumentet og det vil da være ugyldig. Figur 2.8 er hentet fra XML-dokumentet benyttet i diplomoppgaven. Denne følger modellen da Component inneholder Descriptor og Resource. I figur 2.7 ser man at det er påkrevd



FIGUR 2.6: Eksempel på en Digital Item Declaration modell. [14]



Figur 2.7: Figuren viser hvordan DID-modellen for Component er bygd opp.

at Resource er en del av Component, mens Descriptor er frivllig. I dette eksempelet er det også valgt å ta med attributtene mimeType og ref i Resource, dette kan man også se er gyldg fra figur 2.7.

```
<!--===============Norgesbilletten===============-->
<did:Component id="component_norgesbilletten">
  <did:Descriptor>
    <did:Statement mimeType="text/xml">
      <dip:ObjectType>urn:norgesbilletten</dip:ObjectType>
    </did:Statement>
  </did:Descriptor>
  <did:Resource mimeType="text/txt" ref="http://folk.ntnu.no/~vareide/emulatorfiler/norgesbilletten.txt"/>
</did:Component>
```

FIGUR 2.8: Figuren viser et eksempel på hvordan en Component kan bygges opp.

Resource

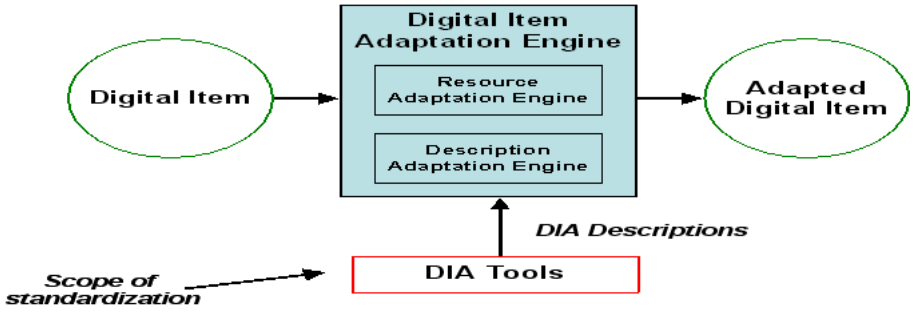
Et Resource-element representerer en ressurs. Den definerer en individuell identifiserbar formue slik som en video eller et lydklipp, et bilde, en elektronisk billett, eller en tekstformue.

Normal er en ressurs i et Resource-element definert ved referanse, ved å spesifisere ressursens Uniform Resource Identifier i attributten ref. URI identifiserer ressursen med den hensikt å tillate programet å gjenfinne ressursinnholdet. URI kan være en tradisjonell URL, som gir den fysiske lokaliseringen for hvor en kan finne innholdet, eller en mer abstrakt identifiserer, slik som en URN, som identifiserer ressursinnholdet uavhengig av lokalisering. URL spesifiseres i ref-attributten, som for eksempel `http://folk.ntnu.no/vareide/emulatorfiler/logoBecool.jpg`.

2.5.2 Del 7: Digital Item Adaptation

MPEG-21's del 7, Digital Item Adaptation, inneholder verktøy for en kontrollerende tilpasning av Digital Item. [13],[4] En beskrivende figur av tilpasningen er vist i figur 2.9.

Ved overføring av multimedia til terminaler, i MPEG-21 kalt "User", må flere viktige parametre vurderes. Informasjon som audio- og videokodeker, skjerm-oppløsning og skjermstørrelse er eksempler på parametre som bestemmer tilpasningen. Alle disse parametrene returnerer verdier som er input i tilpasningen. MPEG-21 standardiserer settet av beskrivelser, og hvordan disse skal overføres mellom avsender og mottaker. I tillegg standardiserer MPEG-21 bruken av deskriptorbiblioteket i DIA.



FIGUR 2.9: Digital Item tilpasningsmaskin. [13]

DIA definerer et verktøy for å overføre informasjon om terminalens muligheter, som kodekmuligheter og inn- og utgangskarakteristikker. Den standardiserer også hvordan man skal kunne beskrive nettverkskarakteristikk, som definerer stasjonære egenskaper hos et nettverk og betingelser hos dynamiske oppførsler. Disse beskrivelsene muliggjør multimediatilpasning for forbedret overføringseffektivitet og brukeropplevelse. Beskrivelser angående brukerkarakteristikker er mange og kan for eksempel være brukerfortrinn, som er et verktøy spesifisert av MPEG-7. Dette verktøyet er en "beholder" av varierte beskrivelser som direkte beskriver fortrinnet til en User. MPEG-7, også kalt Multimedia Content Description Interface, er en standard utviklet spesielt for å beskrive indekser og behandle multimediainnhold.

DIA spesifiserer metadata for å tilpasse et Digital Item, der DIA angir atomiske descriptorer som påvirker (hever/senker) Quality of Service for presentasjon og konsumering av et Digital Item. Målet ved terminal- og nettverks-QoS er å velge optimale parametersettinger for mediaressurs-tilpasningmaskiner som tilfredstiller restriksjoner pålagt av terminalene eller nettverket ved maksimal QoS.

Metadata er mye brukt for å assosiere tilleggsinformasjon til forskjellige

FIGUR 2.10: Verktøy definert av DIA for overføring av terminalinformasjon

typer og samlinger av multimediainnhold. MPEG-21 standardiserer hvordan man knytter denne metadata til en multimediaressurs. Med hensyn på metadatatilpasning finnes det flere viktige bekymringer. Ved tilpasset innhold må også den assosierte metadataen skifte deretter. Hvis metadataen er overført og behandlet kan det være nødvendig å skalere på grunn av terminal- og nettverksrestriksjoner. Filtrering er ofte nødvendig ved store og detaljerte innhold, dette for å oppnå bare de nødvendigste delene av metadataen for brukeren. For tilfellene hvor det eksisterer flere kilder av metadata for den samme ressursen, kan en effektiv utvei være å integrere disse forskjellige kildene av metadata til én enkel beskrivelse.

For å tilpasse DID-forfatterens intensjoner kan DIAs konfigurasjonsverktøy benyttes. Det er essentielt to typer av verktøy som har blitt spesifisert. Det første blir brukt til å angi hvem som skal ta bestemmelsen i tilpasningsprosessen. Det andre blir benyttet for å bestemme om tilpasningen skal vere synlig for brukeren, eller om tilpasningen skal vere automatisert og styrt av applikasjonen. Dette er definert i MPEG-21 DIA under Digital Item Adaption Configuration. DIAC er et verktøy for å styre eller konfigurere hvem som skal gjøre hva, og hvordan, i et scenario hvor DIA benyttes.

2.5.3 Del 10: Digital Item Processing

Digital Item Processing spesifiserer syntaksen og semantikken av verktøy som kan benyttes til å prosessere Digital Item. Verktøyene skaffer et sett med normative verktøy som spesifiserer prosesseringen av Digital Item på en forhåndsdefinert måte.

Vilkår og definisjoner som benyttes i spesifiseringen av DIP er følgende:

- Digital Item Method: DIP spesifiserer DIM som en måte å lage et forslag til interaksjon mellom User og DI. Digital Item Method spesifiserer et sett av verktøy som muliggjør Digital Item Users å legge til forhåndsdefinerte funksjonaliteter på et DI. En slik sekvens av instruksjoner utgjør en Digital Item Method.
- ECMAScript-tilknytninger for Digtal Item Base Operations.
- Java-tilknytninger for Digital Item Base Operations.
- Innhenting av Java-basert DIXO fra Digital Item Method.

Digital Item Method

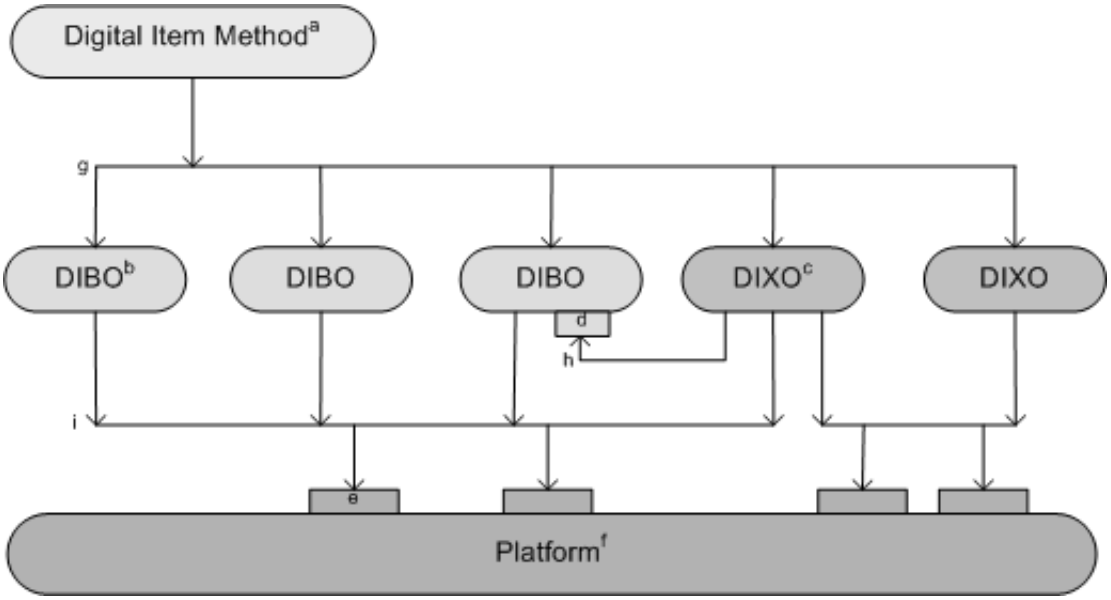
En nøkkel-komponent i Digital Item Processing er Digital Item Method, se figur 2.11. Digital Item Method er et verktøy hvor en User spesifiserer foreslåtte interaksjoner med Digital Item. Slik frembringer Digital Item Method en måte for User å spesifisere et utvalg av foreslåtte prosedyrer for å prosessere en Digital Item på samme nivå som Digital Item selv. For eksempel kan et Digital Item som representerer et musikkalbum inneholde en Digital Item Method for å legge til et nytt musikkspor til albumet. En slik Digital Item Method kan brukes til å forsikre at det nye musikksporet er lagt til Digital Item. Samtidig vil DIM bevare det foreslåtte formatet for Digital Item Declaration for et slikt musikkalbum Digital Item.

Digital Item Methods er ment for å bare jobbe med deler av DI på DID-nivå, de er ikke ment for lavere ressursprosesseringsnivå. Derfor er parametrene til DIM DID Objects som representerer DIDL-elementer knyttet til Object Type av Object Map. Object Type og Object Map er forklart senere i rapporten.

Digital Item Method inkluderer følgende verktøy:

- Digital Item Base Operations - DIBO spesifiserer et høyt nivå av normative samhandlinger til de basise typene av interaksjon med Digital Item. Settet med normative definerte DIBOer har generelle applikasjoner på tvers av et stort område av ressurser, applikasjoner, etc.

- Digital Item Method Language - DIML spesifiserer et normativt språk for definering av interoperable DIM og fra hvordan DIBO er i stand til å kalles. ECMAScript-bindingene av DIBOene er en normativ del av DIML.
- Digital Item Method linket med DID - dette spesifiserer altså den normative mekanismen for inkludering av Digital Item Method i en DID.
- Digital Item Method execution - dette spesifiserer utførelsesmiljøet av en DIM.
- Digital Item eXtension Operation - DIXO spesifiserer en normativ mekanisme for muliggjøring av funksjonaliteter av normative sett av DIBO på en effektiv måte.



FIGUR 2.11: Modell av Digital Item Methods og dens verktøy. [15]

Digital Item Method Lanuage

Digital Item Declaration Language beskrevet i avsnitt 2.5.1 brukes til å lage en statisk deklarering. DIP assisterer prosesseringen av Digital Item ved å skaffe verktøy som tillater User å legge til User-spesifiserte funksjonaliteter til Digital Item Declaration. Standardiseringen av Digital Item Processing muliggjør interoperabilitet på prosesseringsnivå.

Digital Item Method Language er språket som DIM er uttrykt i. DIML-spesifiseringen inkluderer ECMAScript Language Spesification definert av

ISO/IEC 16262:2002. Det henter syntaks og struktur for å forfatte en DIM ved å bruke DIBO /refsec:DigitalItemBaseOperation og DIXO.

Når man lager en DI, brukes DIDL til å deklarere de forskjellige delene av DI. Den resulterende DID er en komposisjon av de forskjellige DID-elementene. Å tillate bruken av disse DID-elementene som argumenter for DIMs, Object Types og Object Map er definert i DIP-spesifikasjonen.

DIP-informasjon

Det finnes to typer av DIP-informasjon inkludert i et DI, DIM-informasjon og Object Type informasjon. DIM-informasjonen kan deles i to deler, Method Information og DIM implementation. Utformeren inneholder all informasjon som er nødvendig for å kunne tillate en DIP-maskin å kalle en DIM. Sistnevnte inneholder den egentlige implementeringen av DIM, f.eks. ECMAScript-kode med kallene til DIBO og DIXO.

- Method Information: MethodInfo er lagret i DID som Descriptor eller Statement inne i Component'en som inneholder DIM-implementeringen. Den benyttes i DIP-maskinene til å lokalisere en DIM i en DID. Derfor er uansett Component som inneholder MethodInfo Descriptor gjenkjent som en Component inneholdende en DIM. MethodInfo Descriptor inneholder en liste av Argument-elementer som indikerer Object Types av argumentene krevd for utførelse av DIM.

  Descriptoren MethodInfo tillater også DIM-forfatteren å signalisere en profil eller profiler til hvordan DIM tilpasser seg. Ved å bruke denne profilinformasjonen kan en DIP-maskin bestemme om den har kapasitet til å utføre DIM eller ikke.

- DIM Implementation: Andre del av DIM-informasjonen inkludert i DID er implementeringen av DIM i DIML. DIM-implementeringen er inkludert som en Resource child av Component inneholdende en MethodInfo Desciptor. Resource kan enten inneholde ECMAScript kode av DIM innlagt i DID, tilsiktet ved å bruke URI i ref-attributten av Resource, eller ved å bruke DID Reference element. DIM-koden er lagret i en såkalt CDATA i XML, CDATA er Complex Data som er unntatt validering. Det vil medføre at XML-dokumentet kan valideres selv om en har informasjon i dokumentet som ikke validerer mot skjema. Dette er tilfellet med DIBO.

Digital Item Base Operation

DIBO er den minste mulige handlingen som kan utføres på et DI. DIBO er funksjoner (av globale og lokale objekter) for hvordan en DIM er bygd og de lager en samhandling mellom DIM- og DIBO-funksjonaliteten laget av DIP-maskinen. Funksjonene inneholder definerte parametre som benyttes i samhandlingen.

En DIBO er beskrevet av:

- et normativt definert grensesnitt
- normative definerte semantikker

Grensesnittet spesifiserer DIBO-navnet, parametrene til DIBO, og om DIBO returnerer en verdi. Semantikken av DIBO spesifiserer funksjonaliteten av DIBO, i tillegg til semantikken av parametrene og de eventuelle returnerte verdiene.

DIBO-parametrene kan inkludere:

- DID-objekter som representerer DIDL-elementer koblet til Object Types. Object Type er en type argument godkjent til en DIM. Den tillater argumentet å bli prosessert innenfor DIM, ifølge dens semantikk. Et Object Map frembringer forholdet mellom DID-elementene til Object Typene. Dette tillater dem å bli brukt som argument i en DIM.
- parametre bundet til en type frembringt av Digital Item Method Language beskrevet i avsnitt 2.5.3.

Fordi om en DIBO har et normativt interface og normative semantikker, betyr ikke dette at DIBOene skal være implementert på en normativ måte. DIBO-implementeringen er overlatt til implementereren av DIBOene.

En liste over de ulike DIBOene er vedlagt i vedlegg /refsec:VedleggA.

Digital Item eXtension Operations

En DIXO er en handling som tillater ikke-standardiserte operasjoner å bli anropt av en DIM. DIXO-mekanismen utfører utvidelse av settet med standardiserte DIBOs på en interoperabel måte.

DIP-maskin

DIP engine er prosesseringsenheten som er ansvarlig for å lage et Object Map, og utførelse av DIM. Utførelse av DIM inkluderer: kalling av DIBOs, lasting av DIXOs, utførelse av DIXOs, og tolkning av DIML-koden.

Gjennomgang av Digital Item Processing:

1. MPEG-21 peer mottar DID. Peer gjenkjenner at den har mottatt en DID ved for eksempel å se på namespace for hvordan xmlkoden er uttrykt. Deretter overføres kontrollen til DID-maskinen.

2. DID-maskinen åpner DID og søker gjennom dokumentet for å finne DIP-informasjon.

3. Siden DID inneholder DIP-informasjon vil DID-maskinen overføre denne informasjonen til DIP-maskinen.

4. DIP-maskinen lager så en liste over DIMene og lager Object Map.

5. Hvis DIP-informasjonen inneholder en DIM ved navn main blir denne DIM automatisk valgt. Main krever ikke argumenter i DIDen. I et main-tilfelle vil gjennomgangen fortsette i punkt 8.

6. Listen over DIMene presenteres til MPEG-21 User. Denne listen av DIMer kan sees på som en meny av mulige prosesser som MPEG-21 User kan kreve fra DIP-maskinen.

7. MPEG-21 User velger en DIM og valget blir overført til DIP-maskinen.

8. DIP-maskinen forsikrer seg om at DIP-implementeringen er lastet og tilgjengelig for utførelse i ECMAScript-miljøet. Hvis det finnes DIXO brukt i DIM, er disse DIXOene nedlastet og initialisert.

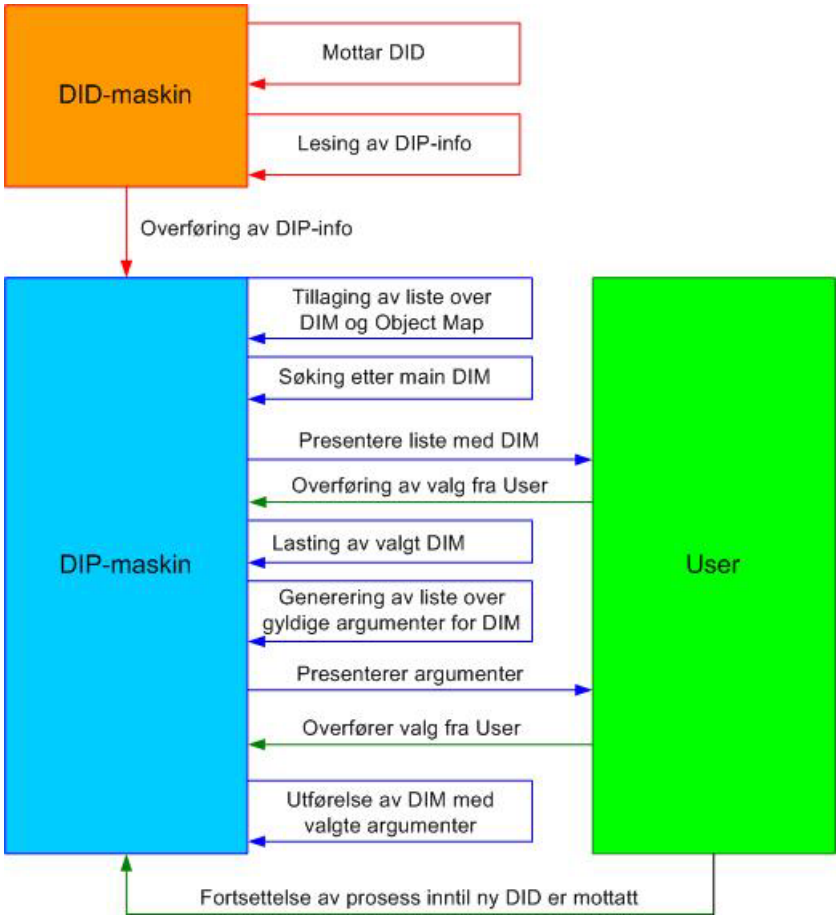
9. Basert på informasjonen presentert i Object Map, genererer DIP-maskinen en liste med argumenter som er gyldig for valgt DIM. Hvis det ikke er noen argumenter fortsetter gjennomgangen i punkt 12.

10. Listen med gyldige argumenter presenteres for MPEG-21 User. Denne listen kan sees på som en meny av mulige elementer for hvordan den valgte DIM kan opptre.

11. MPEG-21 User velger et argument og informerer DIP-maskinen om valget.

12. DIM bli så utført ved å kalle DIBOer og DIXOer listet i DIM-implementeringen. Merk: Utførelsen av DIBOer og DIXOer kan trigge flere rettighets-sjekker.

13. Etter utførelsen av DIM returnerer DIP-maskinen til tilstanden hvor den presenterer listen av DIM til MPEG-21 User.



Figur 2.12: Gjennomføring av DIP

Tilknytning til DID

DIP er knyttes til DID ved å skaffe normative verktøy som gjør at DIP kan inkluderes i et DI. For å lettere kunne skjønne denne tilknytningen kan man sammenligne MPEG-21 teknologien med Extensible HyperText Markup Language. XHTML-dokumenter beskriver strukturen av en webside og man kan legge til interaktivitet til websiden benytter man JavaScript. Samme prinsippet kan legges til et DI. DID beskriver strukturen av DI og for å legge til interaktivitet til DI benytter man DIM.

Siden DIP samhandler med DI er det nødvendig å se på DID for å forstå motivasjonen bak utviklingen av DIP. DID er et statisk Extensible Markup Language (XML) deklarering som inneholder ressurser og metadata enten ved referanse eller direkte i XML-dokumentet. Figuren 2.13 viser et eksemepl på DID som deklarerer en filmliste ved å bruke DIDL-elementer. Gjennom den statiske egenskapen til XML-dokumentet, og derfor et DID, er ikke svaret på spørsmålet: ”Hvordan skal dette DID prosesseres?” så rett fram.

```
<?xml version="1.0" encoding="UTF-8"?>
<did:DIDL xmlns:did="urn:mpeg:mpeg21:2002:02-DIDL-NS">
  <did:Item>
    <did:Descriptor>
      <did:Statement mimeType="text/plain">Min filmsamling</did:Statement>
    </did:Descriptor>
    <did:Component>
      <did:Resource mimeType="video/mpeg" ref="http://folk.ntnu.no/~vareide/emulatorfiler/filmtrailer_robots.mpg"></did:Resource>
    </did:Component>
    <did:Component>
      <did:Resource mimeType="video/mpeg" ref="http://folk.ntnu.no/~vareide/emulatorfiler/filmtrailer_becool.mpg"></did:Resource>
    </did:Component>
  </did:Item>
</DIDL>
```

FIGUR 2.13: Figuren viser et eksempel på XML-dokument uten DIM.

DIM-definisjonen er innlagt i eller referert fra Resource-elementet i DIDL. Et slikt Resource-element skal være innlagt i en DIM-deklarering. DIM er innlemmet i en DID ved å bruke DIP-definerte XML-elementer som er innlagt i passende lokaliseringer innenfor DIDL av DID. W3C XML Schema er brukt for å definere disse DIP-elementene.

DIM-definisjonen selv er en sekvens av operasjoner skrevet i DIML. Dette inkluderer kalling av DIBO for å aksessere funksjonaliteten av User skaffet av Peer sin implementering av DIBO-semantikker.

En DIM-definisjon kan legges inn automatisk i en DID ved hjelp av base64-koding eller ved å plassere DIM-definisjonen i en CDATA del. I denne oppgaven er siste alternativ benyttet og figur 2.14 viser et eksempel på dette.

```
<did:Resource mimeType="application/mp21-method"><![CDATA[
function visFilminfo(){
   var objTypes = new Array("urn:film");
   var msg = new Array("Velg film:");
   var film = DIP.getObjects(objTypes, msg);

   var comp = film[0].getElementsByTagName("did:Component").item(3);

   DIP.play(comp, false);

}]]></did:Resource>
```

FIGUR 2.14: Eksempel på Resource med CDATA.

2.6 QuickTime

QuickTime-spilleren fra Applet var et naturlig valg siden det er knyttet APIer sammen med spilleren. Disse APIene er et grensesnitt mot Java og lar User integrere QuickTime-spilleren i Javaprogrammet. Hele funksjonaliteten av QuickTime-spilleren vil være inkludert.

2.6.1 QuickTime for Java

Pakken QuickTime for Java kobler QuickTime og Java sammen. Utviklere som lager Java-software kan bruke kjennetegn av QuickTime i deres applikasjon. Det er to lag i QuickTime for Java:

- Core Layer - gir tilgang på API for QuickTime
- Application Framework Layer - gjør det enkelt for Java-applikasjoner å integrere kjennetegnene på QuickTime. Integreringen med Java Runtime er her sammen med delingen av hendelser mellom disse to plattformene.

I API knyttet til QuickTime for Java er programmereren gitt metoder for å få full kontroll og å sette opp denne spilleren i et Java-miljø. Programmereren kan velge hvilke komponenter som skal være synlig på skjermen, sette hastigheten for spilleren og velge type spiller.

2.7 3G

3rd Generation Partnership Project, 3GPP, er et samarbeidsprosjekt som ble etablert i desember 1998. [1] Til et 3.generasjons mobilsystem skal 3GPP fremstille globalt tilpassede tekniske spesifikasjoner og rapporter for audiovisuelt innhold. De har definert et filformat beregnet for mobile terminaler. 3gp-filformatet spesifiserer en måte å organisere forhåndslagret kodestrøm

slik at filen blir tilgjengelig for lokal dekoding og avspilling, overført til et annet media.

3G står for 3. generasjon, en generell, trådløs bransjebetegnelse for høyhastgihets mobil datalevering over mobilnettverk.[25] 3G-nettverk tillater bruker å sende og motta båndbreddeintensiv informasjon slik som fullbegevegelse-video, videokonferanse, høykvalitetets lyd og webdata anfordret virtuell "anytime anywhere".

3.generasjons mobiltelefoni blir ofte omtalt som en samlebetegnelse for høyere datahastighet i mobilnettet. [19] 3G-tjenester er tjenester som krever høyere datahastighet for bedret brukeropplevelse. Eksempler på 3G-tjenester kan være mobil-TV og videosamtaler.

2.7.1 3G i forhold til MPEG-21

MPEG-21 har verktøy som skalerer innholdet etter en har detektert at terminalen har 2G, 2.5G eller 3G. Dette kan gjøres ved hjelp av for eksempel DIBOen adapt. Se [15] for mer informasjon.

2.8 UMTS

UMTS er et helt nytt mobilnett, som gjør at data kan overføres 8-10 ganger raskere enn i GSM-nettet. UMTS er en forkortelse for Universal Mobile Telecommunications System. UMTS er en global standard, noe som betyr at det på sikt vil være mulig å benytte 3G-tjenester i de landene som tilbyr UMTS. Innføring av UMTS representerer et stort sprang fremover når det gjelder overføringskapasitet og tjenestemuligheter. Nedlastingshastigheten i UMTS er teoretisk 2 Mbps. I praksis vil UMTS gjennomsnittlig gi gjennomsnittlig fra 144 kbps ved hurtig bevegelse av termianlen, mens det vil gi opp til 384 kbps når brukeren beveger seg med tilnærmet gåhastighet. Opplastingshastighet er opp til 64 kbps.

Filmweb vil være en tjeneste som ikke er avhengig av UMTS-dekning, men brukeropplevelsen vil bli bedre ved UMTS-dekning da man her har høyere overføringshastighet enn ved vanlig GSM-dekning.

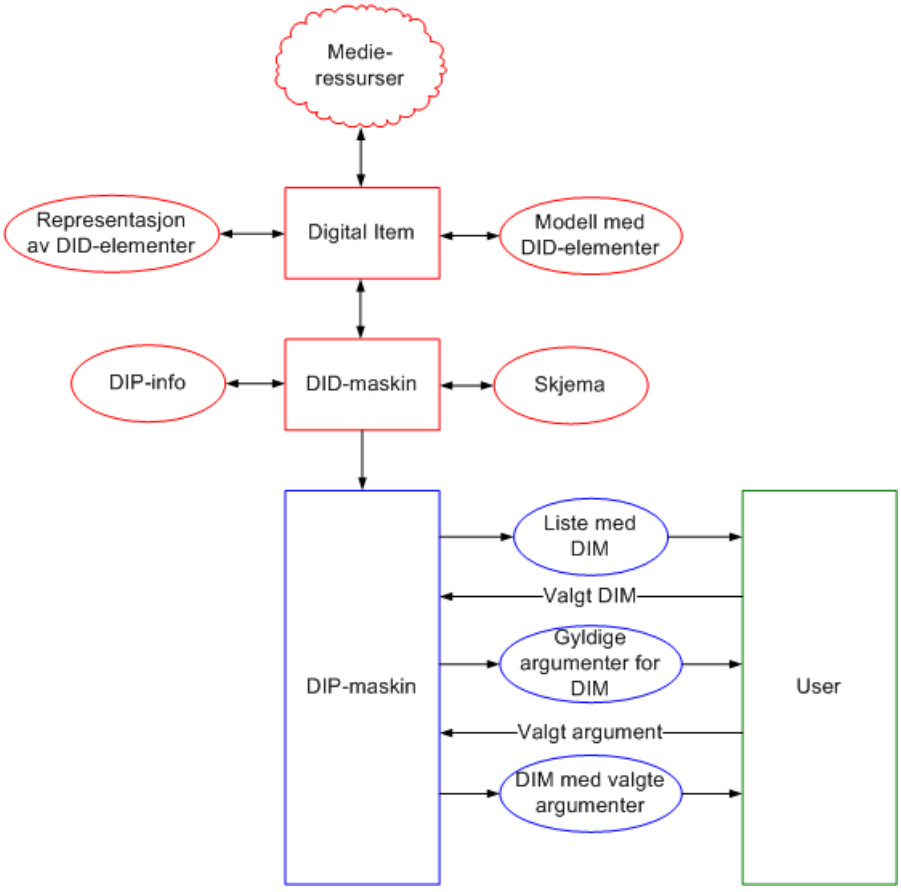
Kapittel 3

Resultater

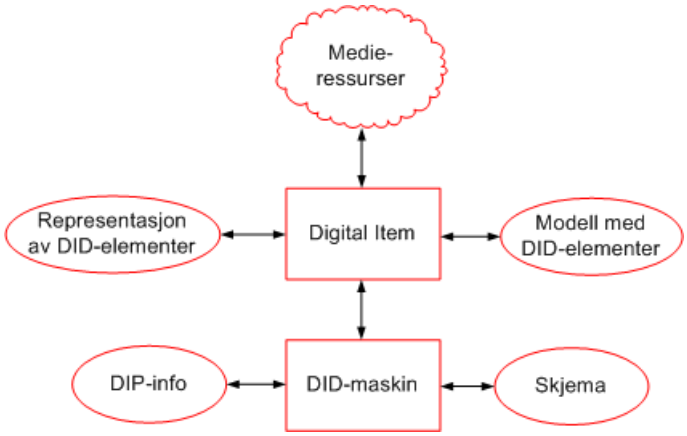
Resultatet i denne oppgaven er en kombinasjon av Digital Item Declaration og Digital Item Processing sine verktøy. XML har blitt benyttet som formateringsverktøy og Java er brukt som kjøremiljø. Implementert Javakode og verktøyene fra DID og DIP har resultert i en funksjonalitet som er tilgjengelig for både innholdsleverandør og innholdsforbruker.

3.1 Implementasjon

Ved implementasjon er det tatt utgangspunkt i DIP-maskinen beskrevet i avsnitt 2.5.3. Figur 3.1 beskriver sammenhengen mellom de tre hoveddelene DID-maskin, DIP-maskin og User.



FIGUR 3.1: Oppbygging av implementasjon i diplomoppgave.



FIGUR 3.2: Oppbygging av DID-maskin

3.1.1 Digital Item

Digital Item er et XML-dokument bygd opp som en DID-modell med DIDL som språk, se vedlegg E.1. Dette er enheten for overføring av de digitale mediaressursene med tilhørende metadata. De digitale medieressursene er i dette tilfellet tekst, bilde og film. Det fremstilte Digital Item er bygd opp etter betegnelsene og konseptene som DID definerer. XML er benyttet siden MPEG-21 del 2 har definert et formateringsspråk i XML. Digital Item er validert mot MPEG-21 skjema.

Første del av Digital Item inneholder en Descriptor med Resource som inneholder en URL til Filmweb sin logo. Dette beskriver Digital Item. Siden medieressursene er lagret eksternt må URL benyttes som referanse.

Digital Item består av tre Item og to Component. To av Item inneholder hver sin film og da i dette tilfellet filmene Robots og BeCool. Item for BeCool er vist i figur 3.3. Item for hver film er bygd opp helt likt hvor første del av Item er en Descriptor med beskrivelse av filmene.

Item inneholder tre Descriptor hvor den første er en Component med Resource av et ikon for filmen. Ikonet ligger som en URL i Resource og med mimeType ”image/jpeg” som beskriver at dette er et bilde av type jpeg. Neste Descriptor inneholder et Statement med mimeType ”text/xml”. Statement består av en ObjectType av typen ”urn:bilde”. Siste Descriptor inneholder Statement med mimeType ”text/plain”. Det vil si at Statement er en vanlig tekst og i dette tilfellet er dette navnet på filmen.

Videre inneholder Item flere Component som er diverse medieressurser tilknyttet filmen. Medieressursene er i dette tilfellet billettbestilling, filmtrailer, filminformasjon og filmomtale. Hver av Component inneholder hver sin De-

```
<!--====================BeCool====================-->
<did:Item id="becool">
  <did:Descriptor>
    <did:Component>
      <did:Resource mimeType="image/jpeg" ref="http://folk.ntnu.no/~vareide/emulatorfiler/logo_becool.jpg"/>
    </did:Component>
  </did:Descriptor>
  <did:Descriptor>
    <did:Statement mimeType="text/xml">
      <dip:ObjectType>urn:film</dip:ObjectType>
    </did:Statement>
  </did:Descriptor>
  <did:Descriptor>
    <did:Statement mimeType="text/plain">BeCool</did:Statement>
  </did:Descriptor>
  <did:Component id="becool_billett">
    <did:Descriptor>
      <did:Statement mimeType="text/xml">
        <dip:ObjectType>urn:billett</dip:ObjectType>
      </did:Statement>
    </did:Descriptor>
    <did:Resource mimeType="text/txt" ref="http://folk.ntnu.no/~vareide/emulatorfiler/billettbestilling.txt"/>
  </did:Component>
  <did:Component id="becool_trailer">
    <did:Descriptor>
      <did:Statement mimeType="text/xml">
        <dip:ObjectType>urn:filmtrailer</dip:ObjectType>
      </did:Statement>
    </did:Descriptor>
    <did:Resource mimeType="video/mpeg" ref="http://folk.ntnu.no/~vareide/emulatorfiler/filmtrailer_becool.mpg"/>
  </did:Component>
  <did:Component id="becool_kinoprogram">
    <did:Descriptor>
      <did:Statement mimeType="text/xml">
        <dip:ObjectType>urn:filminfo</dip:ObjectType>
      </did:Statement>
    </did:Descriptor>
    <did:Resource mimeType="text/txt" ref="http://folk.ntnu.no/~vareide/emulatorfiler/filminfo_becool.txt"/>
  </did:Component>
  <did:Component id="becool_omtale">
    <did:Descriptor>
      <did:Statement mimeType="text/xml">
        <dip:ObjectType>urn:filmomtale</dip:ObjectType>
      </did:Statement>
    </did:Descriptor>
    <did:Resource mimeType="text/txt" ref="http://folk.ntnu.no/~vareide/emulatorfiler/filmomtale_becool.txt"/>
  </did:Component>
</did:Item>
```

FIGUR 3.3: Figuren viser Item for filmen BeCool fra Digital Item.

scriptor med spesifisert ObjectType.

De to Component i Digital Item er Norgesbilletten og Filmweb nyhetsbrev. Filmweb Nyhetsbrev består av en Descriptor og en Resource. Descriptor er et Statement med ObjectType ”urn:nyhetsbrev”. Resource er tom, men må være med da dette er definert i DID-modellen. Hvis denne utelukkes vil ikke XML-dokumentet være gyldig. Se eksempel på Component i figur 3.4.

```
<!--===============Norgesbilletten===============-->
<did:Component id="component_norgesbilletten">
  <did:Descriptor>
    <did:Statement mimeType="text/xml">
      <dip:ObjectType>urn:norgesbilletten</dip:ObjectType>
    </did:Statement>
  </did:Descriptor>
  <did:Resource mimeType="text/txt" ref="http://folk.ntnu.no/~vareide/emulatorfiler/norgesbilletten.txt"/>
</did:Component>
```

FIGUR 3.4: Component for Norgesbilletten.

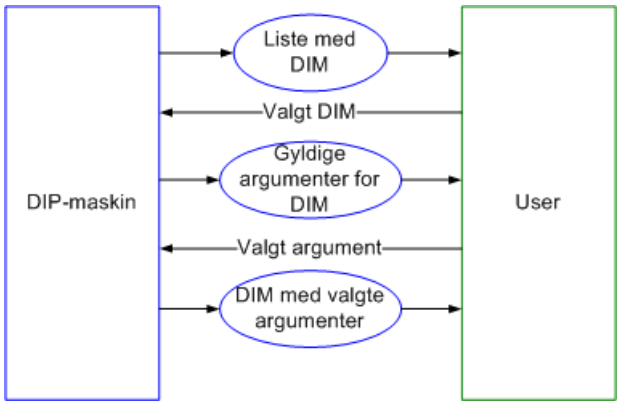
Component Norgesbilletten består også av en Descriptor og en Resource. Descriptor er også her et Statement, men inneholder ObjectType ”urn:norgesbilletten”. Resource inneholder URL til en tekstfil for Norgesbilletten og mimeType ”text/txt”, som viser at dette er en tekstfil av type txt.

Tredje og siste Item består av seks DIM. En DIM var som sagt et verktøy hvor en User spesifiserer foreslåtte interaksjoner med Digital Item. Den inneholder dermed informasjon for hvordan medieressursene skal prosesseres. Medieressursene det da er snakk om er Item og Component nevnt over. Dette er beskrevet nærmere i avsnitt 3.1.2.

Ved å bygge opp Digital Item på denne måten har det resultert i at man har inkludert medieressurser, metadata og prosesseringsmetode i én enhet. Det vil si at for overføring til en klient trenger man kun sende en fil. For eksempel kan man sende medieressurs som er en film og prosesseringsinstruksjonene i samme fil.

3.1.2 Digital Item Method

For å kunne benytte DIBO i Digital Item Processing behøves en ECMAScriptmotor. GUI er benyttet da J2ME på mobiltelefoner eller emulatorer ikke inneholder ECMAScriptmotor. GUI står for Graphical User Interface, som kan oversettes med grafisk brukergrensesnitt og er en illustrasjon av hvordan tjenesten ville sett ut på mobiltelefon. Et GUI implementert for PC er benyttet å presentere innholdet i 3G-tjenesten. Implementasjon av ECMAScriptmotor, DIBO og GUI er implementert i Java. UML-diagram for klassene er vist i vedlegg D.1, mens Javakode ligger i vedlegg E.1



FIGUR 3.5: Oppbygging av DIP-maskin

Alle DIM ligger i et Item. Først i Item ligger en Descriptor med Statement. Dette Statement inneholder en tekst, som skal benyttes til overskrift. Tabell med alle DIM fra XML-dokumentet genereres, et eksempel på DIM er vist i figur 3.6. Tabellen lages ved å plukke ut alle Descriptor i Component for DIM. I 3G-tjenesten er det valgt å ha en Descriptor med Resource som inneholder en URL til et ikon som beskriver DIM. Overskrift sammen med tabell av DIM presenteres for User. User er i dette tilfellet en person. Presentasjonen av DIMtabell og overskrift gjøres i GUI. Overskriften forteller User at det må gjøres et valg mellom DIM. User velger DIM og dette valget returneres til DIP-maskin.

```
<!--DIM-filmtrailer-->
<did:Component id="component_dim_filmtrailer">
  <did:Descriptor>
    <did:Component>
      <did:Resource mimeType="image/jpeg" ref="http://folk.ntnu.no/vareide/emulatorfiler/trailer.jpg"/>
    </did:Component>
  </did:Descriptor>
  <did:Descriptor>
    <did:Statement mimeType="text/plain">Trailer</did:Statement>
  </did:Descriptor>
  <did:Descriptor>
    <did:Statement mimeType="text/xml">
      <dip:MethodInfo/>
    </did:Statement>
  </did:Descriptor>
  <did:Resource mimeType="application/mp21-method"><![CDATA[
    function visFilmtrailer(){
      var objTypes = new Array("urn:film");
      var msg = new Array("Velg film:");
      var film = DIP.getObjects(objTypes, msg);

      var comp = film[0].getElementsByTagName("did:Component").item(2);
      DIP.play(comp, false);
  }]]></did:Resource>
</did:Component>
```

FIGUR 3.6: Eksempel på DIM fra Digital Item.

Foreløpig har Digital Item bare inneholdt bilde og tekst. MimeType har fortalt klienten hvilken type medieressurs som overføres. For tekst har dette vært ”text/plain”, mens bildene beskrives av ”image/jpeg”. Ved tjenester som inneholder mange forskjellige typer gjør DID-modellen det enkelt for klient å vite hvilken type medieressurs som skal presenteres.

Når User har gjort sitt valg henter DIP-maskinen ut Resource i DIM. Det som skjer da er at Resource forteller hvordan programmet skal prosessere videre. Dette gjøres ved hjelp av DIBO-funksjoner i MPEG-21 definert som Digtial Item Method Language, som nevnt i avsnitt 2.5.3. ECMAScript-tkode ligger som CDATA i Resource med mimeType ”application/mp21-method”. ECMAScriptkoden tolkes i en ECMAScriptmotor. Kildekoden for ECMAScriptmotoren er hentet fra Mozilla og er vist i figur 3.7. [20] Mozilla er blant annet en åpen kildegruppe for utviklerere og testere. Mozilla Rhino er en åpen kildeimplementering av ECMAScript skrevet i Java. [21] Det er typisk innebygd i Java-applikasjoner for å skaffe script til sluttbrukere.

```
public void executeFunction(Element scriptResource) {

    String script = scriptResource.getTextContent();

    Context cx = Context.enter();

    System.out.println("Attempting to execute function:\n" + script);

    Function f = cx.compileFunction(scope, script, "<cmd>", 1, null);

    Object result = f.call(cx, scope, f, new Object[]{new Integer(10)});

    //Convert the result to a string and print it.
    System.err.println("Result: " + cx.toString(result));

    close();
}
```

FIGUR 3.7: Kildekode for ECMAScriptmotor hentet fra Javakode i vedlegg E.1

MPEG-21 har som tidligere nevnt, forhåndsdefinert flere DIBO som er funksjoner for prosessering av Digital Item. I denne oppgaven er det benyttet tre forskjellige DIBO til prosessering, de DIBOene er getObjects, getObjectMap og play. Detaljert forklaring på disse DIBOene er vedlagt i vedlegg A.1.

getObjects

GetObjects er benyttet til å lage en liste over de filmene som er tilgjengelig i Digital Item. Denne funksjonen fungerer slik at den lager en tabell over flere ObjectType, og for hver ObjectType vises en beskjed. Eksemplet i figuren

viser hvordan dette kan gjøres for én type ObjectType. Metoden i Javakoden er laget kun for et slikt eksempel. I denne tjenesten lager altså getObjects en tabell over alle medieressursene som inneholder ObjectType "urn:film" og for denne ObjectType vises teksten "Velg film:". Eksempel på getObjects er vist i figur 3.8.

```
var objTypes = new Array("urn:film");
var msg = new Array("Velg film:");
var film = DIP.getObjects(objTypes, msg);
```

FIGUR 3.8: Eksempel på getObjects hentet fra Digital Item.

getObjectMap

GetObjectMap henter alle Component med ObjectType og lagrer i en vektor. ObjectMap inneholder flere funksjoner definert i MPEG-21. I oppgaven har det blitt laget Javametoder for noen av disse funksjonene:

- getNumberOfTypes() - antall typer av ObjectType som finnes i XML-dokumentet
- getTypeName(int index) - navnet på ObjectType for ønsket indeks.
- getNumberOfObjects(String TypeName) - antall ObjectType i XML-dokumentet for den bestemte ObjectType
- getObject(String typeName, int index) - finner Component for bestemt ObjectType og indeks
- getObjects(String typeName) - returnerer en liste med alle objektene av bestemt ObjectType

Videre er det laget et eksempel på hvordan getObject fra ObjectMap kan benyttes til å hente akkurat et konkret objekt, se figur 3.9.

```
function visNorgesBilletten(){
   var objectMap = DIP.getObjectMap(didDocument);
   var object = objectMap.getObject("urn:norgesbilletten", 0);

   DIP.play(object , false);
```

FIGUR 3.9: Eksempel på getObjectMap hentet fra Digital Item.

play

DIBOen play benyttes videre til å spille av objektet. I tillegg til objektet trenger play å vite om det skal spilles asynkront eller ikke. Dette gjøres ved å sette true eller false. Play spiller av medieressursen i denne diplomen i et eget vindu. Avhengig av hvilken type medieressurs det er åpnes forskjellige typer vinduer. Filmtrailere trenger en avspiller og til dette er det brukt QuickTime for Java.

For å sjekke hvilken filtype det er, benyttes også her mimeType. I oppgaven er det kun benyttet mpeg-filer og mimeType for filmtrailerne er derfor "video/mpeg". Dette kan også benyttes av andre klienter. Klienten kan sjekke hvilken filtype det er og sjekke om avspilleren støtter denne. Hvis ikke den støtter filtypen, kan man benytte DIBOen *alert*. Denne returnerer tekst til bruker og man kan eventuelt legge inn en feilmelding.

Play returnerer status fra avspilleren. Ved ingen avspilling eller stop returneres status RELEASED. Ved pause returneres STATICPLAY, mens det ved avspilling returneres TIMEPLAY. Dette kan for eksempel benyttes til å sjekke avspilleren og feilmeldinger.

GUI-presentasjonene

Hovedsiden for tjenesten er som sagt en presentasjon av Descriptor i alle DIM i tillegg til logo for Filmweb og overskrift. Knappen "Velg" går videre i menyen, mens "Avslutt" lukker hele vinduet. Nederst i vinduet er en statuslinje. Hovedvinduet er vist i figur 3.10.

FIGUR 3.10: Hovedvindu for mobiltjenesten.

Videre i menyen åpnes et nytt vindu, hva som kommer frem da er avhengig av hvilket valg du gjør. Velger man ”Kjøp billett”, ”Trailer”, ”Kinoprogram” eller ”Omtale” får man frem vinduet vist i figur 3.11. Her også kommer man videre i menyen hvis man trykker på knappen ”Velg”, mens man avslutter ved å trykke på knappen ”Avslutt”.

FIGUR 3.11: Vindu for videre valg i menyen.

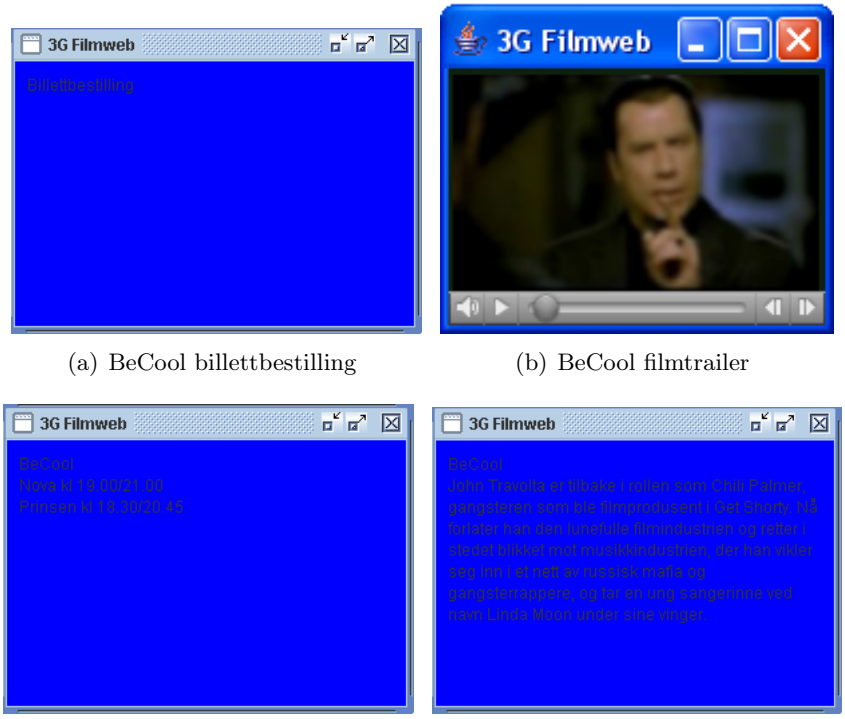
Figur 3.12 viser DIMvalgene for filmen BeCool. Figurene 3.12(a), 3.12(c) og 3.12(d) er tekstvindu, mens figur 3.12(b) er QuickTime-spiller for Java.

Ved å velge Norgesbilletten eller Filmweb nyhetsbrev får man frem vinduene vist i figur 3.13.

3.2 Funksjonalitet av implementasjon

Med denne implentasjonen kan innholdsleverandører kontollere hvordan innhold er prosessert på klientside i mye større grad enn hva som er tilgjengelig i dag. Til en viss grad er denne funksjonaliteten helt ny. For en innholdsleverandør som Filmweb, som kan gå til filmdistributørene og garantere en så kontrollert multimedialeveranse, vil en potensielt kunne øke tilgangen på tilgjengelig innhold. En slik innholdsflyt er bevist mulig med denne imlementasjonen.

Apple, Envivio, Harris, Real, Snell og Wilcox, Vidiator og Nextreaming er eksempler på programleverandører for distribuering av multimedieinnhold. En må i dag sitte manuelt å kode innhold ved hjelp av videokodingsverktøy og flytte det manuelt ut på streamingservere. Dette innholdet blir så tilordnet en URL og lagt tilgjengelig på en webside. Der stopper kontrollen med innholdet. Link til video blir flytta inn i videoplayer på mobiltelefon gjennom at server også sender ved mimetype til innholdet. Det forteler telefonen at innholdet skal presenteres av for eksempel videoplayer på telefonen.



(a) BeCool billettbestilling (b) BeCool filmtrailer

(c) BeCool kinoprogram (d) BeCool omtale

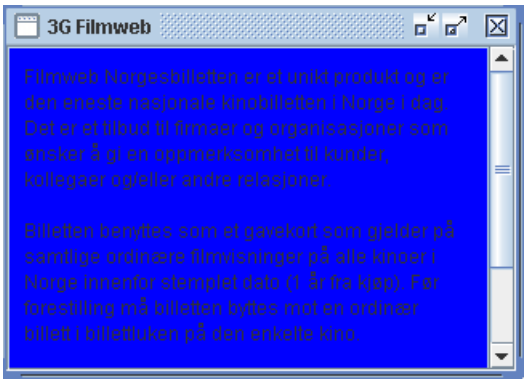
FIGUR 3.12: DIMvalg for BeCool.

Mimetypen blir sendt som en del av htmlinformasjonen. Det vil si hver gang webserveren, som sender videoen eller link til videoen, har ei fil av for eksempel type .3gp sender den også med mimetypen video/3gpp som et argument. I et wap/html-miljø er en ikke sikret at klienten kan tolke nødvendige data.

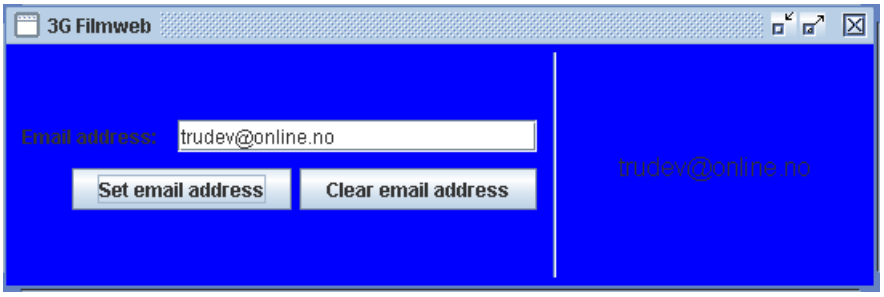
MPEG-21 er et nyttig rammeverktøy for distribuering av medieressurser. DID-modellen er strukturert og gir en god oversikt. For en innholdsleverandør er det enkelt å bruke modellen til å frembringe ønsket informasjon. Ved å benytte elementene i DID-modellen kan man enkelt lage et Digital Item. Det kan lages en grunnstruktur hvor en innholdsleverandør kan fylle inn ønsket informasjon. Ved endringer av Digital Item gir den strukturerte modellen også rom for dette.

MPEG-21 har definert mange vektøy i DIP som gir en kontrollert multimedialeveranse. Implementasjonen åpner for bruk av disse verktøyene i Digital Item Processing.

I implementasjonen over er det kun benyttet deler av funksjonene. Eksempel på hvordan disse funksjonene kan benyttes ser man ut fra tjenesten. Ved å benytte DIBO *areConditionsSatisfied* blir overføringen mer kontrollert da man kan sjekke om overføringen er vellykket. Hvis dette ikke er tilfellet har



(a) Vindu for informasjon om Norgesbilletten



(b) Vindu for Filmweb Nyhetsbrev

FIGUR 3.13: GUI for Norgesbilletten og Filmweb nyhetsbrev

man muligheten til å gi brukeren tilbakemelding, dette kan man benytte *alert* til.

Ved å benytte funksjoner i DIP tilknyttet lisens kan man implementere dette i mobiltjenesten. Disse funksjnene vil være *getLicense* og *queryLicenseAuthorization*. Disse kan benyttes til sjekke om User har lisens til å kunne motta innholdet. Hvis det er ønskelig med innhold som User må betale for kan en slik funksjon benyttes til å sjekke om User er villig til å betale for innholdet.

Ved å benytte Digital Item Processing kan innholdsleverandør enkelt prossere medieressursen. Innholdsleverandør velger ut fra prosseringsverktøyene hvordan man ønsker å fremstille informasjonen. En Javakode kan lages på et generelt grunnlag og man kan bruke denne til flere type tjenester. Ved spesielle tjenester kan man selvfølgelig tilpasse Javakoden til dette.

Ved å benytte en slik implementasjon vil det være enklere for en innholdsleverandør å distribuere innhold. Da implementasjonen gir stor kontroll trenger ikke leverandøren å tenke på for eksempel filformat og skjermoppløsning. Det som trengs er medieressurser med metadata og funksjoner

for prosessering. Ved å gjøre distribusjon av innhold enklere for innholdsleverandør gir dette leverandøren mer arbeidsflyt.

Kapittel 4

Diskusjon

Ved bruk av MPEG-21 får man en kontrollert overføring og vellykket prosessering da den tilpasser innhold til klienten. Ved å benytte del 2 av standarden får man et oversiktlig Digital Item som gjør det enkelt for innholdsleverandørere å lagre og å redigere multimediaressurser. Del 10 av standarden inneholder funksjoner som innholdsleverandører kan benytte til å prosessere multimedieressursene.

På et mer konseptuelt nivå er MPEG-21 nyttig da dette er et åpent rammeverk, dette skaffer innholdsleverandør og tjenesteforsørger lik mulighet da det er åpent for alle. I forhold til innholdsforbruker vil det gi dem et større tilgang innhold.

Ved utprøving av andre lignende tjenester ser man fort både styrker og svakheter. Tilgjengelighet er en klar fordel i denne sammenheng. Med dagens utviklingshastighet innen mobilteknologi både når det gjelder skjermkvalitet og prosessorkraft øker tjenestetilbydernes muligheter stadig. Dette igjen fører til bedre tjenester til bruker. Svakheter er at man ofte opplever at tjenestene ikke fungerer slik som de skal. Medieressursene kommer ikke frem på mobiltelefonen og man får ingen tilbakemelding om hvorfor dette skjer. Ved å benytte en tilsvarende implentasjon som i denne oppgaven unngår man disse svakhetene.

J2ME i mobiltelefonene inneholder ikke ECMAScriptmotor som trengs for å kunne bruke DIBO i Digital Item Processing. I tillegg er ikke minnekapasiteten stor nok i mobiltelefonene. For at disse funksjonene skal kunne tas i bruk til en mobiltjeneste trengs raskere og mer minnesterke enheter eller portering fra J2SE til J2ME. Slik som den teknologiske utviklingen går kan dette være mulig i nær fremtid.

Ved i tillegg å implementere del 7, Digital Item Adaptation, i Digital Item tillater innholdsforsørgeren å kontrollere kapasiteten hos klient for å kunne

motta innholdet. Dette muliggjør kontrollert multimedialevering, hvor innholdsfrembringeren alltid kan kontrollere hvilken type innhold som er mulig å motta på mobiltelefonen.

Kapittel 5

Konklusjon

Oppgaven resulterte i en programkode som presenterer innholdet av en Digital Item i GUI. MPEG-21 ble benyttet til å lage Digital Item, til dette ble del 2 av standarden, Digital Item Declaration, benyttet. Del 10 av standarden, Digital Item Processing, ble benyttet til prosseseringen av Digital Item. MPEG-21 gir innholdsleverandører for mobiltjenester en strukterert og kontrollert måte i distribuere multimedieinnhold på.

Ved å lage en generell Javakode for Digital Item kan innholdsleverandøren kjapt og enkelt endre Digital Item for å presentere nye multimedieressurser. For Filmweb kan man enkelt legge inn Item for filmene og med DIBOene fra Digital Item Processing prossesere Digital Item.

Kapittel 6

Fremtidig arbeid

Starten for fremtidig arbeid for implementasjonen vil være å teste ut flere DIBOfunksjoner fra Digital Item Processing.

Digital Item Processing inneholder også funksjonene DIXO. DIXO er en handling som tillater ikke-standardiserte å bli anropt av DIM. Ved å implementere dette kan man for eksempel legge inn reklame, som kan finansiere tjenesten. Et eksempel på hvordan det kan gjøres er å legge inn reklame som er filmavhengig. Det vil si at når en type film spilles kommer det opp en bestemt reklame, men en annen film presenterer en annen reklame.

Filmweb.no loggfører i dag all bruk på internettsiden deres. Det er viktig for dem å ha et register så de kan holde oversikt over kinonorge. Ved å loggføre Selections kan man generere statistikk som vil gi Filmweb en oversikt over bruken av 3G-tjenesten.

Bibliografi

[1] 3GPP. `http://www.3gpp.org`, 2005.

[2] O. I. Hillestad A. Perkis, P. Drege. Uma enabled environment for mobile media using mpeg-21 client and server technology. *Workshop on Image Analysis for Multimedia Interactive Services*, 2005.

[3] P. Drege A. Perkis, R. Vekatesh. A qos-driven uma viewer in mpeg-21 framework. *European Signal Processing Conference*, 2004.

[4] C. Timmerer A. Vetro. Digital item adaptation: Overview of standardization and research activies. *IEEE TRANSACTIONS ON MULTIMEDIA VOL. 7, NO. 3*, 2005-06.

[5] C. Timmerer A. Vetro. Introduction to the special section on mpeg-21. *IEEE TRANSACTIONS ON MULTIMEDIA VOL. 7, NO. 3*, 2005-06.

[6] P. Borgersen. *Dynamiske websider*. Gyldendal Akademisk og Stiftelsen TISIP, Høgskolen i Sør-Trøndelag, Trondheim, Norge, 2003.

[7] A. K. Dahl. `http://www.hardware.no/artikkel/17131`, 2005.

[8] ecma international. Ecmascript for xml (e4x) specification. *Standard ECMA-357, 2*[1] *Edition*, 2005. `http://www.ecma-international.org/publications/files/ECMA-ST/Ecma-357.pdf`.

[9] R. V. D. Walle F. D. Keukelaere, S.D. Zutter. Mpeg-21 digital item processing. *IEEE TRANSACTIONS ON MULTIMEDIA VOL. 7, NO. 3*, 2005-06.

[10] S. Holzner. *Inside XML*. New Riders Publishing, United States of America, 2001.

[11] G. M. Drury I. S. Burnett, S. J. Davis. Mpeg-21 digital item declaration and identification-principles and compression. *IEEE TRANSACTIONS ON MULTIMEDIA VOL. 7, NO. 3*, 2005-06.

[1] nd

[12] T. Lookabaugh J. D. Gibson, T. Berger. *Digital Compression for Multimedia*. Morgan Kaufmann Publishers, San Francisco, CA 94104-3205, 1998.

[13] ISO/IEC JTC1/SC29/WG11. Information technology - multimedia framework(mpeg-21)-del7: Digital item adaptation. *ISO, Geneva*, 2004-03-19.

[14] ISO/IEC JTC1/SC29/WG11. Information technology - multimedia framework(mpeg-21)-del2: Digital item declaration. *ISO, Geneva*, 2005-04-17.

[15] ISO/IEC JTC1/SC29/WG11. Information technology - multimedia framework(mpeg-21)-del10: Digital item processing. *ISO, Geneva*, 2005-07-25.

[16] Y. Lamo. `http://home.hib.no/AI/data/master/mod250/xml/xml.pdf`, 2005.

[17] B. McLaughlin. *Java og XML*. OReilly og Associates, Sebastopol, CA 95472, 2001.

[18] J. Hunter med W. Crawford. *Java, Servlet Programming*. O'Reilly and Associates, 2001.

[19] Telenor Mobil. `http://www.telenormobil.no`, 2005.

[20] Mozilla. `http://www.mozilla.org/`, 2006.

[21] Mozilla. `http://www.mozilla.org/rhino/`, 2006.

[22] K. Tamura N. H. Maruyama. *XML and Java, Developing Web Applications*. Addison Wesley Longman, 1999.

[23] E. Lervik og V. B. Havdal. *Programmering i Java*. Stiftelsen TISIP og Gyldendal Akademisk, 2003.

[24] A. Perkis P. Drege, T. Skjolberg. Mpeg-21 enabled client for rich mobile content delivery. *Pervasive Signal Processing Conference and Expo*, 2005.

[25] QuickTime. `http://www.apple.com/quicktime/technologies/3gpp/faq.html`, 2006.

[26] G. Roberts R. Winder. *Developing JAVA Software*. John Wiley and Sons, 2000.

[27] Y. M. Ro T. C. Thang, Y. J. Jung. On some qos aspects of user profile in universal multimedia access. *Multimedia Group, Information and Communications University, Yuseong, Daejeon, 305-732, Korea*, 2003.

[28] W3C. `http://www.w3.org/TR/xpath`, 2006.

[29] B. Wragg M. Paramasivam og C. Barlas X. Wang, T. DeMartini. The mpeg-21 rights expression language and rights data dictionary. *IEEE TRANSACTIONS ON MULTIMEDIA VOL. 7, NO. 3*, 2005-06.

Tillegg A

A.1 Digital Item Base Operations

Benyttede DIBO i diplomoppgaven.

Syntaks	*Beskrivelse*	*Parametre*	*Returverdi*
getObjectMap(document)	Mottar ObjectMap fra et DID-document	document	ObjectMap
getObjects(objectTypes, requestMessages)	Skaffer User en eller flere valg med objekter av en gitt ObjectType. Objektene for hver valg er de elementene identifisert i Object Map i DI til å bli av den gitte Obejct Type.	objectTypes, requestMessages	Element[]
play(element, async)	Spiller spesifisert Item, Component eller Descriptor og venter valgfritt for bekreftelse før den returnerer til DIM.	element, async	PlayStatus

Tabell A.1: Tabell over benyttede DIBOer

A.1.1 getObjects

DIBOen `getObjects` anmoder User å velge DIDL-elementer av spesifisert Object Types fra den bestemte DID som inneholder eller refereres til DIM ut fra hvordan DIBO er kalt.

Parameteren `objectTypes` er en tabell med tekstverdier som spesifiserer Object Type for hvert element som User blir anmodet om å velge. Object

Type er en Object Type funnet i Object Map for den bestemte DID.

For hvert Object Type spesifisert i parameteren `objectTypes`, tillater DIBO User å velge ett element som passer til den Object Type.

Parameteren `requestMessages` er en tabell av string-verdier som inneholder en kravbeskjed for hver av dataverdiene indikert av deler av tabellen `dataType`. Parameteren `requestMessages` kan være null, i dette tilfellet ingen tydelige beskjeder er spesifisert. Hvis parameteren `requestMessages` ikke er null, er det en feil hvis antall medlemmer av parameteren `requestMessages` ikke er lik antallet av beskjeder i parameteren `dataTypes`. Uansett, en eller flere av medlemmene av parameteren `requestMessages` kan være null.

Valgte elementer av den spesifikke Object Type er tilgjengelig for anropte DIM via den returnerte tabellen av Element-objekter. Dette kan igjen benyttes som parameter for andre DIBO.

A.1.2 getObjectMap

Denne DIBO tillater gjenfinning av et Object Map fra en forekomst av et DID-dokument.

Parameteren `document` skal være et DOM Document-objekt som representerer forekomsten av DID-dokumentet med ObjectType-informasjonen som trengs for å konstruere Object Map. Object Map er representert i DIML av et objecMap-objekt.

Et eksempel kan være hvis en DIMforfatter ønsker å prosessere alle DID Objects av en bestemt Object Type. For eksempel, i et musikkalbum Digital Item, for å få alle Objects assosiert med et musikkspor Object Type, kan DIMforfatteren bruke getObjects-operasjon av ObjectMap-objekt. Tidligere en dette, ville DIMforfatteren trenge å kalle DIBOen getObjectMap for å først finne igjen ObjectMap.

A.1.3 play

Denne DIBO sørger for at DIDL-elementet representert av element-parameteren gjengis i en transient og direkte merkbar representasjon.

Måten å spille elementet på, tilpasset innholdet, er etterlatt som en implementeringsvalg av DIBO-implementereren.

For en asynkron utførelse, skal ikke denne DIBO returnere utførelseskontrollen til anropt DIM inntil spillestatusen går over til STATICPLAY eller RELEASED.

For tidsbasert media vil spillestatusen med suksessfull spilling av elementet bli TIMEPLAY. Når media og sluttid er nådd, vil statusen gå over til STATICPLAY.

For ikke-tidsbasert media vil spillestatusen på fullstendig spilt element bli STATICPLAY.

Hvis elementet ikke blir fullstendig spilt vil spillestatusen bli RELEASED.

DIBO returnerer et PlayStatus-objekt som frembringer en håndtering til spilleinstansen av elementet og dens status. Denne håndteringen er brukt som en parameter til andre DIBO for å kontrollere statusen av spilleinstansen av elementet.

Tillegg B

B.1 Tabell med forkortelser og foklaringer

Tabell B.1: Tabellen inneholder forkortelser og forklaringer

MPEG	Moving Pictures Experts Group	Internasjonalt standardorgan
DI	Digital Item	Strukturert digitalt objekt som inkluderer en standard representasjon, identifikasjon og metadata innenfor MPEG-21 sitt rammeverk
DID	Digital Item Declaration	Deklarering av ressurser, metadata og deres internsammenheng av en Digital Item
DIDL	Digital Item Declaration Language	XML-basert språk som inkluderer valideringsregler spesifisert av ISO/IEC 21000-2 for en standard representasjon i XML av en DID
DII	Digital Item Identification	Spesifiserer hvordan Digital Item skal identifiseres unikt.
DIA	Digital Item Adaptation	Inneholder verktøy for en kontrollerende tilpasning av Digital Item
DIAC	Digital Item Adaptation Configuration	Verktøy for å styre eller konfigurere hvem som skal gjøre hva, og hvordan, i et scenario hvor DIA benyttes
DIP	Digital Item Processing	Spesifiserer syntaksen og semantikken av verktøy som kan benyttes til å prosessere Digital Item
DIM	Digital Item Method	Verktøy for å uttrykke den foreslåtte interaksjonen av en User med et Digital Item hos nivået ved Digital Item Declaration
DOM	Document Object Model	Et grensesnitt for applikasjonsprogrammering
DIML	Digital Item Method Language	Språket som DIM er uttrykt i
DIBO	Digital Item Base Operation	Basisoperasjon som skaffer tilgang til funksjonalitet implementert av en Peer og brukt til å forfatte Digital Item Method
DIXO	Digital Item eXtension Operation	Operasjon som tillater utvidede funksjonaliteter til å bli anropt fra en Digital Item Method
DIXL	Digital Item eXtension Operation Language	Programmeringsspråk i hvordan Digital Item eXtension Operation er definert

CDI	Content Digital Item	Inneholder ressusrser slik som MP3-filer
XDI	Context Digital Item	Inneholder informasjon om hvordan konteksten i CDI skal benyttes
W3C	World Wide Web Consortium	
XML	Extensible Markup Language	Språk definert av World Wide Web Consortium
HTML	Hypertext Markup Language	Formateringsspråk
SGML	Standard Generalized Markup Language	Generelt formateringsspråk, med enorme kapasiteter
3GPP	3rd Generation Partnership Project	Samarbeidsprosjekt
NMT	Nordisk Mobil Telefon	1. generasjons mobilnett
GSM	Global System for Mobile Communications	2. generasjons mobilnett
UMTS	Universal Mobile Telecommunications System	3. generasjons mobilnett

Tillegg C

C.1 Resultat fra spørreundersøkelse

Januar 2006 ble det gjort en spørreundersøkelse blant 10 studenter fra NTNU. Undersøkelsen gikk ut på finne ut hva studentene ønsket av en internettjeneste som Filmweb på mobiltelefonen. En forutsetning for å kunne svare på spørreundersøkelsen var at studentene hadde kjennskap til Filmweb som internettjeneste. Spørreundersøkelsen er vedlagt i CD-vedlegg E.1.

Studentene måtte rangere på en skal fra 1-5 hva som var viktigst for de å ha i en internettjeneste og på mobilen. Studentene kunne velge mellom tjenestene kinoprogram, filmomtale, topp10-liste for kino, bestilling av kinobilletter og filmtrailer. Konklusjonen fra dette var at ved en internettjeneste var det viktigst å se kinoprogrammet, mens det minst viktige var å se topp10-liste for kino, se tabell C.1. I en mobiltjeneste var det viktigst å kunne bestille billetter, mens det minst viktige var å se filmtrailer, se tabell C.2.

I tillegg måtte studentene svare på hvilken tjeneste de kunne tenkt seg som startside på en mobiltjeneste. Da var resultatet likt bortsett fra én student, kinoprogrammet var ønskelig startside.

Til slutt kunne studentene føye til eventuelle andre kommentarer. Disse kommentarene hadde studentene:

- Oversiktlige menyer
- Som 3G-tjeneste må siden være oversiktlig og rask og laste, det vil si lite tung grafikk. Ved trailere kan en tillate seg litt ekstra båndbredde.
- Enkel å laste ved bruk av 3G

Tabell C.1: Viktig ved Filmweb.no

Valg												Konklusjon
Kinoprogram	1	1	1	2	1	2	1	2	1	2		1
Filmomtale	3	3	3	1	3	4	4	3	3	1		3
Topp10-liste for kino	5	5	5	5	5	3	2	4	5	3		5
Bestilling av kinobilletter	2	2	2	4	2	1	3	1	2	4		2
Filmtrailer	4	4	4	3	4	5	5	5	4	5		4

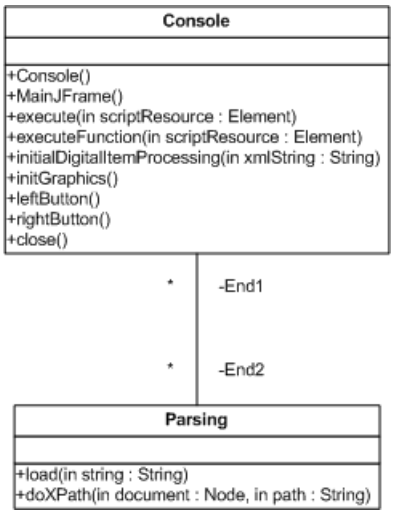
Tabell C.2: Viktig ved 3G-Filmweb

Valg												Konklusjon
Kinoprogram	1	1	1	2	2	2	2	2	2	2		2
Filmomtale	3	3	3	1	4	4	4	3	3	3		3
Topp10-liste for kino	4	4	4	5	5	3	3	4	4	1		4
Bestilling av kinobilletter	2	2	2	4	1	1	1	1	1	3		1
Filmtrailer	5	5	5	3	3	5	5	5	5	5		5

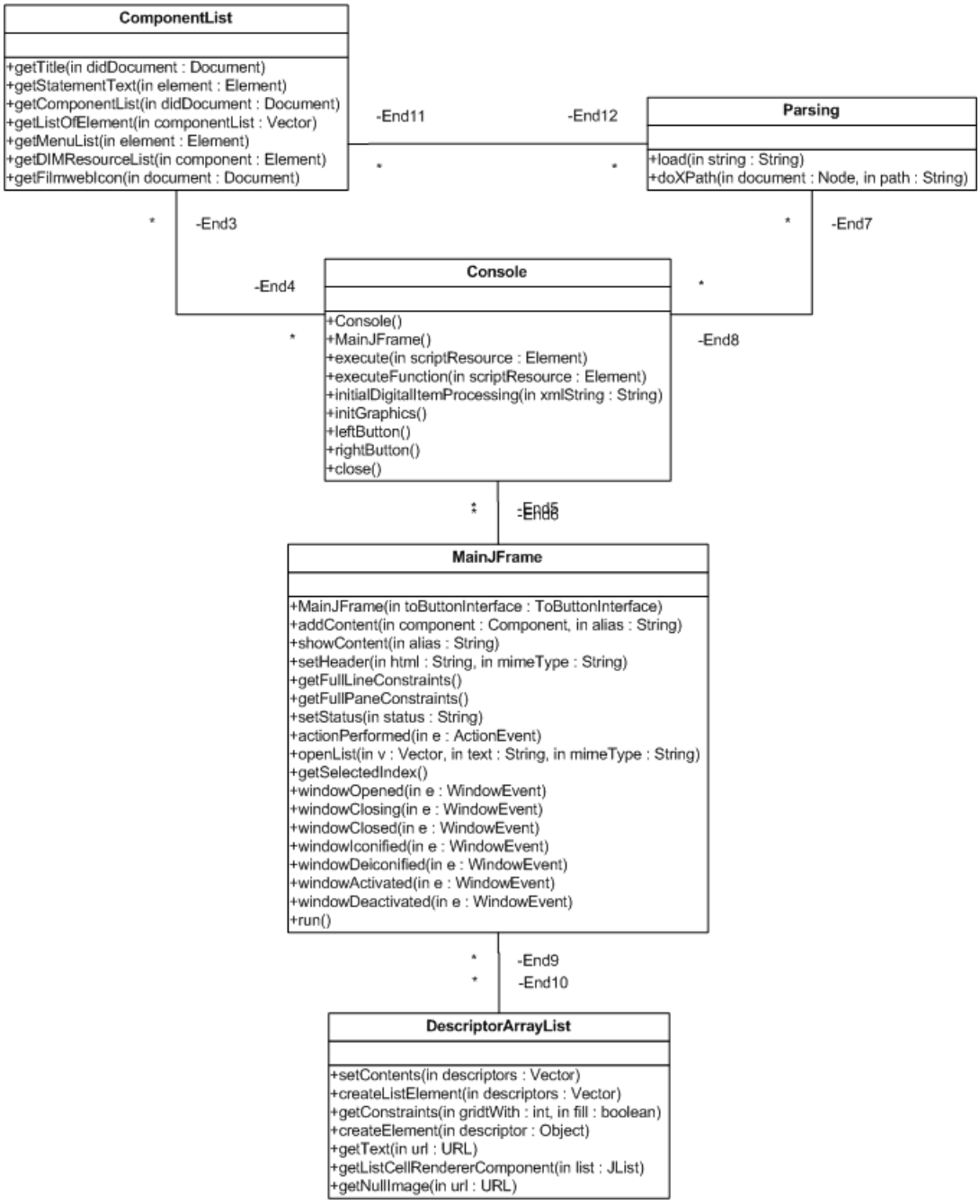
Tillegg D

D.1 UML-diagram

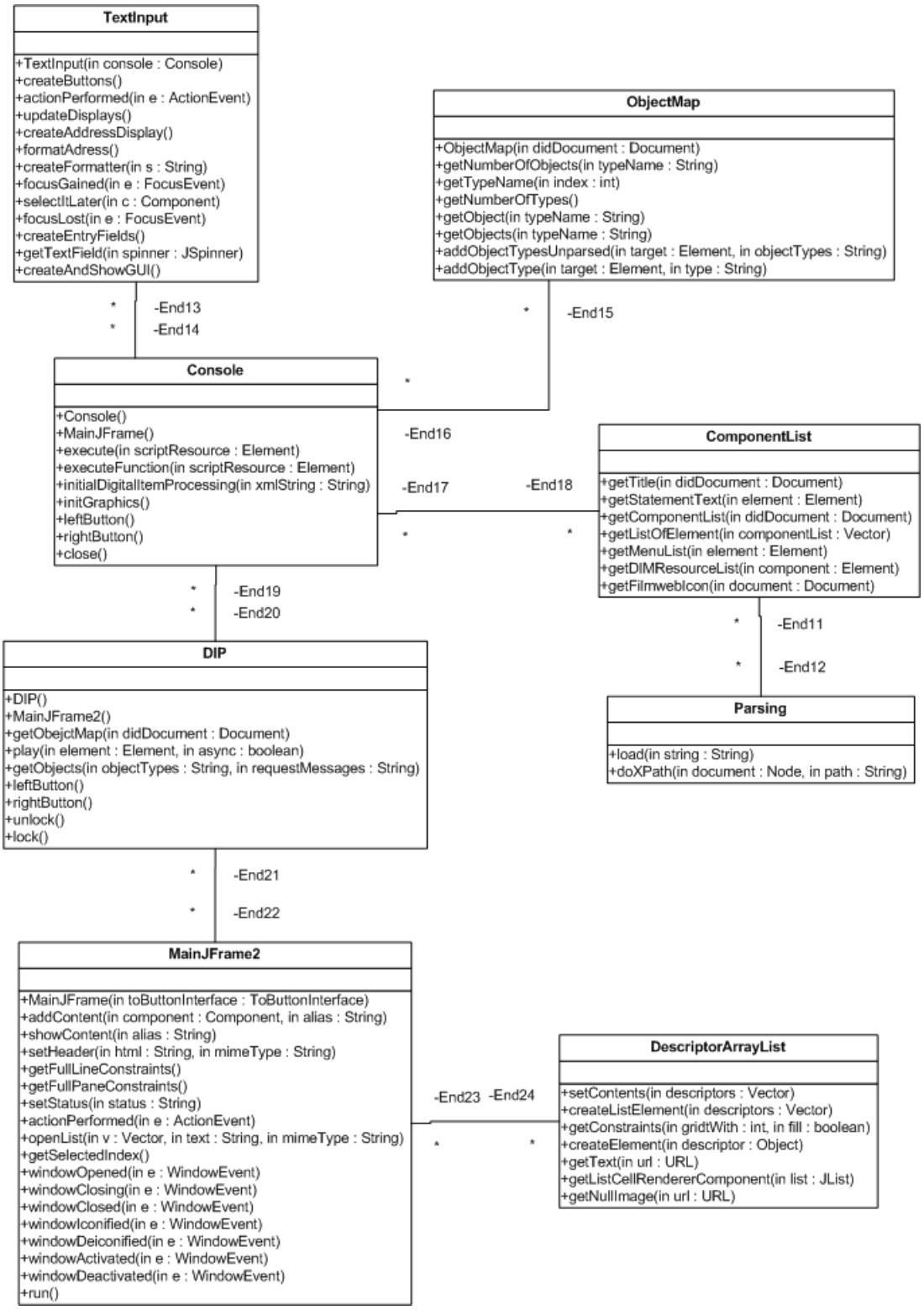
UML-diagram for Javakode:



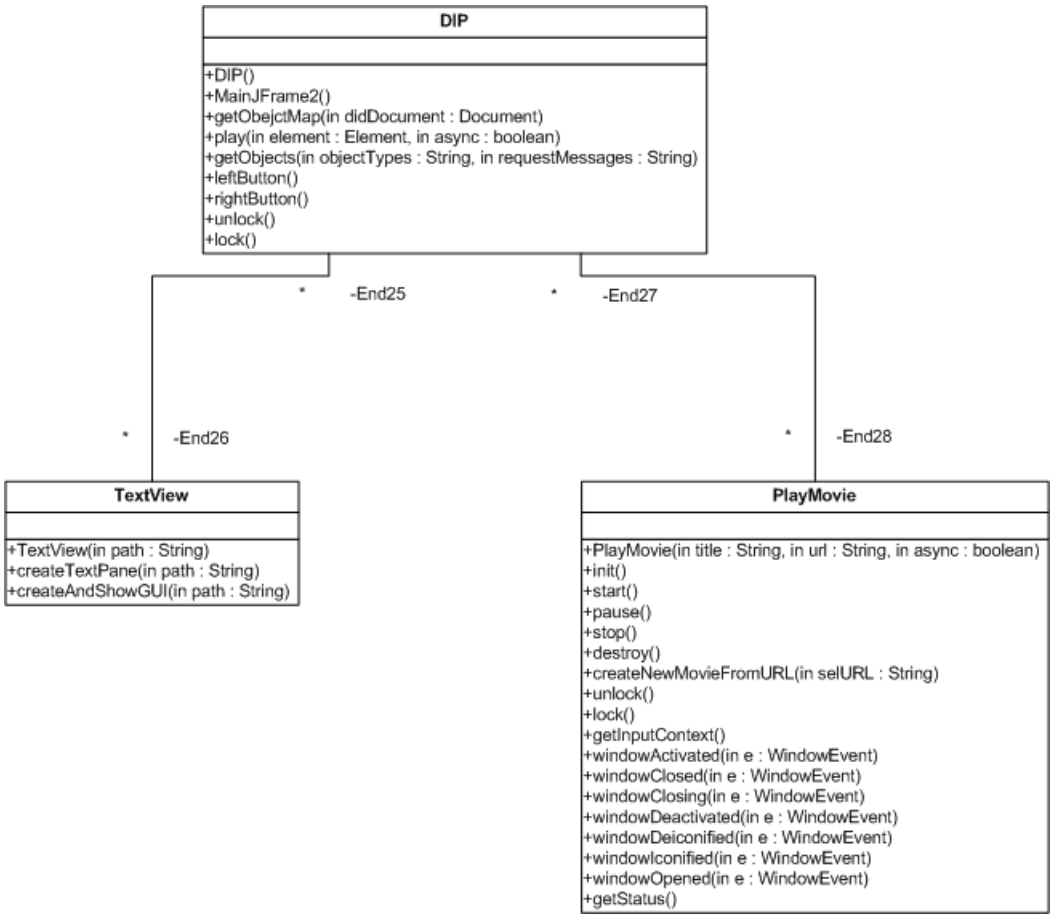
Figur D.1: Console vs Parsing



Figur D.2: GUI for DIM



FIGUR D.3: GUI for filmvalg



Figur D.4: GUI for valgt film for valgt DIM

Tillegg E

E.1 CD-rom

Til slutt i diplomrapporten er det vedlagt et CD-ROM med følgende vedlegg:

- ECMAScriptmotor - prosjektfiler som treng for ECMASriptmotor
- Flytskjema - flytskjema for hovedhendelser og UML-diagram
- XML - XML-dokument for diplomoppgaven
- Java - bibliotek-, prosjekt- og klassefiler
- Rapport - rapport vedlagt som pdf-fil
- Spørreundersøkelse - skjema brukt til spørreundersøkelse for studenter ved NTNU