Arkitektur for digitale bibliotek

Trond Aalberg og Knut Hegna

BIBSYS

Postadresse: BIBSYS, 7491 Trondheim
Telefon: 73 59 70 67
Telefaks: 73 59 68 48
Epost: bibdrift@bibsys.no
www.bibsys.no

2000

ISBN 82-7729-027-6

Innhold

Forord **7**

1 Digitale bibliotek **11**
1.1 Hva er digitale bibliotek? . . . 11
1.2 Sentrale aspekter ved digitale bibliotek . . . 13
1.3 Digitale læremidler . . . 15
1.4 BIBSYS som digitalt bibliotek . . . 17
1.5 Er World Wide Web et digitalt bibliotek? . . . 20
1.6 Arkitektur for digitale bibliotek . . . 21

2 Informasjonsobjekter **25**
2.1 Informasjon og data . . . 25
2.2 Informasjonsobjekt . . . 26
2.3 Dokumenter . . . 30
2.3.1 Hva er et dokument? . . . 30
2.3.2 Digitale dokumenter . . . 32
2.3.3 Nettdokumenter . . . 32
2.3.4 Filer og dokumenter . . . 33
2.4 Samlinger . . . 34
2.4.1 Samlingsbygging . . . 35
2.4.2 Samling som intellektuell ressurs . . . 35
2.4.3 Samlinger etter bibliografisk profil . . . 36
2.4.4 Integrerte samlinger . . . 36

3 Metadata **39**
3.1 Bakgrunn . . . 39
3.2 Definisjon og avgrensning . . . 41
3.3 Formål . . . 42
3.3.1 Metadata som støtter gjenfinning . . . 43
3.3.2 Metadata om tilgjengelighet . . . 44
3.3.3 Metadata om sanseliggjøring . . . 44
3.3.4 Metadata om identifikasjon/verifikasjon/autentisering . 45
3.3.5 Metadata som kontekst . . . 45

3.3.6 Metadata som støtter bevaring . . . 45
3.3.7 Metadata som administrativt underlag . . . 47
3.4 Overordnete kjennetegn på metadata . . . 47
3.4.1 Stabile og dynamiske metadata . . . 47
3.4.2 Autoriserte metadata . . . 47
3.4.3 Organisering av metadata . . . 47
3.4.4 Samvirke mellom metadataformater . . . 48
3.5 Metadata : formateksempler . . . 49
3.5.1 Formater med spesiell innretning . . . 49
3.5.2 Generelle formater . . . 52
3.6 Oppsummering . . . 56
3.6.1 Biblioteksektorens spesifikke behov . . . 56
3.6.2 Supplering med andre formater . . . 56
3.6.3 Samvirke . . . 57
3.7 FRBR-modellen . . . 57
3.7.1 Målet med bibliografiske beskrivelser . . . 58
3.7.2 Modellens byggeklosser . . . 59
3.7.3 Modellen . . . 66

4 Digital informasjon 71
4.1 CNRI-arkitekturen . . . 71
4.1.1 Fokus for arkitekturen . . . 72
4.1.2 Arkitekturens elementer . . . 72
4.2 Hypertekst . . . 74
4.2.1 Om hypertekst . . . 74
4.2.2 Dexter-modellen . . . 75
4.2.3 HTML – Hypertext Markup Language . . . 77
4.2.4 HyTime . . . 79
4.3 Markup-språk . . . 79
4.3.1 XML – Extensible Markup Language . . . 80
4.3.2 XML-dokumenter . . . 81
4.3.3 XML-baserte løsninger . . . 82
4.3.4 Stilsett . . . 84
4.4 Medietyper . . . 85
4.5 Tegnsett . . . 87
4.5.1 ASCII . . . 87
4.5.2 Andre tegnsett . . . 88
4.5.3 ISO/IEC 8859 . . . 88
4.5.4 UNICODE og ISO/IEC 10646 . . . 89

5 Identifikatorer 91
5.1 Identifikatorer i digitale bibliotek . . . 91
5.2 Terminologi . . . 92
5.3 Formålet . . . 94

5.4 Problemområder 95
5.5 Aspekter ved identifikatorsystemer 96
5.5.1 Identifikatoren 96
5.5.2 Administrering 98
5.5.3 Bruksområde 100
5.5.4 Identifikatortjenester 100
5.6 Identifikatorsystemer – eksempler 102
5.6.1 ISO-standarder 102
5.6.2 SICI - Serial Item and Contribution Identifier 106
5.6.3 BICI - Book Items and Contributions Identifier 108
5.6.4 Identifisering og adressering på Internett 109
5.6.5 PURL og virtuelle URL 116
5.6.6 The Handle System 118
5.6.7 DOI Digital Object Identifier 120
5.6.8 DNS (Domain Name Service) 123

6 Relasjoner og lenker 125
6.1 Relasjoner 125
6.2 Lenker 128
6.3 Bibliografiske relasjoner 132

7 Infrastruktur 135
7.1 Åpne systemer 135
7.2 Klient/tjener og protokoller 136
7.3 Objekter 139
7.4 Mellomvare 140
7.5 World Wide Web 142
7.5.1 Web-teknologi 142
7.5.2 HTTP 143
7.5.3 HTTP og interaktivitet 145
7.5.4 HTTP og sesjoner 146
7.5.5 Dynamiske web-sider 147
7.6 Informasjonsgjenfinning 151
7.6.1 Z39.50 151
7.6.2 HTTP-basert informasjonsgjenfinning 154
7.6.3 DASL 154
7.6.4 DLIOP 155
7.6.5 STARTS 155
7.6.6 Databasetilgang 157

8 Systemarkitektur 159
8.1 Ett-lags og to-lags arkitekturer 159
8.2 Tre-lags og multi-lags arkitekturer 160
8.3 Komponenter og grensesnitt 161

8.4 Komponentbasert arkitektur for digitale bibliotek . . . 163
8.5 Tjenester . . . 164
8.5.1 Søking . . . 165
8.5.2 Lagring og tilgang til informasjon . . . 165
8.5.3 Identifikatortjenester . . . 166
8.5.4 Metadatatjenester . . . 166
8.5.5 Andre tjenester . . . 167
8.6 Et eksempel – Dienst . . . 168

A Typer av standarder **171**

B Organisasjoner **173**
B.1 ANSI . . . 173
B.2 CENL . . . 173
B.3 CNRI . . . 173
B.4 ECMA . . . 174
B.5 IFLA . . . 174
B.6 Internett-organisasjoner . . . 175
B.7 The International DOI foundation . . . 176
B.8 ISO . . . 176
B.9 LC . . . 177
B.10 NISO . . . 177
B.11 The Unicode Consortium . . . 177
B.12 The World Wide Web Consortium . . . 178

C Metadata - formateksempler **179**
C.1 MARC . . . 179
C.2 VRA core . . . 182
C.3 Dublin Core . . . 183
C.4 NKKM . . . 184
C.5 TEI-header . . . 185
C.6 IAFA template . . . 188

D FRBR i praksis **189**
D.1 Entiteter og relasjoner i bibliografiske poster . . . 189
D.1.1 Et teatermanus . . . 189
D.1.2 Kronologien og strukturens utvikling . . . 194
D.1.3 Heart of darkness . . . 194
D.1.4 På skjermen . . . 197

Bibliografi **201**

Forord

Innledning

Denne rapporten er et av delprosjektene innen BDB (BIBSYS Digitalt Bibliotek). Prosjektet gjennomføres med særbevilgning fra Arbeidsgruppen for digitale læremidler [9, 8].

Oppdraget er gitt av BIBSYS som har sin forankring i de bibliotek som bruker BIBSYS-systemet i sine rutiner. Dette forholdet reflekteres i rapporten, og vår behandling av digitale bibliotek er på mange områder preget av å ha det tradisjonelle bibliotek som referanseramme.

Digitale bibliotek dreier seg om noe mer enn "digitalisering" av tradisjonelle bibliotek. Elektronisk publisering, museumssystemer, arkivsystemer og institusjonshukommelse (corporate memory), er eksempel på områder som det ville være naturlig å diskutere i en rapport som behandlet digitale bibliotek i sin fulle bredde. Sett fra denne synsvinkelen har vi valgt en snevrere innretning, men behandlingen har en innretning som er bredere enn *digitale læremidler*. Komponentene vi behandler er helt sentrale i begge tilfeller. I kapittel 1 legger vi rammen for hvordan vi i denne rapporten oppfatter uttrykket *digitale bibliotek*.

Sammen med de andre delprosjektene innen BDB vil rapporten være en viktig byggekloss. I mange tilfeller har vi nøyd oss med å peke på problemområder og tilgjengelig teknologi, heller enn å presentere konkrete løsninger. Anbefalinger og forslag til videre arbeid vil komme i et eget notat.

Arbeidet med rapporten er utført i perioden juli 1999 – juni 2000. Prosjektleder har vært Ole Husby, BIBSYS.

Litteraturstudier har vært en viktig del av prosjektet, noe som reflekteres i bibliografien til slutt i rapporten. Litteraturstudiene har gitt grunnlag for å omtale en rekke modeller og løsninger som er relevante i sammenheng med implementasjon av digitale bibliotek. Det er mange miljøer rundt i verden som arbeider både teoretisk og praktisk med digitale bibliotek, og noe av dette arbeidet er reflektert i rapporten.

Forfatterne har sin daglige arbeidsplass i hver sin by. Ut over månedlige prosjektmøter er samarbeidet gjennomført med CVS (Concurrent Versions System) som verktøy. Dette har gjort det mulig å arbeide på de samme filene fra hver vår arbeidsplass. Rapporten er skrevet med typografisystemet LaTeX

og med programpakken BibTEX for å håndtere referanser og litteraturliste.

Vårt håp er at rapporten vil være både en kilde til opplysning og et grunnlag for den videre diskusjonen om og utviklingen av digitale bibliotek.

Disposisjon

Vi starter denne rapporten med å drøfte hvilket innhold vi gir betegnelsen *digitale bibliotek* og hva som legges i uttrykket *arkitektur for digitale bibliotek*.

I kapittel 2 presenterer vi *informasjonsobjekt* som en overordnet størrelse for de enheter som digitale bibliotek skal forvalte og gi tilgang til, og vi gir en gjennomgang av begrepene *dokument* og *samlinger*. Disse begrepene gjelder både tradisjonelle og digitale bibliotek, men framveksten av elektronisk tilgjengelig informasjon over nettverk tilsier en ny gjennomgang.

Metadata og *metadatastruktur* behandles i kapittel 3. Spesiell vekt blir lagt på presentasjonen av IFLAs forslag til struktur for bibliografisk informasjon (FRBR).

Viktige elementer i en arkitektur for digitale bibliotek er hvordan informasjonsobjekter representeres digitalt, identifikasjon av informasjonsobjektene, og relasjoner mellom informasjonsobjekter. Dette behandles i henholdsvis kapittel 4, 5 og 6.

I kapittel 7 behandler vi forskjellige sider av den infrastrukturen som er fundamentet for digitale bibliotek som et samvirke av distribuerte tjenester, og i kapittel 8 diskuterer vi en overordnet systemarkitektur for digitale bibliotek.

Rapporten har flere tillegg. Tillegg A gir en generell innføring i forskjellige typer av standarder, og tillegg B er en beskrivelse av de mest sentrale organisasjonene som er omtalt i denne rapporten. Tillegg C inneholder eksempler på forskjellige metadataformater, og tillegg D gjennomgår et konkret eksempel på hvordan FRBR-modellen kan virke i praksis.

Figurene

I denne rapporten er det brukt mange figurer. Noen av disse er kun ment for å illustrere eller eksemplifisere det som beskrives i teksten, andre er mer formelle modeller som brukes for å spesifisere egenskaper eller forhold som vanskelig kan beskrives eller forklares ved hjelp av tekst.

I denne rapporten har vi et pragmatisk forhold til modelleringsspråk. Flere av modellene er utviklet av andre og gjengis i sin opprinnelige form. Noen av disse modellene er basert på en heller uformell grafisk notasjon og er mest egnet som illustrasjon, andre benytter en formell grafisk notasjon hvor de forskjellige piler og bokser har en spesifikk betydning.

Vi har valgt å spesifisere egne modeller ved hjelp av Unified Modelling Language (UML) [18, 142]. Dette er et standardisert grafisk språk for å *spesifisere, konstruere, visualisere og dokumentere programvare-intensive systemer*.

Dette modelleringsspråket er brukt for de mer formelle modellene, men der visualisering er det primære har vi også valgt mer uformelle diagrammer og illustrasjoner.

Forfatterne

Trond Aalberg er cand.scient. med hovedfag i informasjonsforvaltning fra Institutt for datateknikk og informasjonsvitenskap, NTNU. Han er for tiden stipendiat ved samme institutt, hvor han tar en doktorgrad i feltet digitale bibliotek.

Knut Hegna er utdannet cand.real. med hovedfag i informatikk, og har siden 1979 arbeidet med edb-systemer for bibliotek, dels nasjonalt rettete tjenester (samkataloger og nasjonalbibliografier) og dels eksperimentelle edb-tjenester for lokale forhold i Informatikkbiblioteket ved Universitetet i Oslo, der han nå arbeider som førstebibliotekar.

Kapittel 1

Digitale bibliotek

1.1 Hva er digitale bibliotek?

I de senere årene har det vært en bred satsing på digitale bibliotek. Vi har i dag en teknologisk infrastruktur som gjør det mulig raskt og kostnadseffektivt å formidle informasjon over nett til et globalt publikum. Det er mange faktorer som har bidratt til dette, ikke minst World Wide Web som er blitt en felles arena for formidling av informasjon. Denne utviklingen er nå kommet i en fase hvor det er fokus på kvalitet både ved den informasjonen som gjøres tilgjengelig og de tjenestene vi bruker for å finne fram til informasjonen. På samme måte som tradisjonelle bibliotek har fungert som et verdiskapende mellomledd mellom brukere og produsenter av informasjon, vil digitale bibliotek ivareta denne rollen i en verden av digital informasjon tilgjengelig over nett.

Det er vanskelig å definere hvor grensene for det digitale bibliotek går. Tradisjonelt har bibliotekbegrepet vært knyttet til samlinger av dokumenter, og digitale bibliotek oppfattes derfor av mange som samlinger av digitale dokumenter. Dette er et snevert syn på digitale bibliotek som i liten grad innbefatter det som er i ferd med å skje. Informasjonsteknologien, især bruken av nettverk, gir oss stadig bedre muligheter til samle og organisere informasjon som eies, administreres og forvaltes av andre, og gi brukerne direkte aksess til slike "virtuelle samlinger". Om dette er noe nytt eller bare en videreføring av tjenester som lenge har eksistert i tradisjonelle bibliotek, for eksempel tidsskriftindekser, online-kataloger og fjernlån, kan diskuteres, men teknologien gir oss mulighet til vesentlig bedre integrering og transparent aksess til informasjon fra et bredt spekter av kilder. Et annet viktig trekk ved digitale bibliotek er muligheten for å integrere. Når det digitale formatet og nettverket er formidlingskanalen, virker teknologien som en katalysator som fjerner etablerte skiller mellom organisasjoner og tradisjoner. Det er derfor riktig å se på digitale bibliotek som en utvidelse av tradisjonell informasjonsforvaltning og ikke som avgrensede enkeltsamlinger av digitale dokumenter.

Generelt kan vi si at digitale bibliotek har oppstått som et eget område

fordi det er stor interesse i og behov for informasjonsteknologi og nettverk for å organisere og gjøre tilgjengelig informasjon og kunnskap. Selve begrepet *digitale bibliotek* er det derfor vanskelig å gi en presis definisjon av, og digitale bibliotek vil også i fremtiden være et diffust begrep som forskjellige aktører legger egne interesser i. Dette er ikke noe særegent for digitale bibliotek, også for tradisjonelle bibliotek er det stor variasjon i hva de inneholder, hvilke tjenester de tilbyr, og hvilke brukergrupper og brukerbehov de er rettet mot.

Carl Lagoze og Sandra Payette gir følgende definisjon av hva digitale bibliotek er, med vekt på at et digitalt bibliotek er *en samling digitale objekter som forvaltes* [112]:

> ... we propose the following working definition of a digital library. A digital library is a managed collection of digital objects (content) and services (functionality) associated with the storage, discovery, retrieval, and preservation of those objects. Management begins with selection of the digital objects in the collection. Objects are selected from a global information space (e.g. the set of all published books, or the set of all objects on the Internet), and become constituents of the library collections based on criteria applied by collection managers (which may be human or automated). Management also entails the definition of the services included in the digital library. Some common examples of services are indexing, which allows discovery of objects in the collection; preservation, which assures the longevity of the objects in the collection; and awareness, which alerts users to changes in the collection.

I introduksjonen til første utgave av Springer forlags "International Journal on Digital Libraries" finner vi følgende beskrivelse av digitale bibliotek [1]:

> Digital Libraries are concerned with the creation and management of information sources, the movement of information across global networks and the effective use of this information by a wide range of users.....

Teknologien gir oss stadig nye muligheter, og digitale bibliotek er et område som utvikles i samspill med den teknologiske utviklingen. I stedet for å lage avgrensende definisjoner på digitale bibliotek, bør vi heller være opptatt av å forme fremtidens verktøy for organisering av informasjon og interaksjon med informasjon [26]:

> Any attempt to define what a digital library is or is not will either be too vague or too restrictive. The best we can do is to take a close look at emerging technology and attempt to understand the impact it will have on how we will organize and interact with repositories of knowledge in the future, be they digital or otherwise.

Avhengig av vårt ståsted eller forhold til digitale bibliotek vil vi ha forskjellig oppfatning av hva som er det sentrale ved digitale bibliotek. Peter Murray beskriver digitale bibliotek som et puslespill som oppfattes forskjellig i forhold til det perspektiv man har [134]:

- Brukerne vil oppfatte det digitale bibliotek som integrert aksess til distribuerte informasjonsressurser og tjenester.
- Bibliotekene vil se det digitale bibliotek som integrert forvaltning og organisering av informasjon.
- Fra en systemteknisk vinkel er det digitale bibliotek en integrering av standarder for å utveksle og kommunisere om informasjon.

1.2 Sentrale aspekter ved digitale bibliotek

Digitale bibliotek er et satsingsområde som ikke er basert på en enkelt type informasjonsressurs, en enkelt teknologi, type informasjonssystem eller en enkelt brukergruppe. Digitale bibliotek er mer et samlebegrep og en møteplass for en rekke brukerbehov, informasjonsressurser og teknologier. Vi finner likevel noen trekk ved digitale bibliotek som er karakteristiske:

- **Informasjonssentrert**
  Digitale bibliotek er informasjonssentrerte. Det som skal forvaltes, organiseres og gjøres tilgjengelig, er enheter av informasjon. Ofte karakteriseres disse som dokumenter eller dokumentlignende objekter (forkortes: DLO), men digitale bibliotek kan også inneholde mange andre objekter som kan være kilde til informasjon, f.eks. interaktive informasjonskilder som programvare. I denne rapporten benytter vi uttrykket *informasjonsobjekter* som en fellesbetegnelse på de enhetene digitale bibliotek inneholder, eller viser vei til.

  Selv om informasjonsobjektene er det primære i digitale bibliotek, vil digitale bibliotek også være basert på mange andre former for informasjon, enten dette er informasjon om dokumentene, informasjon om brukere, informasjon om sammenhengen mellom dokumenter, og annen informasjon som er relevant i kontekst av bruk og forvaltning av informasjonsobjekter.

- **Digitale informasjonsobjekter**
  Den informasjon som skal fylle et digitalt bibliotek, må enten finnes i digital form eller være tilstede i det digitale bibliotek via digitale representasjoner, enten som metadata eller kanskje bare som en identifikator.

  Å avgrense digitale bibliotek til kun å gjelde digitale informasjonsobjekter, slik enkelte gjør, kan fort bli snevert. Mye informasjon vil også i fremtiden kun finnes som fysiske informasjonsobjekter, for eksempel

papirbasert. Slike informasjonsobjekter må likevel kunne integreres i det digitale bibliotek ved hjelp av digitale representasjoner. Vi vil derfor ikke få enten digitale eller tradisjonelle bibliotek, men også mange hybride bibliotek som kombinerer det digitale med det fysiske.

- **Samlingsorientert**
  I bibliotek bygger man opp kvalitetskontrollerte samlinger av dokumenter. Dette er også et fundament for digitale bibliotek, men i digitale bibliotek vi vil finne mange forskjellige former for samlingsvirksomhet. Vi kan ha digitale samlinger som gjøres tilgjengelig av én virksomhet, og vi kan ha "virtuelle samlinger" som er portaler inn til mange underliggende primære samlinger eller deler av samlinger[1].

  I et distribuert informasjonsrom som World Wide Web finnes også mye informasjon som er uten fast organisering. Seleksjon og indeksering av de verdifulle delene av slik informasjon, kan være en samlingsoppbygging som er relevant for digitale bibliotek.

- **Nettbasert informasjonsdeling**
  Bruk av nettverksteknologi og spesielt bruk av Internett, er sentralt i digitale bibliotek. Informasjon produseres og forvaltes av et bredt spekter organisasjoner, og med nettverksteknologi kan vi fjerne organisatoriske og geografiske grenser. Bruk av nettverk har også ført til nye mønster for produksjon, publisering og tilgang til informasjon, og vi må i dag forholde oss til en vesentlig mer distribuert virkelighet hvor nettverket er bindeleddet.

- **Integrerte løsninger**
  Som sluttbrukere har vi ofte liten interesse i å personlig organisere og forholde oss til en kompleks verden av distribuert informasjon og funksjonalitet. Selv om oppdagelseslysten fortsatt er stor for de fleste av oss når det gjelder Internett, har vi behov for effektive løsninger som er tilpasset våre behov og vårt ekspertisenivå. Digitale bibliotek er å integrere informasjon og funksjonalitet på en måte som effektiviserer og forbedrer brukernes tilgang til informasjon. Et essensielt virkemiddel for å oppnå dette på tvers av organisasjoner og geografiske grenser, er å bruke standarder og utvikle nye standarder der det er behov for dette.

- **Interaktivitet**
  Digital informasjon, teknologi og nettverk utgjør tilsammen et medium som kjennetegnes ved store muligheter for interaksjon. Brukerne er ikke lenger passive konsumenter av informasjon og tjenester, men møtet mellom bruker og informasjon er en dialog. I digitale bibliotek har brukerne

[1] Med samling menes her både et sett av faktiske informasjonsobjekter som en selv råder over. En virtuell samling vil være et sett av pekere (lenker) til informasjon hvor en ikke selv råder over et fysisk eller digitalt eksemplar

store muligheter for interaksjon med både tjenester og informasjonen (hypertekst).

1.3 Digitale læremidler

På samme måte som tradisjonelle bibliotek har hatt en sentral rolle i undervisningsinstitusjoner, vil digitale bibliotek være et hovedelement i pedagogisk virke i fremtiden. Med den informasjonsteknologiske utviklingen kom også de digitale læremidlene, og med Internett kom bruk av nettverk til samme formål. Digitale læremidler og nettbasert undervisning er i dag et aktuelt satsingsområde for mange utdanningsinstitusjoner.

Læremidler

Læremidler kan være flere ting[2]:

- **Et redaksjonelt produkt** utviklet med læring som det primære mål. Dette kan være publikasjoner utviklet av eller i samarbeid med lærere og forlag, men også forelesninger og forelesningsnotater, undervisningsvideoer, øvingsoppgaver og løsningsforslag. Slike læremidler vil ha en veldefinert "model reader", være basert på en bestemt institusjonell praksis og forankring, og være rettet mot bestemte fag med en bestemt målgruppe på et bestemt nivå. Læremidlet har en didaktisk intensjon "bygget inn" i læremidlet (sekvensiering, oppgaver, didaktisk kommentarapparat, etc.), og læremiddelfunksjonen er den primære funksjonen til produktet.

- **En læringsressurs** hvor læremidlet er noe som brukes med læring som formål. Dette synet på læremidler inkluderer punktet over, men omfatter også produkter hvor læring ikke er definert som hovedfunksjonen. Slike læremidler vil ikke være basert på samme institusjonelle praksis og forankring, og vil heller ikke være rettet mot bestemte fag med en bestemt målgruppe på et bestemt nivå. Læremidlet har ingen didaktisk intensjon "bygget inn" i læremidlet, og "model reader" kan være både udefinert eller en annen. Sett fra avsender kan læremiddelfunksjonen være sekundær eller noe avsender ikke er oppmerksom på – læremiddelaspektet er noe som defineres gjennom bruk. Aviser, lovtekster, oppslagsverk og lignende er eksempler på slike læremidler.

- **En læringsomgivelse** hvor læremidlet er funksjonalitet til rådighet i en læringssituasjon – informasjonsmedier som er tilgjengelige, sosiale interaksjonsmuligheter og andre tjenester som kan brukes til læring. Dette vil omfatte begge punktene over, men i tillegg også omfatte veiledning,

[2] bygger på Jon Lanestedts (USIT) innledning på BIBSYS ideseminar om digitale bibliotek i Trondheim 10. mai 1999.

> bibliotektjenester, evalueringer av øvinger, og mye annet. I dette synet på læremidler er det ikke noe skille mellom enkeltobjekter og de tjenestene som brukes til læring.

Digitale læremidler kan diskuteres etter samme inndeling. Som redaksjonelt produkt vil digitale læremidler kun skille seg fra læremidler på andre medier ved at de er digitale og potensielt kan formidles over nett. En digital lærebok er kun en digital variant av trykte bøker, og det samme gjelder for andre læremidler. Som læringsomgivelse er det digitale og bruken av nettverk likevel et nytt medium eller en ny arena med muligheten for å fremme læring. Spesielt gir denne teknologien mulighet for en mer interaktiv læringsomgivelse. Informasjonsteknologien generelt er et verktøy for produksjon og formidling av læremidlene og etablering og administrering av læringsomgivelsene.

Arbeidsgruppen for digitale læremidler har i sin rapport fra 1997 definert sin bruk av "digitale læremidler" slik [8]:

> Arbeidsgruppen har tolket "digitale læremidler" i vid forstand, som det totale læringsmiljøet der ulike aspekter ved informasjonsteknologi utnyttes som et verktøy for å fremme læring via produkter ("læremidler") og prosesser ("læringsformer"), og der biblioteket fungerer som en integrert medspiller i forhold til begge komponenter.

Det digitale bibliotek har en sentral rolle i forhold til digitale læremidler. Arbeidsgruppen for digitale læremidler påpeker behovet for å:

> Utvikle og gjøre tilgjengelig digitale læremidler, herunder utvikling av digitale bibliotektjenester for formålet, samt kartlegge og gjøre tilgjengelig digitale læremidler utviklet av andre.

Arbeidsgruppens "Program digitalt bibliotek for digitale læremidler" definerer oppgavene til det digitale bibliotek slik [9]:

> Med utgangspunkt i denne tolkningen av nøkkelbegrepet "digitale læremidler" i arbeidsgruppens mandat, vil det digitale bibliotek måtte forstås som noe annet og mer enn en digital database med bibliotekets tradisjonelle oppgaver. Det vil være nødvendig at det "digitale bibliotek" i tillegg til å systematisere kunnskap om digitale læremidler brukes i et tett samarbeid med ulike fagmiljø, bidrar med kunnskapsorganisering og designkompetanse, kvalitetssikrer og organiserer internettressurser, gjør studentarbeider tilgjengelige ved å publisere på forespørsel, etc. En vesentlig del av arbeidsgruppens oppgave vil være å samle erfaringer og kunnskap om det potensialet det "digitale bibliotek" har i forhold til den totale læringssituasjonen, og på grunnlag av denne innsikten å foreslå hvordan disse tjenestene bør organiseres og driftes.

Bruk av Internett i læring er et område med stort potensial. Ved hjelp av nettbasert undervisning kan læring foregå uavhengig av fysisk oppmøte på læringsinstitusjonen. I sin enkleste form er dette bare en videreføring og forbedring av tidligere tiders fjernundervisning og brevkurs, hvor kommunikasjon i stedet foregår ved hjelp av epost. I de mer avanserte variantene av nettbasert undervisning og bruk digitale læremidler (som ressurs) kan dette gi en komplett omgivelse for læring (hvis vi ser bort fra behovet for menneskelig kontakt). Nettbasert undervisning er mer eller mindre synonymt med synet på læremidler som en læringsomgivelse fra punkt 3 over. Et skille går likevel mellom læring hvor nettverksbasert informasjon og tjenester er et læremiddel, og læring hvor f.eks. Internett er hovedarena både som læringsomgivelse og for administrasjon og organisering av læringen (f.eks. nettuniversitet).

1.4 BIBSYS som digitalt bibliotek

BIBSYS har en relativt lang historie som felles informasjonssystem (bibliotekautomatiseringssystem) for norske fag og forskningsbibliotek. Kjernen i BIBSYS er en felleskatalog – en database – hvor metadata om bibliotekenes fysiske dokumenter registreres. Alle typer materiale kan registreres i basen med MARC-formatet (bøker, tidsskrift, musikktrykk, bilder, kart osv.), uavhengig av om disse er digitale eller ikke. I dag består denne databasen av 2.8 millioner poster som beskriver 7.5 millioner fysiske dokumenter.

På grunn av den sentrale rollen BIBSYS har som felles informasjonssystem for de deltagende bibliotek, har det vært en naturlig utvikling mot BIBSYS som informasjonssenter også for en rekke andre ressurser som skal forvaltes og kanaliseres ut til bibliotekenes sluttbrukere. BIBSYS er ikke et *bibliotek*, men vi kan likevel karakterisere BIBSYS som et *digitalt bibliotek*, og det er mange elementer som viser at BIBSYS allerede er kommet langt på veien mot å bli et digitalt bibliotek:

- **Bruk av web**
  Bruk av nettverk har helt siden starten av vært et av BIBSYS fundamenter. I starten var dette basert på terminalemuleringer. Dette er videreført parallelt med at stadig nye kommunikasjonsmuligheter (epost-søk, gopher og web) har vært tidlig utprøvd og tatt i bruk. I dag er web det dominerende grensesnittet for sluttbrukerne.

- **Artikkeldatabase**
  BIBSYS har siden 1996 driftet en stor artikkeldatabase fra Institute for Scientific Information (ISI) bestående av 13 millioner artikler innen mange fagområder. En viktig tjeneste for ISI-basen er koblingen mot beholdningsinformasjonen i BIBSYS og mulighetene til enkelt å bestille kopier. BIBSYS har også siden januar 2000 driftet tilsvarende databaser fra Silverplatter.

- **FORSKDOK**
  BIBSYS har et eget system for å registrere forskningsdokumentasjon - FORSKDOK - hvor forskerne selv kan registrere både prosjekter (aktiviteter) og resultater (publikasjoner). Disse databasene er fremdeles små, men hele 40 institusjoner benytter i dag systemet. Neste versjon av systemet er under utvikling, hvor det blant annet fokuseres på lenking til de elektroniske dokumentene som beskrives i databasen.

- **Elektroniske tidsskrifter**
  Mange tidsskrifter kommer i elektronisk (digital) utgave[3]. BIBSYS tilbyr en database på over 10.000 tidsskrifter som er knyttet til institusjon og emneord. Brukerne kan her søke fram tidsskriftene, få oversikt over elektroniske tidsskrifter ved egen institusjon, og få presentert tidsskriftene elektronisk. Data i denne databasen er dannet på grunnlag av opplysninger fra leverandørene.

- **Digitale dokumenter**
  Det er i dag mulig å benytte BIBSYS-systemet for å registrere digitale dokumenter. Metadata lagres i BIBSYS-basen inklusiv en adresse til det digitale dokumentet (MARC-felt 856). Ved web-søk mot basene får brukerne direkte tilgang til dokumentene via en lenke i utlistingen.

- **Lager**
  I dag eksisterer det ikke noe endelig system for lagring av digitale dokumenter. BIBSYS har gående et prosjekt der verktøy for lagring av digitale dokumenter blir prøvd ut.

- **Gjenbruk av katalogiseringsdata**
  BIBSYS har i en årrekke vedlikeholdt en database (MARC-brønn) over poster importert fra Library of Congress. Disse dataene kan gjenbrukes i katalogiseringsmodulen for å spare arbeid. I den seinere tid er det også laget et system for å importere Dublin Core-data direkte fra nettdokumenter som ønskes katalogisert i BIBSYS.

- **Z39.50**
  Det har i mange år pågått internasjonale prosjekter med målsettingen å standardisere tilgang til databaser. Søkeprotokollen Z39.50 er en ANSI/ISO standard som er utarbeidet til dette formålet. BIBSYS har Z39.50-grensesnitt (Z39.50-tjener) både mot felleskatalogen, forskdok og NWI[4], og i GENSØK er det integrert en Z39.50-klient for å støtte gjenbruk av katalogposter fra andre biblioteksystem.

[3] Tidsskrifter som er tilgjengelig på Internett kalles ofte for *elektroniske* tidsskrifter, men vi kunne like gjerne kalt disse for *digitale* tidsskrifter

[4] NWI - Nordic Web Index - et nordisk initiativ for å lage en database over nordiske ressurser.

- **ZSøk**
  BIBSYS har utviklet et felles og integrert grensesnitt mot mange databaser, i hovedsak bibliografiske baser, men også fulltekstdatabaser. Denne tjenesten har fått navnet Zsøk, og er basert på verktøyet SiteSearch fra OCLC og bruk av Z39.50. Tjenesten gir sluttbrukerne mulighet til å søke i mange databaser samtidig gjennom ett og samme brukergrensesnitt, og tjenesten er integrert med felleskatalogen i BIBSYS og med en kopibestillingsfunksjon.

- **Emneportal**
  Internett er en viktig kilde til informasjon. Dagens mange søkemotorer gir ikke noen tilfredsstillende inngang til denne informasjonen. De er for lite presise, gir for mange treff, og tilfredsstiller ikke kravene fra fagmiljøene til kvalitet på informasjonen. Mange bibliotek har i dag utarbeidet egne oversikter over web-sider (portaler) som de tilbyr sine brukere. Her blir et kvalitetssikret utvalg ressurser katalogisert og registrert. BIBSYS vil i samarbeid med bibliotekene bygge opp en felles tjeneste til dette formålet. BIBSYS vil foreta den tekniske utviklingen og driften av systemet, og påtar seg det redaksjonelle ansvar for basen. Bibliotekene står for katalogiseringen og registreringen av ressursene. Fordelene med denne basen vil være større driftsikkerhet, enhetlig klassifisering og registrering, høy kvalitet på ressursene, og løpende kontroll av systemet (lenker, kvalitet osv).

- **BIBSYS LINK**
  BIBSYS har utviklet en standard for lenking mot beholdningsinformasjon i BIBSYS og til kopibestilling. Dette gir brukerne mulighet til effektivt å kunne bestille lån eller kopier fra et BIBSYS-bibliotek. Flere systemer har implementert denne standarden (bl.a. OVID, SilverPlatter og OCLC).

- **Docutrans**
  BIBSYS og Universitetsbiblioteket i Trondheim har et system for automatisk å sende ut digitale kopier til sluttbrukerne. Dette gjøres ved å koble en scanner sammen med informasjonen i BIBSYS-databasen. Løsningen kan generelt skape opphavsrettslige problemer, men en regner med at dette vil være uproblematisk bl.a. for eldre materiale. Det er et ønske at BIBSYS skal drifte dette systemet for alle deltakerbibliotekene som ønsker det.

- **Adgangskontrollsystem**
  Brukerne vil etter hvert bli tilbudt mange ulike tjenester gjennom BIBSYS eller via fellesavtaler eller egne avtaler. Ikke alle institusjoner vil ha de samme avtalene og tilgang til de samme tjenestene. Heller ikke alle brukere ved en institusjon vil ha adgang til de samme tjenestene. De

ulike tjenestene må sperres med brukernummer/passord. Det er derfor behov for et felles adgangskontrollsystem som bestemmer adgangen for den enkelte bruker, som administreres lokalt, og som tilordner ett felles brukernummer/passord til brukeren. Det må også være mulig å legge inn rutiner for betaling i systemet. BIBSYS ønsker å kunne tilby en slik tjeneste, og har vurdert noen systemer. Fordelen med et slikt felles system er det kan samordne og effektivisere det administrative arbeidet med brukernavn-/passord-oversikter og oversikter over IP-adresser. BIBSYS ser på adgangskontrollsystemer som en del av BDB-prosjektet.

- **BIBSYS Digitalt Bibliotek – BDB**
  BDB er et treårig prosjekt som ble startet i 1998. Hovedmålsettingen med BDB er å opprette eller gi tilgang til nye og eksisterende digitale informasjonsressurser, og med brukere innen det norske universitets- og høgskolesystemet som den primære brukergruppe. Dette er et prosjekt som innbefatter flere av prosjektene over, men hvor en også ser på mer fundamentale problemstillinger som arkitektur og metadata (bl.a. denne rapporten).

1.5 Er World Wide Web et digitalt bibliotek?

World Wide Web har drastisk endret vår adgang til informasjon fordi det gir enkel og brukervennlig tilgang til tjenester og dokumenter via nettverk. World Wide Web er også et attraktivt medium for formidling, noe som har resultert i mange nye kilder til informasjon, både fra seriøse og mindre seriøse aktører. Denne gjensidige stimuleringen av tilbud på og etterspørsel etter informasjon, sammen med en rekke andre faktorer som fritt tilgjengelige web-lesere og at flere og flere har tilgang til en datamaskin, har ført til en eksplosiv utvikling i bruk av World Wide Web og de ressurser som er tilgjengelig i dette informasjonsrommet.

På World Wide Web finner vi i dag en rekke tjenester som tilbyr søking i indekserte web-sider eller som organiserer web-sider i tematiske lister. Vi finner også en rekke web-steder hvor vi får direkte tilgang til bibliografiske databaser. Dette kombinert med lenkemekanismen i HTML som kan binde sammen ressurser, gjør at World Wide Web for sluttbrukere av informasjon kan fremstå som et eneste stort digitalt bibliotek, fordi de både kan lete etter informasjon og hente ut informasjon på en delvis organisert måte.

Det å karakterisere hele World Wide Web som et digitalt bibliotek er likevel ikke riktig. Tradisjonelle bibliotek er basert på kvalitetssikring, hvor samlingen er en ressurs som forvaltes, utvikles og kvalitetssikres på en planlagt måte. Den informasjonsforvaltning som finner sted når web-indekseringsroboter traverserer World Wide Web, er ofte basert på andre kriterier enn det som er grunnlaget for digitale bibliotek.

Clifford Lynch påpeker i en artikkel at World Wide Web i utgangspunktet ikke er designet for organisert publisering og gjenfinning av informasjon [121]. World Wide Web er ikke et digitalt bibliotek, men hvis det skal fortsette å vokse og bli brukt, er det behov for noe i likhet med tradisjonelle bibliotektjenester for å organisere, aksessere og bevare informasjonen på nettet. Selv med hjelp av dette vil ikke nettet ligne tradisjonelle bibliotek fordi innholdet er mer spredt, mangeartet og omfattende enn i tradisjonelle samlinger. Konsekvensen av dette er et behov for å kombinere det tradisjonelle bibliotekhåndverket som katalogisering, klassifisering og seleksjon med informasjonsteknologiens muligheter for automatisk indeksering og lagring.

Også Lagoze og Payette påpeker i sin definisjon av digitale bibliotek at World Wide Web ikke er et digitalt bibliotek [112]:

> ... By this definition, the World Wide Web, by itself, is NOT a digital library. It represents a set of objects jointed together technically (by the protocol HTTP), but not by any collection management or service management decisions.

Vi kan derfor konkludere med at World Wide Web som helhet ikke er et digitalt bibliotek, fordi informasjon og tjenester er for dårlig administrert og kvalitetssikret. Dette gjelder også for mange av de søketjenestene som finnes. Dette innebærer ikke at disse tjenestene er mindre verdifulle, bare at vi skal være forsiktig med å bruke bibliotekbegrepet på disse. Det vil likevel være mange tjenester og web-steder som er fullverdige digitale bibliotek, men da fordi de som enkeltsystemer tilfredstiller de krav vi assosierer med bibliotekbegrepet. Forholdet mellom digitale bibliotek og World Wide Web er mer preget av at disse er komplementære. Internett og World Wide Web er et viktig fundament for å realisere digitale bibliotek, og digitale bibliotek er viktig for World Wide Web fordi det gir muligheten for kvalitetssikrede og effektive tjenester for informasjon distribuert over nett.

1.6 Arkitektur for digitale bibliotek

I informasjonsteknologien brukes ofte *arkitektur* som en betegnelse på en konseptuell beskrivelse av et system. En arkitektur uttrykker hovedprinsippene i en løsning, og vi spesifiserer vanligvis arkitekturer med diagrammer som viser hovedtrekkene ved hjelp av bokser og piler.

I ordbøker og oppslagsverk finner vi arkitektur definert som "byggekunst", og brukt om forskjellige stilarter for byggverk. Kunnskapsforlagets fremmedordbok gir følgende definisjon av arkitektur:

> bygningskunst, bygningsstil; utforming.

I Encyclopædia Britannica finner vi en tilsvarende beskrivelse av arkitektur:

> the art and technique of designing and building, as distinguished from the skills associated with construction.

Går vi over til informasjonsteknologien, finner vi tilsvarende definisjoner av arkitektur, men her er det selvsagt snakk om "design og byggekunst" for programvare og maskinvare. En vanlig forståelse er at arkitektur er hovedstrukturen i et system, hvilke deler det er satt sammen av, og hvordan disse er relatert eller integrert:

> The structure of a system in terms of components and the interrelationships among its components. [155]

> An architecture must define the parts, the essential external charachteristics of each part, and the relationships between the parts. [46]

> A design. The term architecture can refer to either hardware or software, or to a combination of hardware and software. The architecture of a system always defines its broad outlines, and may define precise mechanisms as well. An open architecture allows the system to be connected easily to devices and programs made by other manufacturers. Open architectures use off-the-shelf components and conform to approved standards. A system with a closed architecture, on the other hand, is one whose design is proprietary, making it difficult to connect the system to other systems. (Webopædia[5])

I sitatene over ser vi at en arkitektur både kan være en fremstilling av hovedtrekkene i et system, og en detaljert beskrivelse av hvordan komponentene som inngår i systemet skal virke sammen.

For å beskrive en arkitektur for digitale bibliotek er det behov for modeller, fordi vi har bruk for en generell og konseptuell beskrivelse av arkitekturen – hvilke enheter som er del av systemet, og hvordan disse er relatert til hverandre. En modell er en forenkling, hvor de sentrale aspektene fremheves og de uvesentlige detaljene utelates. Vi lager modeller for bedre å forstå virkeligheten og det systemet vi utvikler, og vi lager modeller fordi både virkeligheten og det systemet vi utvikler, er for komplekse til å forståes som en enhet [18]. Formålet med modeller kan oppsummeres i følgende:

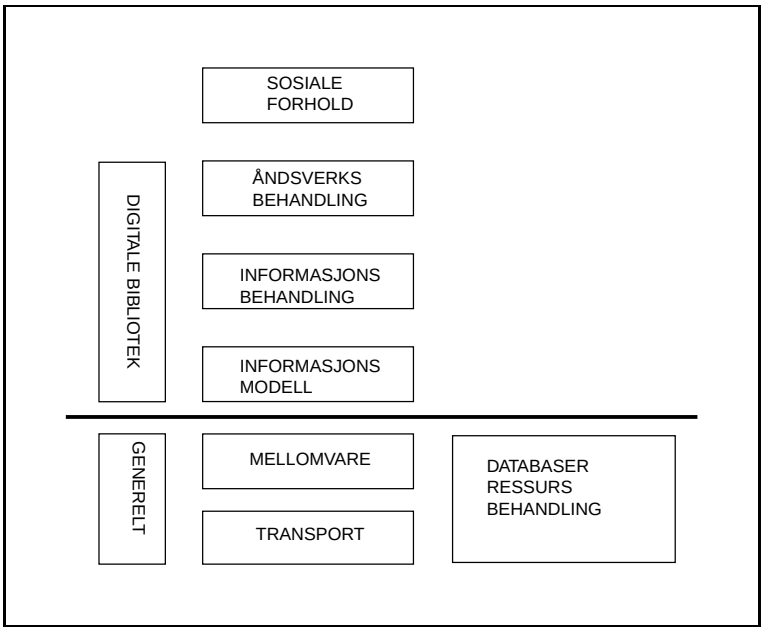
1. Modeller hjelper oss å visualisere et system, enten som det er eller slik vi vil ha det.
2. Modeller gjør at vi kan spesifisere strukturen og oppførselen i et system.
3. Modeller gir oss oppskriften på hvordan vi skal konstruere et system.

[5] HTTP://WWW.pcwebopaedia.com/TERM/a/architecture.HTML

4. Modeller dokumenterer de valg vi har gjort.

Den tradisjonelle bruken av arkitektur er vanligvis relatert til maskin- og programvare, men for digitale bibliotek er dette et snevert utgangspunkt. Et like viktig aspekt ved en arkitektur for digitale bibliotek er forståelsen av informasjonen. På samme måte som for de prosesserende enhetene, har vi behov for en arkitektur for informasjonsobjektene som utgjør innholdet i digitale bibliotek – en informasjonsmodell.

For digitale bibliotek hvor enheter av informasjon skal organiseres og forvaltes på tvers av organisasjoner og enkeltsystemer i et distribuert miljø, er det behov for en forståelse av informasjon som er gyldig på global basis. Behovet for en slik global informasjonsmodell er blant annet beskrevet av *EU-NSF Digital Library Working Group on Interoperability between Digital Libraries* i [53], hvor en global informasjonsmodell er et av lagene i et rammeverk for interoperable digitale bibliotek. Informasjonsmodellen utgjør fundamentet for de øvrige aspekter ved digitale bibliotek.



Figur 1.1: Lagene i et rammeverk for interoperable digitale bibliotek

En arkitektur for digitale bibliotek kan derfor best beskrives ved to innfallsvinkler eller akser:

- **Den informasjonssentrerte aksen** hvor fokus er på informasjonsobjektene. Vi trenger en global, fleksibel, generell og omforent forståelse av informasjonsobjektene vi skal håndtere, slik at digitale bibliotek baseres på en felles oppfatning av det innhold de skal kommunisere om og

utveksle. Dette er tema for første del av denne rapporten, hvor vi ser på informasjonsobjekter (kap. 2), metadata (kap. 3), digital informasjon (kap. 4), identifikatorer (kap. 5) og relasjoner (kap. 6).

- **Den tjenestesentrerte aksen** hvor fokus er på interoperabilitet mellom digitale bibliotektjenester i et åpent distribuert system. Dette beskrives fra to sider, den teknologiske infrastrukturen vi tar utgangspunkt i (kap. 7), og den systemarkitektoniske hvor fokus er på hvordan vi skal modularisere funksjonaliteten i digitale bibliotek (kap. 8).

I tradisjonell systemutvikling fokuseres det ofte også på modellering av tidsaspektet – hvordan systemets tilstand forandres over tid og hvordan interaksjonen er i systemet. Dette aspektet har vi lagt mindre vekt på fordi dette er en side som er nærmere en implementering enn det som er fokus for dette prosjektet.

Kapittel 2

Informasjonsobjekter

2.1 Informasjon og data

Informasjon er et tvetydig ord som brukes på mange måter. Michael Buckland identifiserer tre forskjellige måter å bruke ordet informasjon på [23]:

- **Informasjon som prosess.** Når vi blir informert, endres det vi vet, og informasjon er derfor en del av prosessen hvor noen blir informert.
- **Informasjon som kunnskap.** Informasjon er det som meddeles når noen blir informert. I denne bruken av ordet er informasjonen noe uhåndgripelig eller abstrakt som kunnskap, meninger eller tro. For å formidle informasjon-som-kunnskap må den subjektive og konseptuelle informasjonen uttrykkes eller beskrives på en fysisk måte, f.eks. som tekst.
- **Informasjon som ting.** Fysiske objekter som data og dokumenter karakteriseres som informasjon fordi de er informative.

En tilsvarende kategorisering av hva informasjon er, finner vi gjengitt i [155], men her er dette gruppert i:

- **Det objektive informasjonsbegrepet** som tar utgangspunkt i informasjon som noe nøytralt og absolutt. Informasjon eksisterer uavhengig av mottaker, og er en enhet som venter på å bli brukt. Informasjon endres ikke på sin vei til mottaker selv om informasjonen kan endre representasjonsform.
- **Det subjektive informasjonsbegrepet** er basert på at informasjon er knyttet til det å informere. Informasjon er dermed noe som oppstår i prosessen "å informere". Det vi lagrer er bare tegn eller data som først er informasjon når de presenteres for brukere som blir informert.

Vi skal ikke gå videre på diskusjonen av informasjon i denne rapporten. Vår bruk av betegnelsen er basert på den mer trivielle oppfatningen at digitale bibliotek inneholder eller integrerer distinkte objekter som er kilde til informasjon.

I denne sammenhengen er det mest naturlig å ta utgangspunkt det objektive informasjonsbegrepet hvor informasjon er en fysisk enhet, det som Buckland kaller *informasjon-som-ting*.

Et skille det er viktig å avklare, er forholdet mellom data og informasjon[1]. I Store norske leksikon (1989) finner vi denne beskrivelsen:

> Data, flertall av datum (lat. noe som er gitt).
> I dagligtale kjensgjerninger, faktiske opplysninger, f.eks. personlige data som alder, høyde o.l. I forbindelse med databehandling betegner data enhver fysisk representasjon av opplysninger, viten, meninger etc. i motsetning til innholdet, som kalles informasjon. Representasjonen kan bestå av tekst (skrift på papir) av lyd-, lys-, eller elektriske signaler, i hullmønstre på hullkort m.m. Mens data kan behandles maskinelt og overbringes mekanisk, kreves det tenkende vesener for forståelse av den informasjon som er representert.
>
> Informasjon
> (lat.), undervisning, underretning, opplysning, viten. I elektronisk databehandling skiller man mellom data og informasjon. Dette skillet svarer til det man trekker mellom bokstavrekken som gir et ord form på papiret, og ordets faktiske betydning ...

En annen beskrivelse av disse begrepene er gitt av Sølvberg i [156]:

> Data are numbers or letters and represent objective facts, presented without any judgement or context. Information is data that is placed in context, and is endowed with relevance and purpose.

Med bakgrunn i dette kan vi f.eks. si at et dokument er informasjon og ikke data, fordi dokumentet gir en helhetlig kontekstramme for de data det inneholder – en bruker kan forholde seg til dokumentet uten å være avhengig av en ytre kontekst, f.eks. et brukergrensesnitt. En liste med tall er derimot ikke informasjon før vi selv er i stand til å sette disse verdiene i en kontekst, f.eks. ved å kjenne hva verdiene representerer, når de ble generert, eller hvor de kommer fra.

2.2 Informasjonsobjekt

Som en overordnet størrelse for enhetene vi forvalter i digitale bibliotek, har vi valgt å benytte *informasjonsobjekt*. Uttrykk som program, fil, dokument, artikkel o.l., er egnet som dagligdagse betegnelser på instanser av informasjonsobjekter, men i mange sammenhenger er disse uttrykkene lite egnet som formelle typebetegnelser, fordi de er relaterte til en spesifikk bruk. Det som er

[1] Vanligvis inkluderer slike diskusjoner data, informasjon og kunnskap, men for enkelhets skyld har vi her utelatt kunnskap.

programvare eller dokument i én kontekst, kan være en datafil i en annen. Felles for alle disse er at de kan håndteres som enheter med identitet – objekter.

I Internett-sammenheng brukes ofte "resource" synonymt med vår bruk av informasjonsobjekt [15]:

> A resource can be anything that has identity. Familiar examples include an electronic document, an image, a service (e.g., "today's weather report for Los Angeles"), and a collection of other resources. Not all resources are network "retrievable"; e.g., human beings, corporations, and bound books in a library can also be considered resources.

Tilsvarende finner vi også uttrykk som "web resource" og "Internet resource", men det å relatere ressurser til en spesifikk teknologi eller plattform, er uegnet for digitale bibliotek. Vi har derfor valgt å bruke "informasjonsobjekt" fordi det vektlegger informasjonsaspektet på en teknologi-uavhengig måte. Objektbegrepet er valgt fordi dette er blitt en vanlig betegnelse for distinkte enheter.

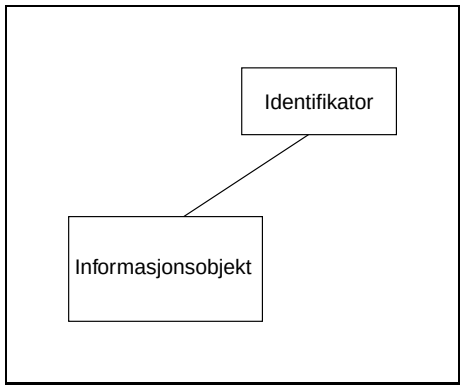
I digitale bibliotek-sammenhenger brukes ofte uttrykket "digitale objekter". Dette stammer fra en artikkel av Kahn/Wilensky (se kap. 4.1), som flere prosjekter har hentet sitt begrepsapparat fra. Vi har valgt å ikke bruke dette uttrykket, fordi vi ønsker å vektlegge at digitale bibliotek ikke er avgrenset til kun digital informasjon.

Logiske og fysiske informasjonsobjekter

Et informasjonsobjekt kan være både en logisk og en fysisk enhet. I forvaltning av informasjonsobjekter er det ofte behov for logiske enheter uavhengig av de enkelte fysiske eksemplarene eller kopiene, men når informasjonen skal formidles eller benyttes av brukerne, er det selvsagt nødvendig med en representasjonsform som kan håndteres (f.eks. en fil på en diskett) og sanses (f.eks. en utskrift på papir eller et bilde på en skjerm). IFLAs *Functional requirements for bibliographic records*, som er grundig beskrevet i kap. 3.7, er en informasjonsmodell som definerer enhetene *verk*, *uttrykk*, *manifestasjon* og *eksemplar* (produktentitetene), og beskriver relasjonene som kan finnes mellom disse. Denne modellen er et relevant utgangspunkt, ikke bare som grunnlag for bibliografiske beskrivelser, men også som en konseptuell modell av informasjonsobjekter.

Uavhengig av om informasjonsobjektet er en logisk eller en fysisk ting, har det identitet. Det er noe vi kan identifisere og referere til. Identifikatorer er viktige hjelpemidler også for å forvalte logiske størrelser, fordi disse kan fungere som representasjoner for de logiske størrelsene i et informasjonssystem. Bruken av identifikatorer er en sentral del av digitale bibliotek, og dette diskuteres mer inngående i kap. 5. En identifikator er et objekt[2] som er assosiert til et informasjonsobjekt som vist i fig. 2.1. Her benyttes modelleringsspråket UML, og informasjonsobjekt og identifikator er klasser som er forbundet med en assosiasjon.

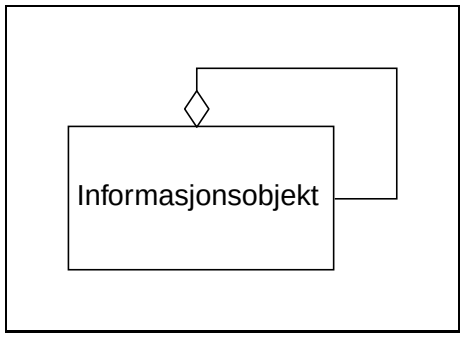
[2]Som oftest er en identifikator bare en tekststreng, men til en identifikator kan det også være knyttet metadata om det som identifiseres.



Figur 2.1: Informasjonsobjekt med identifikator

Sammensatte informasjonsobjekter
Et informasjonsobjekt kan være en elementær enhet, men vi kan ofte dele et informasjonsobjekt i flere delenheter som igjen kan defineres som selvstendige informasjonsobjekter, f.eks. artiklene i et tidsskrift. Tilsvarende kan vi opprette nye informasjonsobjekter ved å sette sammen andre informasjonsobjekter, f.eks. et kompendium med artikler eller en samling av programvare. Disse aspektene gjelder også for den fysiske sammensetningen av et informasjonsobjekt. Et digitalt dokument kan være basert på mange filer selv om det på skjerm eller papir fremstår som en samlet enhet, og det kan ha en indre logisk struktur av kapitler og avsnitt selv om det bare består av én fil.

Illustrasjon 2.2 viser dette uttrykt i UML ved hjelp av aggregering. I vanlig språk kan dette uttrykkes slik: et informasjonsobjekt kan bestå av andre informasjonsobjekter, og et informasjonsobjekt kan inngå som del av et annet informasjonsobjekt.

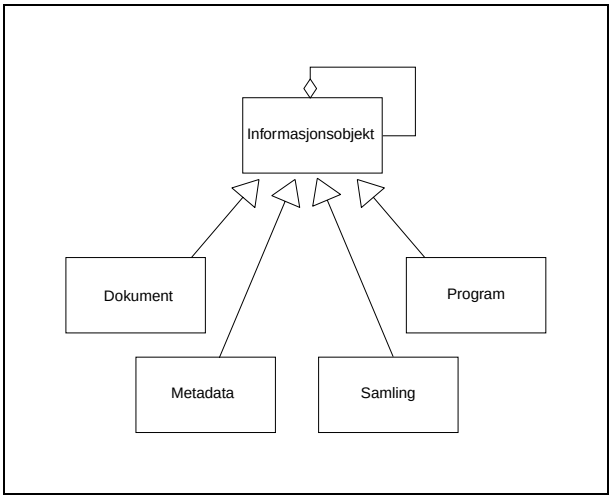


Figur 2.2: Informasjonsobjekt

Subtyper

Informasjonsobjekt er en overordnet størrelse, men i de fleste sammenhenger er det mer spesialiserte objekter vi skal håndtere. Dokumenter og programvare kan være forskjellige spesialiserte typer av informasjonsobjekter, og vi kan også definere metadata og samlinger som typer av informasjonsobjekt. Alle disse forskjellige subtypene har det til felles at de har identitet.

Illustrasjon 2.3 viser hvordan dette kan uttrykkes i UML. Kombinert med aggregerings-egenskapen har vi da en modell som fanger opp både de logiske og fysiske strukturelle aspektene ved informasjonsobjekter, enten dette er dokumenter som er satt sammen av flere filer, dokumenter med integrerte metadata, samlinger av dokumenter og metadata, samlinger av samlinger, programvare med integrert dokumentasjon, osv.



Figur 2.3: Informasjonsobjekt med subklasser

Statisk og dynamisk informasjon

I mange tilfeller, f.eks. for publiserte artikler o.l., vil informasjonsobjektene være stabile over tid (statiske). For digitale informasjonsobjekter kan vi riktignok endre filformat, og se artikkelen på skjerm eller på papir, men selve innholdet og meningen – informasjonen – er den samme.

Et informasjonsobjekt kan også være dynamisk. Informasjonsobjektet kan endre innhold over tid, eller informasjonsobjektets innhold kan fremtre forskjellige avhengig av bruk eller bruker. Denne dynamikken kan være tilstede uten at informasjonsobjektet endrer identitet. Dagens værmelding er informasjon som endres over tid, men objektet som produserer denne informasjonen kan være det samme (f.eks. en web-side). En samling av dokumenter som brukerne kun har tilgang til via et søkegrensesnitt, vil fremtre forskjellig avhengig av hvordan søkeuttrykket er formulert.

Dette dynamiske aspektet er spesielt fremtredende for digital informasjon. På Internett og World Wide Web er det lite som skiller et statisk dokument fra en interaktiv tjeneste. Web-dokumenter kan være interaktive og ligne på tjenester, mens tjenestene produserer informasjon og fremtrer som dokumenter. Fellestrekk er at de begge har identitet og kan adresseres via de samme mekanismene. Vår bruk av informasjonsobjekt som overordnet betegnelse, omfatter alle informasjonsobjekter på denne skalaen.

2.3 Dokumenter

2.3.1 Hva er et dokument?

Spørsmålet i overskriften er ikke trivielt, og synet på ordet dokument[3] er forskjellig i forskjellige fagmiljøer. Før Internett har diskusjonene beveget seg rundt spørsmålet om hva som skal kunne passere som dokument (gjenstander, skrift, bilder). I en Internett-tid er spørsmålene om avgrensning, bevaring og dynamikk av dokumenter sentrale. Det er derfor viktig å ha en bevissthet om hva er et dokument er.

Vi starter her med en mer generell tilnærming til spørsmålet i overskriften. De norske katalogiseringsreglene [160] definerer dokument slik:

> Dokument. 1. Et verk, en gruppe verk eller en del av et verk i en hvilken som helst fysisk form, publisert, utgitt eller betraktet som en enhet; gir som sådan grunnlaget for en bibliografisk beskrivelse. 2. (Manuskripter). Brukes i den fysiske beskrivelsen om det enkelte skriftstykke, aktstykke i en manuskriptsamling.

Sentralt i katalogiseringsreglenes definisjon står ordet *verk* som defineres slik:

> en sammenhengende informasjonsmengde, et intellektuelt produkt som utgjør en avsluttet helhet. Et verk er et immaterielt begrep, mens en publikasjon eller dokument er en materiell fremtredelsesform (en utgave) av dette verket.

Et dokument er altså – etter katalogiseringsreglene – et verk som er manifestert i en eller annen fysisk form som kan være gjenstand for bibliografisk beskrivelse (funksjon).
I den norske Straffelovens § 179 (om dokumentforfalskning) heter det:

> Ved Dokument forstaaes i denne Lov enhver Gjenstand, som i Skrift eller paa anden Maade indeholder et Tilkjendegivende, der enten er af Betydning som Bevis for en Ret, en Forpligtelse eller en Befrielse fra en saadan eller fremtræder som bestemt til at tjene som Bevis.

[3] dokument, fra latin *documentum*, belærende eksempel, bevis.

Straffeloven knytter også dokument til en fysisk framtredelsesform, men relaterer det til en funksjon i rettslig sammenheng.

Michael Buckland [24, 25] har i to artikler en gjennomgang av diskusjoner rundt dokumentbegrepet de siste hundre år. Vi gjengir her noen hovedtrekk i hans framstilling. Han påpeker hvordan ordning av dokumenter har vært knyttet først til ordet *bibliografi*, dernest til *dokumentasjon*, og i våre dager til *informasjon*. Buckland stiller følgende spørsmål:

> *If 'documentation' is what you do to or with documents, how far could you push the meaning of 'document' and what are the limits to 'documentation'?*

Et godt spørsmål, men det spørs om ikke 'dokumentere' og 'dokumentasjon' - ihvertfall på norsk - omfatter virksomhet også med andre objekter, for eksempel kan man dokumentere at et bestemt handlingsmønster gjør at et edb-system oppfører seg slik eller slik. Handlingsmønsteret og edb-systemet blir ikke oppfattet som dokumenter av den grunn.

Buckland tar for seg diskusjoner som foregikk i første halvdel av dette århundre om hva et dokument er. Han gjengir følgende definisjon:

> Nowadays one understands as a document any material basis for extending our knowledge which is available for study or comparison" (Schürmeyer,1935).

Et underorgan i Folkeforbundet ga følgende definisjon :

> Any source of information, in material form, capable of being used for reference or study or as an authority. Examples : manuscripts, printed matter, illustrations, diagrams, museum specimens, etc.

Suzanne Briet (1894-1989, bibliotekar og dokumentalist) starter med forutsetningen:

> A document is evidence in support of a fact" og fortsetter "[a document] is any physical or symbolic sign, preserved or recorded, intended to represent, to reconstruct or to demonstrate a physical or conceptual phenomenon.

Hun er nokså direkte :

Objekt	**Dokument ?**
Stjerne på himmelen	nei
Foto av stjerne	ja
Stein i elv	nei
Stein i museum	ja
Dyr i naturen	nei
Dyr i zoo	ja

Flere av definisjonene over nevner et aspekt som også går igjen i definisjonen av DLO[4] - nemlig en viss form for bestandighet over tid (preserved, recorded, available, bevares uforandret).

Ovennevnte definisjoner er nokså vide i sitt syn på hva som godtas som dokumenter (nemlig også gjenstander under visse forutsetninger). Men det fins også definisjoner som innsnevrer:

> A document is the repository of an expressed thought (Duyvis (1894-1961) - arbeidet i FID).

Denne definisjon legger hovedvekten på det tankemessige, det åndelige og hvordan det er lagret.

Ranganathan (indisk bibliotekteoretiker) så på dokument som synonym for: 'embodied micro thought' on paper 'or other material, fit for physical handling, transport across space, and preservation through time'.

2.3.2 Digitale dokumenter

Buckland oppsummerer i sin andre artikkel at synet på dokument har beveget seg fra å legge vekt på den fysiske formen til å vektlegge den funksjonen informasjonen framtrer med. Som eksempel nevner han en logaritmetabell som tidligere forelå i bindsterke verk, men som nå kan framtre som digitale oppslagsverk der man ikke kan vite om det er en tabell som brukes eller om logaritmer blir beregnet i øyeblikket etter forespørsel. Hva er den funksjonelle forskjellen mellom de trykte bøkene og den interaktive tjenesten?

Det er fristende i første omgang å foreta en avgrensning på vegne av det vi driver med i bibliotekene. Vi beskjeftiger oss ikke med gjenstander som steiner, vikingsverd, utstoppete dyr, selv om disse kan betraktes som dokumenter når de plasseres i en museumssammenheng.

Derimot beskjeftiger bibliotekene seg med bildende kunst, grammofonplater, videoer, lydbånd, cd-er og bøker. I den grad vi beskjeftiger oss med dokumenter av den første typen (gjenstander), så er det i form av avbildninger.

Vi har holdt oss til Folkeforbundets definisjon:

> En hvilken som helst kilde til informasjon som kan gjøres til gjenstand for referanse (sammenlikning), studium, eller som autoritet. Eksempler : manuskripter, trykte skrifter, illustrasjoner, diagrammer, museumsgjenstander.

2.3.3 Nettdokumenter

I sin natur er alle nettdokumenter avbildninger som kan gjengi en eller flere av de virkelige objektenes egenskaper (multimedia!). Dermed kan nettdokumenter - i egenskap av avbildninger - lett omfatte framstilling av andre ting

[4] DLO - **D**okument**L**iknende **O**bjekt. Et tidlig mål for Dublin Core var å dekke beskrivelsen av DLO.

enn den snevre definisjonen av dokument. Tilbake står da aspektet med det tidsbestandige og stedsuavhengige. Vi kan definere et nettdokument slik:

> et nettdokument er en digitalisert avbildning [av noe] som kan gjøres til gjenstand for studier og sammenlikning og derved bidra til økt kunnskap.
>
> Avbildningen framstår fysisk likt under ellers like betingelser for ulike personer til ulike tider og steder. Avbildningen kan overføres via et digitalisert nettverk.

En viktig egenskap ved denne definisjonen er at den avgrenser seg mot interaktive tjenester. Denne egenskapen kan betraktes som en svakhet i en nettverden av interaktive tjenester og dynamiske dokumenter. Når det kommer til katalogisering kan disse bare beskrives på et allment nivå, nærmest som en beskrivelse av et tidsskrift (og ikke dets artikler). Denne avgrensningen må ikke føre til at man lukker systemene inne. I digitale læringsmiljøer skal bibliotekenes dokumenter og metadata inngå som en integrert del av et system som også omfatter programmer, interaktive tjenester osv.

2.3.4 Filer og dokumenter

I nettverdenen er det også et annet aspekt. All informasjon som framtrer på skjermen eller som en utskrift på papir, har sitt opphav i elektroniske signaler (bit-strømmer) fra filer, fra programmer eller fra kombinasjoner av disse.

Grunnlaget for beskrivelsen er den framtredelsesformen bit-strømmen tar, slik at vi kan sanse den. Det er framtredelsesformen vi må betrakte som dokument, ikke filen. I en situasjon der vi trenger hjelpemidler for å sanse informasjonen trengs likevel beskrivelse av hvilke metoder (programmer) som er nødvendige for å få filen sansbar.

Vi tar ikke hensyn til operativsystemets fragmentering av filer på disken eller nettverkets oppdeling av bit-strømmen i enkle pakker som samles når de kommer fram til mottakeren.

Likevel er det ikke så enkelt, for et dokument kan framstå oppdelt, f.eks. som en innholdsfortegnelse på skjermen med lenker til hvert enkelt kapittel. Et dokument kan framstå som enhetlig, men kan være satt sammen av mange filer (bilder i en web-side kan være slik).

Avgrensning i en verden av hypertekstlenker er komplisert. Vi trenger informasjon som kan samle de filene som kan få dokumentet til å framstå som en enhet. Vi må også vite hvordan vi skal betrakte slike oppdelte dokumenter når de skal beskrives. Skal hver fils framtredelsesform betraktes som et eget dokument, eller skal framtredelsesformene samlet betraktes som dokument? Det siste er mest nærliggende, i motsatt fall kunne vi ende opp i en situasjon der det ville være naturlig å katalogisere hver enkelt side og hver enkelt illustrasjon i en bok.

De digitale dokumentenes komplekse natur vil kreve beskrivelser på flere nivåer, dels dokumentet som helhet, dels de komponentene som dokumentet er sammensatt av. I kapittel 3 kommer vi nærmere inn på de forskjellige behovene.

Vi trenger også informasjon om den logiske og fysiske formen som filen inntar, med tanke på bevaring gjennom konvertering til nye formater tilpasset ny maskin- og programvare.

Det er vanskelig å forutsi hvilke problemer vi på lang sikt vil ende opp i på dette feltet, for eksempel med hensyn til multimediadokumenter, hypertekstlenker osv.

Vi har innført begrepet informasjonsobjekt for å møte den situasjonen som er beskrevet ovenfor. Dette er en overbygning som gir mulighet for aggregering. Et dokument er et informasjonsobjekt, og kan som sådant være sammensatt av flere informasjonsobjekter (en HTML-fil og bilder).

Avhengig av konteksten kan det være nødvendig med beskrivende opplysninger på lavere nivåer enn dokumenter, men dette vil foregå på et mer teknisk plan.

Kjennetegnene ved dokumentene omfatter karakteristika på mange plan. Noen trekker vi ut for å lage kataloger og oversikter som gir oss mulighet til å foreta en vurdering av dokumentet på forskjellige plan uten å ha dokumentet selv i hendene. Andre har et mer teknisk eller administrativt siktemål. Mer om dette i kapittel 3.

2.4 Samlinger

Samlinger i biblioteksammenheng kan ses fra flere synsvinkler. Vi har samlinger som intellektuell ressurs, samlingens inndeling etter materialtyper, etter bibliografisk behandling, eller inndeling i mindre samlinger rettet mot spesielle brukerbehov.

Hensikten med å lage faglige litteratursamlinger er å trekke fram informasjonen som inneholder den kunnskapen som er mest faglig relevant for brukerne og den som er mest brukt. Det er enklere for brukerne å gå til biblioteket enn å kjøpe boka. Å kjøpe boka vil i mange tilfeller også være umulig fordi omløpstida er meget kort, og bøkene blir bare i salg i kort tid. Bevaring er derfor en viktig side ved samlingen.

Samlingen har også en viktig funksjon som fellesressurs. Den stilles til rådighet for hele brukergruppen og representerer som sådan et tiltak basert på en solidarisk tankegang som er nokså fjernt fra dagens markedslibraistiske ideologi.

Den solidariske tankegangen utvides ytterligere ved at det enkelte bibliotek stiller sine samlinger til rådighet for andre bibliotek gjennom fjernlåns- og fjernkopiordninger.

2.4.1 Samlingsbygging

Samlingsbygging og seleksjon baserer seg på to forhold: for det første *etterspørsel* etter dokumentene, og for det andre *verdien* dokumentene har for kvaliteten av samlingen, dens brukbarhet som forsknings- og undervisningsredskap *over tid*.

Når vi snakker om etterspørsel, brukes ofte uttrykket *just-in-time*. Brukt aleine er det et uttrykk for en markedstenkning omkring bibliotekets tjenester. Når det gjelder dokumentverdi for samlingen, blir denne ofte nedsettende omtalt med uttrykket *just-in-case*. Disse to uttrykkene blir ofte stilt opp som motsetninger - enten det ene eller det andre. Det er viktig å holde fast ved at en god bibliotektjeneste må ivareta begge disse aspektene - både etterspørsel og samlingsverdi.

En samling er til for å brukes. Ikke desto mindre er det klart at "just-in-case"-prinsippet ikke kan forkastes. En av hovedoppgavene for biblioteket er langsiktig bevaring. Forkaster man dette siktemålet og utelukkende satser på øyeblikkets informasjonsbehov, kan biblioteket erstattes av en kommersiell informasjonsmegler.

Med voksende tilgang til elektronisk materiale kan forholdet mellom etterspørsels- og verdibetraktning for samlingen forrykkes. Den intellektuelle prosessen som ligger i seleksjon og organisering med et klart siktemål, kommer under press. Den langsiktige verdibetraktningen på samlingen kan få en mindre plass enn den kortsiktige etterspørselen.

Av økonomiske grunner har verdibetraktningen allerede lenge vært på vikende front, men for forskningsbiblioteket er det viktig ikke å miste den helt av syne.

2.4.2 Samling som intellektuell ressurs

Det skal ikke bare tas inn dokumenter som er etterspurt i dag, man må også ha langsiktige mål med samlingsoppbyggingen. En samling er langt mer enn summen av de enkelte dokumentene. *Utvalget* og *den bibliografiske organisering* gjør samlingen til en intellektuell ressurs. Dette stiller store krav til bibliotekarbeidernes kunnskaper om faget - og fagets generelle utviklingstrekk. Gjennom de valg som gjøres - både når det gjelder hva som anskaffes og ikke minst ved det som *ikke* anskaffes - får samlingen sin kvalitet for forsknings- og undervisningsmiljøet. Sagt på en annen måte: oppbyggingen av samlinger tar sikte på å støtte pågående og framtidig forskning og undervisning innen bibliotekets målgruppe. Seleksjon spiller derfor en viktig rolle i bibliotekets arbeid for å nå sine mål. Biblioteket skal ikke bare være et sted for *tilgang* til alle mulige dokumenter. Det skal være en samling organisert for en målgruppe. Dette vil også gjelde i den situasjonen vi står oppe i nå med økt tilgang til elektronisk informasjon via datanettverket.

Et universitetsbibliotek får i tillegg oppgaven å knytte fagene sammen der

de har berøringspunkter, og ved det gi en bredde i framstillingen av de enkelte fag. De skal altså kombinere det spesielle med det generelle.

I et bibliotek med tilgang til store elektroniske samlinger på nettverket kommer behovet for seleksjon inn igjen med full tyngde. Det er viktig når man setter i gang med store elektroniske samlinger at bibliotekene kan organisere disse inn som en del av sin lokale samling og med sin målgruppe for øyet. Det er ikke noe mål å gjøre alle ressurser *like* tilgjengelig. Det som er mest relevant for målgruppa, må få en mer framskutt posisjon ved å bli bibliotekfaglig organisert sammen med lokale samlinger.

Samtidig må man ikke glemme at teknologien etter hvert vil gi muligheter for individuelt tilpassede samlinger. Man bør tilstrebe systemer som gir brukerne anledning til å manipulere samlingene etter sine egne behov (ekstrahere og organisere deler av samlingen utfra de metadataene som foreligger).

2.4.3 Samlinger etter bibliografisk profil

Brukerne av bibliotektjenester har ofte vanskelig for å utnytte den bibliografiske strukturen for å finne den delen av samlingen som tilfredsstiller deres behov.

Den bibliografiske strukturen samlingen får gjennom konsistent katalogisering og klassifikasjon, kan utnyttes til å presentere deler av samlingen som har en spesiell innretning.

Når brukeren møter tjenesten gjennom brukergrensesnittet, er det viktig å foreta en avveining av hvor mye av denne strukturen som skal presenteres *før* søket i form av klasseskjemaer, søkeformularer osv, og hvor mye man skal bruke strukturene til å ordne trefflistene *etter* søket.

De bibliografiske strukturene kan også i mye større grad brukes til å presentere for brukerne ferdige spesialsydde søk som i form av lenker trekker ut spesialsamlinger basert på form, sjanger, emne eller andre bibliografiske elementer.

2.4.4 Integrerte samlinger

I digitale bibliotek vil samlingen kunne bestå av mange samlinger med forskjellige metadatasett. I BIBSYS kan man for eksempel i dag aksessere både bokkatalog og artikkelreferansedatabaser innenfor samme kontekst. Når man skal integrere eksterne og interne samlinger, er det av avgjørende betydning at de bibliografiske dataene fra de ulike samlingene i størst mulig grad samsvarer både på det semantiske og syntaktiske plan. For å kunne integrere samlinger, trengs det beskrivelser av de enkelte samlingenes katalogiseringsregler for å kunne få til bibliografisk samvirke. Dette er et problemområde som må vies spesiell oppmerksomhet.

Integrasjon av samlinger har også med materialform og tilgjengelighet å gjøre. Bøker kan befinne seg på åpne hyller eller i lukkete magasiner, lokalt

eller i stor fysisk avstand. Elektroniske dokumenter kan befinne seg på filtjenere lokalt eller rundt om i verden. Gjennom det digitale bibliotek må alle dokumentene framstå på en enhetlig måte slik at brukeren raskt kan fastslå tilgjengelighet med hensyn på tid, rettigheter, kostnad og teknologi.

Bevaring er et viktig spørsmål å diskutere når det gjelder integrasjon av eksterne og interne samlinger. Vi ser i dag at det i økende grad inngås avtaler – på kommersiell basis – om tilgang til elektronisk materiale man ikke selv har fysisk kontroll over. I mange tilfeller mister man tilgangen til materialet i det øyeblikket man sier opp avtalen, eller om den kommersielle aktøren går konkurs eller velger en annen avtaleprofil.

En annen side ved slike kommersielt baserte eksterne samlinger, er at de ofte innbefatter klausuler som hindrer formidling til andre enn primærbrukerne. Dette undergraver det solidariske nettverket som bibliotekvesenet representerer. Det kan da være spørsmål om man skal betrakte en slik ekstern samling som en del av sin lokale samling eller ikke. Uansett syn, det er sentralt at for gode brukergrensesnitt at eksterne samlinger rent bibliografisk blir integrert med lokale samlinger.

Kapittel 3

Metadata

3.1 Bakgrunn

Ordet *metadata* dukket opp i Internett-samfunnet rundt 1995. Ordet har spredt seg raskt, uten at man samtidig hadde en felles forståelse av hva ordet betød. En definisjon som ofte blir brukt er **data om data**. Dette er den mest allmenne formen, men som definisjon er den av relativt liten verdi som grunnlag for forståelse av hva det dreier seg om. De fleste som bruker denne definisjonen, er derfor raske med å eksemplifisere. Ofte blir det vist til bibliotekenes katalogposter.

Å definere betyr bokstavelig talt å avgrense. Når man skal definere et ord som metadata, er det viktig som en første avgrensning å gi informasjon om hvilken kontekst man opererer i. Ordet metadata blir brukt av mange forskjellige fagmiljøer, og gis innhold i tråd med det.

Her er eksempler på definisjoner hentet forskjellige steder på nettet og fra noen artikler:

- Thus, metadata is a definition or description of data.
  http://www.whatis.com/meta.htm

- Data about data. Metadata describes how and when and by whom a particular set of data was collected, and how the data is formatted. Metadata is essential for understanding information stored in data warehouses.
  http://webopedia.internet.com/TERM/m/metadata.html

- Metadata. An encoded description of an information package (e.g., an AACR2 record encoded with MARC, a Dublin Core record, a GILS record, etc.); the purpose of metadata is to provide an intermediate level at which choices can be made as to which information packages one wishes to view or search, without having to search massive amounts of irrelevant full text.
  http://orc.dev.oclc.org:5103/metamarda-l/msg00117.html

- METADATA. Literally, data about data. The ”data” is information provided as a means to describe an information providing entity. A bibliographic record (information seen on the computer screen of the online library catalog) acts as a document surrogate in OPACs to find materials (e.g. books or journal articles), in the library (or sometimes outside the library). Metadata placed in electronic document HEADs provide the same basic information with the exceptions that it is not visible to the viewer unless he deliberatly looks at it in the Document Source found under the View option, and the user does not need to view the head to find the document.
  http://www.valdosta.edu/~gfrost/metadata.html)

- ”metadata is a succinct and systematic set of information that references, and can be used to efficiently and accurately retrieve, a larger set of informatio” (Robert DeCandido, in the Internet Searcher's Handbook, 2nd Edition).
  http://www.pla.org/metadata.htm)

- The most common definition of the term 'metadata' is data about data – information that describes other information. For example, this web page has an author, a title, a date of creation, and a unique Internet address; this information constitutes the metadata about this page.
  http://adam.ac.uk/adam/metadata.html

- Metadata is structured data which describes the characteristics of a resource. Chris Taylor.
  http://www.library.uq.edu.au/iad/ctmeta4.html

- Metadata in its broadest sense is data about data. The familiar library catalogue records could be described as metadata in that the catalogue record is 'data about data'. [...] The term metadata is increasingly being used in the information world to specify records which refer to digital resources available across a network. Rachel Heery [74]

- Metadata really is nothing more than data about data; a catalog record is metadata; so is a TEI header, or any other form of description. Caplan [29]

- Metadata is data about data. It describes the attributes and contents of an original document or work. Milstead [128]

- metadata is data associated with objects which relieves their potential users of having to have full advance knowledge of their existence or characteristics. DESIRE-project [45]

- "Metadata" is the Internet-age term for structured data about data. Typical examples are library catalog records, bibliographic headers in

Web pages, terms of use" statements, and ratings.
http://www.iei.pi.cnr.it/DELOS/REPORTS/metadata.html

- **Norsk dataordbok**: Metadata - Data om *dataelementer*, inklusive databeskrivelsen, og data om eierskap, tilgangsbaner, tilgangsretter og dataflyktighet [ISO]

Disse definisjonene inneholder hver for seg elementer som må med i en fullstendig definisjon. Noen har litt om ulike formål, noen litt om bruksområder, noen antyder hva slags type data som inngår, noen antyder struktur. Noen mener at metadata inngår som en del av de dataene de beskriver. Ingen av dem er utfyllende. I tillegg har de en ganske snever innretning.

En hører ofte utsagn om at metadata skal *erstatte* katalogisering. Dette er et falskt valg. Mange av definisjonene ovenfor nevner eksplisitt bibliotekenes katalogposter som eksempler på metadata. Spørsmålet er derfor heller hva slags metadata man skal operere med, hvilken detaljering dataene skal ha, hvor mye kontroll det skal være på dataene, og hvem det er som skal generere dem. For å avgjøre dette må man stille seg mål for hva man ønsker å oppnå med metadataene.

Ved Stanford Digital Library prosjektet [14] løftes *metadata* opp på et annet plan. Deres mål er å integerere heterogene databaser som hver for seg opererer med forskjellige typer metadata. Dette er en situasjon BIBSYS allerede er i. For å få til samvirke mellom heterogene databaser trengs opplysninger også om

- hvilke samlinger som fins
- informasjon om hvordan objekter beskrives innenfor hver enkelt samling
- protokollrelaterte opplysninger om de enkelte tjenestene

Komplette overganger mellom metadataformater kan bare skapes dersom det er enighet på de tre områdene *semantikk, detaljeringsgrad* (granularitet) og *katalogiseringsregler*.

3.2 Definisjon og avgrensning

Vi vil definere metadata slik:

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

Vi vil her avgrense *metadata* til beskrivelse av objekter i samlinger, og ikke ta for oss beskrivelse av samlinger.

Vi behandler ikke metadata som beskriver interne dokumentstrukturer à la XML og SGML, selv om disse kan sies å ha noe med formidlingen å gjøre.

Vi behandler heller ikke automatisk genererte metadata av den typen som søkemaskiner bruker for å rangere trefflistene sine.

Endelig vil vi avgrense til formelle beskrivelser. En formell beskrivelse forutsetter et sett med navngitte opplysninger.

Vi vil ikke avgrense metadata til å gjelde opplysninger som følger dokumentet som en integrert del av det. Metadata kan følge dokumentet, eller de kan befinne seg løsrevet, for eksempel i en katalog. Et dokument kan inneholde en referanse til en metadatatjeneste som kan levere metadata for dette dokumentet. Vi vil heller ikke avgrense til metadata laget etter bestemte katalogiseringsregler. Det er mange produsenter av metadata, og det foregår mye arbeid for å skape en felles semantisk forståelse av dataelementer og gjennom det legge grunnlaget for fornuftige overganger mellom de forskjellige metadataformatene (crosswalks).

Vi vil i neste avsnitt komme inn på hva formidlingen omfatter.

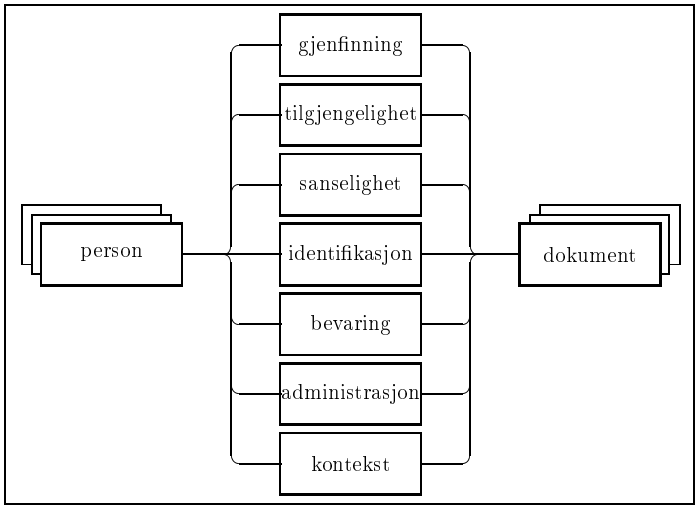
3.3 Formål

Utgangspunktet for at man i det hele tatt snakker om metadata, er behovet for opplysninger som hjelpemiddel og kontroll i formidlingen av dokumenter til personer. Metadata skal være en representasjon av dokumentene som skal inngå i systemer til erstatning for dokumentet selv - et dokumentsurrogat. Denne type metadata skal gi tilstrekkelig med opplysninger om dokumentet slik at brukeren kan gis grunnlag for å fastslå om det virkelige dokumentet er av interesse. Men dette er bare én side ved metadata.

Metadata dreier seg dels om opplysninger som støtter gjenfinning, dels om opplysninger som administrerer tilgangen, dels om opplysninger som autentiserer dokument (og person), dels opplysninger om programvare og teknologi som støtter formidlingen. Noen typer opplysninger faller innenfor mer enn én kategori.

De fleste typene opplysninger gjelder for alle dokumenter, papirbaserte eller digitale, og AACR2 har regler for å registrere det meste. Men enkelte av opplysningene må vies spesiell oppmerksom når dokumentene er digitale og tilgjengelig på et nettverk. I slike tilfeller må en del av metadataene formaliseres i sterkere grad slik at de kan maskinbehandles på en meningsfull måte. En papirbasert hovedoppgave med klausul får man ikke tilgang til uten menneskelig mellomkomst. I et slikt tilfelle må klausulen formuleres i klartekst, det er ikke så nøye med syntaksen. En digitalisert nettversjon av hovedoppgaven må ha data knyttet til seg slik at maskinene kan håndheve klausulen. Det krever streng syntaks på metadataene.

For hver enkelt kategori av metadata må man stille konkrete mål for hva man ønsker oppnå. Først da kan man si hva slags data man trenger og hvilke krav man må stille til dem.



Figur 3.1: Metadatakategorier som styrer formidlingen av dokumenter til personer.

3.3.1 Metadata som støtter gjenfinning

I FRBR-modellen (se 3.7) finner man en rekke eksempler på hvilke egenskaper ved de ulike entitetene som støtter gjenfinning av verk, uttrykk, manifestasjon eller eksemplar. I tillegg redegjøres det for hvilke relasjoner som har betydning for gjenfinning.

Det er et viktig poeng at gjenfinning ikke bare dreier seg om å finne ett bestemt dokument, men også om å kunne orientere seg i og finne fram i en bibliografisk struktur. Gjenfinningsdata dreier seg ikke bare om beskrivelse av hvert enkelt dokument i en samling, men om å sette disse dokumentene i sammenheng.

Målene med bibliografiske data er stilt flere ganger det siste hundreåret. Først ute var Cutter [37] i 1904 som stilte opp følgende mål:

> To enable a person to find a book of which either
>
> (A) the author
> (B) the title
> (C) the subject
> is known
>
> To show what the library has
>
> (D) by a given author
> (E) on a given subject
> (F) in a given kind of literature

To assist in the choice of a book

(G) as to its edition (bibliographically)
(H) as to its character (literary or topical)

Disse målene kan virke enkle i sin utforming, men stiller sterke krav til de bibliografiske dataenes inndeling, form og innhold. For hvert enkelt punkt angir Cutter krav til dataene som må oppfylles for at målene kan nås.

FRBR-modellen har formulert tilsvarende mål i en mer generalisert form og angir hvilken betydning de enkelte egenskapene og relasjonene har for oppfylle forskjellige deler av målsettingene (find, identify, select, obtain).

3.3.2 Metadata om tilgjengelighet

Tilgjengelighet og rettighetsopplysninger for papirdokumenter er som regel angitt i dokumentet selv. Et papirdokument er likevel stabilt. For nettdokumenter endres dette forholdet. Tilgjengelighet dreier seg om flere ting.

- **Tidsrom**. Nettdokumenter har en tendens til å forsvinne, bli flyttet, bli forandret, være av begrenset varighet, eller bli liggende igjen etter at de er uaktuelle. Det er derfor i mange tilfeller nødvendig å angi tidsrom for gyldighet eller fra/til hvilket tidspunkt dokumentet er gyldig.

  I bibliotek er utlånstid, utlånsstatus viktige data som berører tilgjengelighet.

- **Opphavsrett**. Dette gjelder data både om hvem som har opphavsretten og eventuelt hvilken organisasjon man skal henvende seg til for å få klarert opphavsretten.

- **Pris**. Sannsynligvis vil det i framtida være behov for data angående automatisk betaling for tilgang til informasjonsobjekter.

- **Klausuler**. Som for papirdokumenter kan dokumenter være klausulert for kortere eller lengre tidsrom.

- **Lokalisering/eierskap**. I biblioteksammenheng vil samlingstilhørighet være viktige data som berører tilgjengelighet. Dette gjelder ikke bare de fysiske samlingene i biblioteket, men også de nettdokumentene bibliotekene har skaffet seg tilgang til gjennom abonnementsordninger.

Deler av dette angår adgangskontrollsystemet.

3.3.3 Metadata om sanseliggjøring

Tradisjonelt har ikke opplysninger om metoder for sanseliggjøring vært nødvendige fordi dokumentene kunne sanses direkte eller med enkle hjelpemidler.

Med teknologi involvert i formidlingen må man også oppgi hvilke hjelpemidler som må til for at dokumentet skal sanses for å gi brukeren mulighet til å vurdere om den rette teknologien er på plass eller tilgjengelig. Kan vi fortsatt spille voksruller med musikk-kutt på ? Hva med 78-plater, 45-plater, 33-plater, CD-er, MD-er og DVD-er?

Selv for et så dagligdags fenomen som avspilling av populærmusikk kan altså de skiftende fysiske formene være et problem, både 'forover' og 'bakover' i tid. Problemene forsterkes med de digitale dokumentene. Hvem kan i dag lese en 8-tommers diskett[1]? Men det stopper ikke der, for selv om disketten kunne leses, så kan dataene inneholde en logisk struktur som må tolkes av et eller flere programmer før de kan sanses.

3.3.4 Metadata om identifikasjon/verifikasjon/autentisering

Et trykt dokument kan verifiseres ved å sammenlikne med originalen. Man kan i tillegg ved kjemiske og fysiske teknikker fastslå om en påstått original er ekte.

Digitale dokumenter er mye enklere å endre uten spor. Det er i tillegg viktig å kunne fastslå om et dokument er det det gir seg ut for å være. Vi trenger metoder og teknikker for å fastslå dokumentets ekthet. For tida pågår det arbeid med digital vannmerking og skjult skrift (steganografi) for blant annet å sikre at det er mulig å sammenholde det digitale dokumentet med en original med de samme egenskapene eller med visse metadata som kan følge dokumentet eller befinne seg på et tilgjengelig sted.

Mange dokumenter likner hverandre. Vi trenger derfor også metadata som gjør det mulig å skille dokumenter fra hverandre. Katalogiseringsreglene inneholder metoder for dette formålet. Det kan for eksempel dreie seg om å markere hvilken fysisk form dokumentet har, eller hvilken utgave det dreier seg om. Som oftest er det en kombinasjon av metadataelementer som gjør det mulig å skille.

3.3.5 Metadata som kontekst

I mange tilfeller – for eksempel i museumssammenheng – der dokumentene er avbildninger av gjenstander eller rett og slett fotografier av steder og/eller personer, vil det være nødvendig med en beskrivelse ut over det rent tekniske og innholdsmessige. Gjenstandene eller fotografiene må plasseres i en sammenheng som kan gi dem ytterligere informasjonsverdi. Det kan være opplysninger om datoer, steder, personer, og andre forhold.

3.3.6 Metadata som støtter bevaring

Bøker kan sanses direkte, og er relativt bestandige når de er trykt på syrefritt papir. I mange tilfeller er det nok å sette en bok på hylla og man kan hente

[1] Hovedoppgaven til en av forfatterne av denne utredningen er lagret på et magnetbånd skrevet i redigeringssystemet *runoff* på en DEC10-maskin under operativsystemet TOPS-10 i 1980. Kan den personen melde seg som kan få noe ut av dette til en rimelig pris?

den ut 200 år seinere og den er like brukbar. Mikrofilm trenger bare en enkel teknologi for å bli lest (linse og lys), og vil vare lenge om de er produsert og oppbevart under de rette betingelsene. Videobånd, kassettbånd og magnetbånd reduseres over et tidsrom på 10-20 år, og teknologien som trengs er også komplisert og foranderlig.

Digitale dokumenter endrer dette bildet. Sammenholdt med trykte dokumenter er digitale dokumenter mye mer utsatt for forfall over relativt korte tidsperioder. Dette skyldes:

- Digitale dokumenter er ikke sanselige. Det trengs både
  - programvare og
  - maskinvare

  for å gjøre dataene sanselige. Programvare og maskinvare utvikler seg meget fort, og disse er uløselig knyttet til hverandre. Digitale dokumenter må konverteres med korte mellomrom, og vil derfor fjerne seg lenger og lenger fra originalen.

- Det digitale mediets elektromagnetiske natur er også en faktor som må tas hensyn til. Magnetbånd og disker er i seg selv ikke stabile. Selv med stabil teknologi vil det være nødvendig med stadig kopiering over på et friskt medium.

For å sikre videreføringen av digitale dokumenter (altså ikke bevaringen) til ny maskin- og programvare trengs nøyaktige metadata om de tekniske betingelsene som dataene er lagret under. Dataene må konverteres mens teknologien ennå er tilgjengelig. Data har alltid gått tapt, men ved videreføringen kan det være et spørsmål om utvelgelse av hva som skal videreføres. Spesielt viktig blir det å vurdere dette spørsmålet for digitale dokumenter, for om man lar være å videreføre, kan de være tapt for alltid. Et eksempel illustrerer dette:

The US Census Bureau lagret i begynnelsen av 60-åra folketellingsdata fra 1960 på det som ble betraktet som et permanent medium (magnetbånd). I 1976 fastslo US National Archives at sju serier av aggregerte data fra folketellinga i 1960 hadde historisk interesse. Utstyr for å lese magnetbåndene var for lengst forsvunnet, og byrået brukte 3 år på den ingeniørmessige oppgaven å få data over på industristandardformat [126].

I tillegg: når man først får dataene ut fra mediet, oppstår problemet med å tolke dataenes logiske struktur. Denne må også være beskrevet av metadatene. Bruk av SGML[2] er en måte å kode tekstlig informasjon der koding inkluderer sin egen tolkning. Dette kan vise seg å være en måte å lagre tekstlig informasjon på som er velegnet for videreføring.

Uansett: et godt bevart trykt dokument er det samme over tid; det er stabilt. Et digitalt dokument beveger seg over tid lenger og lenger vekk fra sin originale form. Dette aspektet er det viktig å ha in mente.

[2]Standard Generalized Markup Language - et språk for å beskrive tekstlige strukturer.

3.3.7 Metadata som administrativt underlag

Mange dokumenter inngår i en dynamisk prosess der dokumentene både utvikler seg og har ulik status avhengig av hvor i prosessen man befinner seg. I slike situasjoner trengs også metadata som kan fortelle om dokumentenes versjoner, bakgrunnsdokumenter og resultatdokumenter. Dertil trengs metadata om dokumentets gyldighet og status innenfor prosessen.

I et bibliotek fins det også administrative metadata om hvor et dokument befinner seg i den interne prosessen og i forhold til sirkulasjon (utlån/reservering).

3.4 Overordnete kjennetegn på metadata

3.4.1 Stabile og dynamiske metadata

Det går fram av de ulike metadataformene at noen av dem er nokså stabile, mens andre vil endre seg.

Gjenfinningsdata er stabile, det samme må gjelde identifiserende data (identifikatorer både for dokumenter og metadata). Data om tilgang, om fysiske og logiske lagringsformater, om administrative forhold må antas å være dynamiske ettersom de berører forhold som inngår i en mer eller mindre kontinuerlig prosess. Et dokument – papirbasert eller digitalt – kan forsvinne eller bli flyttet. Dette berører metadataene på tilgangsnivå (hyllesignatur, nettadresse), men det berører ikke metadata som beskriver forfatter, tittel eller emne.

Metadataenes karakter av å være enten stabile eller labile vil være bestemmende for teknologiske løsninger for behandlingen av dem.

3.4.2 Autoriserte metadata

Etter hvert som bruken av metadata griper om seg, er det nødvendig også å definere metadata som verifiserer og autentiserer metadata. Søkemaskiner som samler metadata må ha mulighet til å vurdere påliteligheten.

Allerede nå er det mye juks ute og går med metadata for å komme høyt på søkemaskinenes trefflister. Dette er en av hovedgrunnene til at søkemaskiner legger mer vekt på sine egne rangeringsalgoritmer enn på metadata som måtte følge et nettdokument.

Er det forfatteren selv, hans faglige organisasjon eller et bibliotek som følger bestemte katalogiseringsregler som har produsert metadataene? Svaret på dette vil ha betydning for hvilken vekt de skal tillegges i søkemaskinene og ved import til bibliotekkatalogene som grunnlag for katalogisering.

3.4.3 Organisering av metadata

Metadata kan følge dokumentet eller de kan ligge utenfor. På det papirbaserte området kan dette illustreres med at mange amerikanske bøker inneholder en

metadatapost på tittelbladets bakside (CIP[3]-post) som inneholder mer eller mindre fullstendig bibliografisk informasjon produsert av Library of Congress på basis av forhåndsinformasjon om dokumentet. Enkelte forlag bringer også emneklassifikasjon i enkelte av sine publikasjoner innen spesielle fagområder. Disse dataene blir brukt som grunnlag for lokal katalogisering og klassifikasjon.

Bøker kan også inneholde en identifikator i form av Library of Congress number eller BNB-nummer som viser til en post i en database der fullstendig bibliografisk beskrivelse er lagret og kan hentes ut til eget bruk.

I disse to tilfellene er det enten en forbindelse fra dokument til eksterne metadatatjenester, eller dokumentet inneholder selv metadataene. Men i de aller fleste tilfellene har ikke dokumentprodusenten laget noen forbindelse *fra* dokumentet *til* metadataene. Bibliotekene lager derimot en forbindelse ved å klistre inn strekkoder (i BIBSYS: dokid) eller skrive inn tilvekstnummer i det fysiske eksemplaret. Ved hjelp avslike *identifikatorer* (dokid, tilvekstnummer) kan dokumentets metadata hentes fra katalogen.

Denne bokbaserte modellen kan ikke uten videre overføres til den digitale verden. Det er for eksempel ikke like enkelt for bibliotekene å påføre de digitale dokumentene sine identifikatorer. En ønskesituasjon ville være at enten produsenten eller biblioteket kunne påføre digitale dokumenter en identifikator og en adresse til en nasjonal eller lokal metadatatjeneste som på forespørsel kan levere metadata for dokumentet i det formatet som ønskes (MARC, Dublin Core, eller annet).

På nasjonalt plan burde Nasjonalbiblioteket tilby en slik tjeneste for den digitale delen av nasjonalbibliografien og oppfordre dokumentprodusenter til å inkludere f.eks. NBN i dokumentene (mer om NBN på side 116). Om dette ikke gjennomføres, må en ta til takke med en situasjon der det bare eksisterer en en-veis forbindelse fra metadata til dokument (og ikke omvendt).

Metadata skal oppfylle mange funksjoner, men dette betyr ikke at alt skal organiseres i samme enhet (f.eks. i en MARC-post). De forskjellige funksjonene kan distribueres, og forbindelsen mellom dem kan organiseres ved hjelp av hendler og identifikatorer i separate systemer.

På denne måten kan forbindelsen mellom søkesystem, adgangskontrollsystem og dokumenter lages.

3.4.4 Samvirke mellom metadataformater

Samvirke på nettverket (også kalt interoperabilitet) er et sentralt element som må med i vurderingen ved etablering av metadataskjemaer. I første rekke gjelder dette på semantisk nivå: de forskjellige formatene må understøtte samme forståelse av de enkelte elementene (hva er en tittel, hva er en forfatter, hva er en del av et verk, osv). Det neste nivået er det innholdsmessige: navn må registreres på samme måte, man må ha en felles forståelse av datoformater,

[3] **CIP** - Cataloguing in publication.

osv. Det er et problem i forbindelse med samvirke at forskjellige systemer har forskjellig oppfatning av detaljeringsgraden av opplysningene (granularitet).

Beskrivelse og forståelse av disse nivåene kan danne grunnlag for å lage overganger mellom de ulike metadataskjemaene. Slike overganger vil alltid være på det svakeste skjemaets premisser. Ikke desto mindre kan 'fattige' skjemaer fungere som bindeledd mellom flere ellers rike skjemaer om de er bygget på samme forståelse av dataelementene, og registrerer innholdet i dataelementene etter samme regelsett (normalisering). Ansvaret for vedlikeholdet av slike tabeller for overganger er også et problem. Foreløpig er det den enkelte systemleverandør som må sørge for å holde orden på dette, hvis man ikke velger å benytte 'fattige' skjemaer som en de facto standard.

3.5 Metadata : formateksempler

Mange miljøer lager metadataformater for å holde oversikt over sine samlinger. Det kan være generelle formater som dekker et bredt utvalg av dokumenttyper; det kan være formater som dekker behov for organisering av spesielle typer dokumenter. Noen formater er lukket for utvidelser, andre har en åpen struktur.

Enkelte metadataformater er rene skjemaer for utfylling av data, uten regler for utfylling bortsett fra en ren semantisk forståelse av hva slags type opplysning som ligger i de enkelte feltene, andre er koplet sammen med et strengt regelsett for dataene: hvor opplysningene skal hentes fra, hvilken form de skal ha, og hvilke verdier de i visse tilfeller skal ha.

Vi vil her gi en grov presentasjon av noen formater. Dette er ikke en fullstendig liste, men noen eksempler på bredden. I tillegg C har vi gjengitt noen konkrete eksempler fra de ulike metadataformatene.

3.5.1 Formater med spesiell innretning

IAFA-templates
Akronymet står for *Internet Anonymous FTP Archives*. Formatet er egentlig en metode for å beskrive objekter i filarkiver for filoverføring (FTP = File Transfer Protocol). Formatet bygger på tre forskjellige typer felt:

- **Enkle felt** (Plain fields). Felt med en enkelt opplysning. Feltet kan ikke repeteres.
- **Repeterbare felt** (Variant fields). Felt som kan gjentas etter behov, når det fins flere verdier av opplysningen som skal registreres.
- **Klyngefelt** (Cluster fields). Klyngefelt omfatter et sett av felt (delfelt) som bindes sammen.

I emneportalsystemet ROADS (Resource Organisation And Discovery in Subject-based services) brukes IAFA-systemet. I ROADS er det definert et tjuetalls

forskjellige registreringsskjemaer som dekker forskjellige dokumenter og tjenester man kan finne på nettverket (document, service, software, mailarchive, sound, image, video, osv). Se eksempel på side 188.

Hver enkelt bruker av ROADS kan i prinsippet etablere sine egne katalogiseringsregler, men det fins anbefalte katalogiseringsregler for de ulike elementene i skjemaene, se [43].

VRA Core Categories for Visual Rescources
Visual Resources Association [169] er en amerikansk organisasjon etablert for å fremme forskning, utdannelse og samarbeid innen feltet visuelle ressurser. For å imøtekomme behovet for dokumentasjonsstandarder i bildesamlinger, blant annet for å bidra til distribuert/delt katalogisering i en nasjonal bildedatabase, etablerte VRA i 1993 en datastandardiseringskomite (DSC - Data Standards Committee). Målet var å imøtekomme dette områdets økende behov for å forvalte komplekse visuelle samlinger i et nettverksmiljø ved å:

- identifisere, utvikle og formidle informasjon som fremmer en standard deskriptiv praksis.
- formidle visuelle ressursers interesser og behov overfor relaterte offentlige og kommersielle virksomheter.
- etablere forbindelser og samarbeide med lignende interessegrupper i museer, arkiv og bibliotek.

Basert på en undersøkelse med data fra over 60 institusjoner i USA og Canada, og sammenlignbare metadataformater for relatert materiale, er det opprettet et sett basiskategorier for beskrivelser av visuelle ressurser. Data relatert til samlingsforvaltning er ikke tatt med i kategoriene, da dette er data med mange særegenheter, som i tillegg kun er av lokal interesse. Versjon 3.0 av "Core Categories for Visual Resources" (CCVR) ble lansert i juni 2000 [168] og inneholder retningslinjer for konvertering til Dublin Core Metadata Element Set (se nedenfor). Arbeid er i gang for å lage en overgang til MARC-formatet.

CCVR er ment som retningslinjer for utvikling av lokale databaser og katalogposter. Det er ikke en spesifisert instruksjon for systemutvikling eller datamodeller, men er ment som et fundament for utvikling av systemer. Det er ikke et krav at alle kategoriene skal benyttes, og det er heller ikke begrensninger på det å opprette lokale tilleggskategorier. Retningslinjene inneholder definisjoner på hvilke felter som er repeterbare, samt anbefalinger for bruk av andre katalogiseringsstandarder som tesauri og katalogiseringsregler. Kategoriene er ment å dekke beskrivelse av både "works" (dvs. gjenstander og objektet) som er avbildet, og selve det visuelle dokumentet. Se eksempel på side 182.

Feltkatalog for NKKMs EDB-prosjekter
Norge er i en særstilling i verden ved at vi har et landsomfattende, organisert innsamlingsarbeid av fotografiske bilder. Dette er et resultat av Norsk Kulturråds utredning i 1976 om bevaring av gamle fotografier. I regi av Sekretariatet for fotoregistrering (SFFR) og Norges Kunst og Kulturhistoriske Museer (NKKM) ble det på 1980-tallet utviklet edb-baserte registreringssystemer for bilder (Freg) og kulturhistoriske gjenstandsmateriale (Greg).

Datamodelleringen for dette programmet tok i stor grad utgangspunkt i de standardiserte registreringskort som var i bruk i innsamlingsarbeidet. Programvaren er benyttet ved en rekke museer og andre institusjoner.

I forbindelse med utviklingen av neste generasjon av denne programvaren ble det i regi av NKKM i 1992 utviklet en feltkatalog for fotografier, kulturhistorisk gjenstandsmateriale, billedkunst, kunstindustri og museumsbibliotek [139].

Feltkatalogen er et generelt dataformat uavhengig av databaseverktøy. I tillegg til felt og subfeltdefinisjoner inngår også kodelister for autoriserte sted og personroller, og fotografisk typebestemmelse. Katalogen er omfattende og har som mål å dekke de behov for registreringsdata som er kjent, selv om det er opp til de enkelte institusjoner hvilke felt de vil benytte. Et eksempel på registreringskjema er gjengitt på side 184.

Denne feltkatalogen har også vært utgangspunktet for utvikling av Galleri Nor ved Nasjonalbibliotekavdelinga i Rana, selv om dette systemet avviker noe fra den opprinnelige feltkatalogen.

CDWA
Categories for the Description of Works of Art [61] ble utviklet av Art Information Task Force og sponset av Getty Art History Information Program (AHIP) og College Art Association (CAA). Målsetting for formatet er å være et utvekslingsformat mellom databaser, som plandokument for utvikling av nye databaser eller utvidelse av eksisterende databaser. Dette formatet fokuserer på "moveable objects and their images", i stor grad kunst. Formatet består av 26 hovedkategorier med egne underkategorier.

CIMI
Computer Interchange of Museum Information er et rammeverk av standarder for utveksling av data mellom museer. Arbeidet som ble startet opp av Museum Computer Network videreføres av The CIMI Consortium. Underlagt CIMI finnes et prosjekt for å gjøre museenes innhold tilgjengelig, *Cultural Heritage Information Online*, som blant annet bruker SGML og Document Type Definitions (DTD) som struktur for å beskrive museumsobjekter som utstillingskataloger og bilder [32].

FGDC
The Federal Geographic Data Committee (FGDC) initierte i 1992 et arbeid for et felles sett med terminologi og definisjoner for dokumentasjon av geografisk relatert data (geospatial data) som flyfotografier, kart og satelittbilder. Dette resulterte i *Content Standard for Digital Geospatial Metadata* (CSDGM), som vanligvis blir omtalt som FGDC-standarden [34]. Dette er et komplekst format på over 300 felter. Mange av disse feltene er rettet mot spesifikke behov for geografisk relatert materiale, men vi finner også felter som *Title*, *Abstract* og *Keywords*. Det er utarbeidet en SGML DTD for CSDGM.

3.5.2 Generelle formater

IEEE Learning object metadata (LOM)
Dette utkastet til standard[4] definerer syntaks og semantikk for de egenskaper som trengs for en fullstendig beskrivelse av et *læringsobjekt*.

Et læringsobjekt blir definert som en entitet, digital eller ikke-digital, som kan bli brukt, gjenbrukt eller referert til i teknologistøttet læring.

LOM har blant annet som mål:

- å sette studenter og lærere i stand til å søke, hente og bruke læringsobjekter
- å legge grunnlag for deling og utveksling av læringsobjekter på tvers av teknologistøttete læringssystemer
- å legge til rette for å lage læringsobjekter i enheter som kan bli kombinert eller oppsplittet på meningsfull måte
- å legge til rett for automatisk eller dynamisk komposisjon av individuelle leksjoner for enkeltpersoner

Den grunnleggende strukturen i LOM er basert på 9 metadatakategorier:

- *Generelt* omfatter alle kontekstuavhengige egenskaper og de semnatiske deskriptorene for ressursen.
- *Livssyklus* omfatter egenskaper som har med ressursens livssyklus å gjøre.
- *Meta-metadata* omfatter egenskaper som gjelder beskrivelsen selv (altså ikke ressursen som beskrives).
- *Teknisk* omfatter tekniske egenskaper ved ressursen.
- *Utdanning* omfatter læringsmessige og pedagogiske egenskaper ved ressursen.

[4] se http://ltsc.ieee.org/doc/wg12/LOM_WD4.htm

- *Rettigheter* gjelder egenskaper som beskriver betingelser for bruk av ressursen.
- *Relasjoner* omfatter opplysninger som knytrter ressursen til andre ressurser.
- *Annotasjon* tillater kommentarer angående den pedagogiske bruken av ressursen.
- *Klassifikasjon* beskriver ressursen emnemessig.

Hver hovedkategori er videre inndelt. Formatet består totalt av rundt 60 felt. En overgang fra dette formatet til Dublin Core-formatet (se nedenfor) er under utvikling.

Dette formatet er det naturlig å studere videre for mulig anvendelse i digitale læringsmiljøer.

MARC

MAchine Readable Cataloguing. MARC er dels et system for å merke bibliografisk informasjon, dels et system for å utveksle slik informasjon elektronisk. En internasjonal standard (ISO 2709) angir hvordan denne informasjonen kan formidles. Utvekslingen foregår ved et system for å merke de enkelte bibliografiske elementer med koder (tag-er). På side 179 har vi gjengitt MARC data i ISO 2709 form. Et alternativ til ISO 2709 som blir stadig mer aktuelt, er å bruke XML til å transportere MARC-data. Et eksempel på MARC-data i XML er gitt på side C.1.

MARC tar opp i seg alle elementene som er definert gjennom de anglo-amerikanske katalogiseringsreglene (AACR2), men det fins ulike nasjonalt tilpassete MARC-standarder med sine egne utvalg av koder (felter), med sin egen forståelse av hva som skal registreres, og med lokale tillegg. Standarden for utveksling kan likevel brukes, den sier ikke noe om hvilke felt som skal brukes og hvordan de skal fylles ut (ut over et visst minimum), bare hvordan feltene skal struktureres som en lang streng. Forskjellene mellom MARC-formatene gjør det i visse tilfeller umulig å utveksle data. Dette gjelder for eksempel opplysninger om Dewey-klassifikasjon.

IFLA har utviklet et standard MARC-format (UNIMARC [81]) med det siktemålet å bygge bro mellom ulike MARC-varianter (konvertering via dette formatet). Det er også laget en rekke dataprogrammer for konvertering mellom MARC-formater.

Det foregår arbeid for å tilpasse store leverandørers MARC-formater til hverandre. Et eksempel på dette er normaliseringen av USMARC (USA) og CAN/MARC (Canada) til det nye formatet MARC21 [117] og samarbeidet mellom OCLC og Niedersächsische Staats- und Universitätsbibliothek Göttingen [55] for å lette utvekslingen av amerikanskprodusert og tyskprodusert bibliografisk informasjon.

Med sitt utgangspunkt i katalogiseringsreglene er MARC et generelt format som dekker de material- og dokumenttypene som er definert der. Men MARC har også et annet utgangspunkt. MARC-formatet er nært knyttet til kortkatalogens måte å organisere bibliografiske data på. MARC-formatets skille i felter som angår hovedinnførsel (1XX-feltene) og biinnførsler/lenker (7XX-feltene) og andre søkenøkler (6XX-feltene) viser dette. Nettopp fordi MARC-formatet bare tok sikte på å automatisere hva man hadde (kortkatalogen) istedenfor å se på hvilke nye muligheter teknologien gir og heller bygge på det, sitter vi i dag med millioner av komplisert oppbygde MARC-poster som det kan vise seg vanskelig eller umulig å overføre fullstendig til en datamodell a la FRBR. Problemene er bl.a. beskrevet av Michael Gorman [64].

TEI-headers - MARC DTD
Text Encoding Initiative [51] er et prosjekt som har pågått i snart 10 år for å lagre og organisere elektroniske tekster. Tekstene lagres med SGML. En sterk motivasjon i prosjektet er å presentere formalisert informasjon i dokumentets hode som kan fungere som hovedkilde for katalogisering, slik at en katalogpost skal kunne genereres mer eller mindre direkte fra dokumentetes opplysninger om seg selv. Formaliseringen er derfor lagt tett opptil AACR2 og MARC, slik at import av data til kataloger skal bli enkel.

I denne sammenhengen kan det være verdt å nevne at *Library of Congress* har gjennomført et prosjekt [118] *MARC DTD*. Prosjektets mål var å etablere en dokumenttype-definisjon (DTD) for representasjon av bibliografisk informasjon i SGML som følger MARC, og som derfor lar seg overføre til MARCs kodenivå uten tap av data. Prosjektet tok også sikte på å produsere programvare for slik overføring. De første versjonene av slik programvare forelå i 1997 (skrevet i Perl). Prosjektet er nært forbundet med TEI. På side 181 har vi gjengitt utdrag av bibliografisk informasjon i SGML-form basert på en MARC DTD.

SOIF
Summary Object Interchange Format er en attributt-verdi-protokoll (eller metadataformat) som benyttes i Harvest-arkitekturen utviklet ved University of Colorado at Boulder [20]. Harvest er basert på såkalte "gatherer" som genererer metadata og gjør disse tilgjengelige for brokers" som indekserer data og gjør disse søkbare. SOIF benyttes for å formidle metadata mellom de forskjellige komponenter i arkitekturen. Formatet er basert på enkle attributt-verdi-par, uten å standardisere hvilke attributter som skal benyttes. Harvest-manualen lister likevel opp en del vanlige attributtnavn som *Abstract*, *Author*, *Description* og *Keywords* samt en rekke attributter som beskriver mer administrative opplysninger knyttet til generering og formidling av av SOIF-posten [73].

GILS
Government Information Locator Service [65]. USAs myndigheter etablerte GILS for å hjelpe publikum til å finne fram i og få tak i offentlig informasjon.

En utvidelse av prosjeket er *Global Information Locator Service* (også forkortet GILS) som har de samme målene, men utvidet til internasjonalt plan, og prosjektet er derfor sterkt knyttet opp mot internasjonale standarder for kommunikasjon og informasjonsutveksling (f.eks. ISO 23950/ANSI Z.39.50).

GILS inkluderer et omfattende format for å beskrive informasjonsressurser gjennom GILS Core Elements. Det er også definert overganger mellom GILS-formatet og MARC.

Meningen er at informasjon registrert i GILS skal kunne nås direkte over offentlig nettverk eller fra formidlere som bibliotek eller kommersielle tjenesteleverandører.

Dublin Core

Dublin Core Metadata Element Set var i utgangspunktet (1995, [47]) et forsøk på å etablere et felles format for nettdokumenter med formålet *to improve resource discovery on the net* på grunn av søkemaskinenes utilstrekkelighet. Tanken var at produsentene skal inkludere metadata i DC-form i sine nettdokumenter og derved gi søkemaskiner anledning til å tilby søk basert på disse som en forbedring i forhold til fritekstsøk og rangering. Etterhvert har målsettingen for anvendelsesområdet blitt mye bredere, også anvendt frakoplet dokumentet. Dette har sammenheng med utviklingen av formatet selv i tråd med kritikk og forslag fra forskjellige miljøer.

DC definerer nå 15 elementer (felt), [48]. Definisjonen er syntaksuavhengig og tar sikte på å etablere en felles semantisk forståelse av hvert enkelt element. Formatet tar sikte på å være samlende for ulike miljøer som produserer metadata om sine samlinger (museer, arkiver, bibliotek og andre miljøer).

Dublin Core-elementene kan spesifiseres med kvalifikatorer som enten raffinerer det enkelte element (element qualifier) eller angir hvilke verdier et element kan ha (value qualifier). Verdikvalifikatorer kan enten vise til et kontrollert vokabular (f.eks. emneord eller klassifikasjonskoder) eller til regler for tolkning av informasjon i et element (f.eks. hvordan en datoopplysning skal tolkes). Våren 2000 ble det enighet om et sett av kvalifikatorer som løfter Dublin Core skjemaet kvalitetsmessig og som gjør det til et bedre redskap for kunnskapsorganisasjon enn de opprinnelige 15 rene elementene. Kvalifikatorene ble drøftet på en konferanse høsten 1999 og lansert våren 2000 [49].

Bortsett fra de reglene som følger av anvendelse av verdikvalifikatorer (value qualifiers) fins det ingen andre katalogiseringsregler for de enkelte elementene i Dublin Core.

De fleste miljøene som har tatt Dublin Core i bruk har hatt behov for å gjøre tillegg tilpasset deres behov. Dublin Core vil ikke ha mulighet til å erstatte 100% alle mulige andre metadatasystemer. Det skulle være nok å vise til AACR2/MARC-formatet, men også en titt på de øvrige formatene som er nevnt her, viser at de ulike miljøene har metadatabehov som går ut over Dublin Cores 15 elementer. En viktig rolle for Dublin Cores rolle kan derfor bli å være et slags esperanto for de ulike – mer detaljerte – metadataformatene.

Det er laget syntaktiske regler for å representere Dublin Core-data i HTML (se side 183) og XML/RDF (se side 183).

Forskjellige miljøer har laget programvare for overganger mellom Dublin Core og ulike MARC-formater og vice versa. BIBSYS har laget mulighet for å importere Dublin Core-data fra HTML-dokumenter i sitt katalogiseringsskjema for MARC-data.

Dublin Core kommer trolig til å få sentral betydning for samsøk mellom de ulike miljøene som er i ferd med å bygge opp emneportal-tjenester[5]. De fleste slike miljøer velger Dublin Core som beskrivelsesnivå. Felles format er en grunnmur i slike samsøktjenester. BIBSYS emneportal har for eksempel definert sitt metadataformat som en delmengde av Dublin Core med en del administrative tilleggsfelt.

Det er også verdt å nevne Dublin Cores mulighet som postsyntaks i forbindelse med samsøk via Z39.50-protokollen og BATH-profilen definert i tilknytning til denne.

Innenfor Dublin Core-miljøet er det opprettet interessegrupper for særegne miljøer som identifiserer problemområder i Dublin Core-skjemaet og som foreslår løsninger. En gruppe har bibliotekanvendelser som sin interesseprofil. Gruppa har også som mål å holde bibliotekmiljøet informert om Dublin Cores utvikling.

3.6 Oppsummering

3.6.1 Biblioteksektorens spesifikke behov

Bibliotekene organiserer samlinger av dokumenter, ikke bare et lager av enkeltdokumenter. Dette betyr at det gjennom registreringene genereres bibliografiske strukturer gjennom anvendelse av detaljerte katalogiseringsregler som muliggjør navigering i det bibliografiske univers. Registreringene er samlingsorientert, ikke dokumentorientert. For å få til dette kreves det menneskelig innsats. For at denne innsatsen skal bli verdsatt må edb-systemene i større grad ta i bruk de strukturene som allerede fins gjennom AACR II og MARC, og det må vurderes om registreringene skal utvides til FRBR-modellen (se avsnitt 3.7) for å legge til rette for bedre navigasjon.

3.6.2 Supplering med andre formater

MARC-formatet dekker ikke alle behov som har dukket opp gjennom framveksten av forskjellige typer nettdokumenter og den globale kommunikasjonen. MARC-formatet må derfor åpnes opp for å inkludere metadata som har en annen karakter. Dette kan kreve forandringer både på skjemanivå og kodenivå.

Sammenkopling kan for eksempel skje ved at MARC-skjemaet utvides med felter som ved hjelp av identifikatorer/lokatorer kan vise til andre metadatasett.

[5] Emneportal: organiserte samlinger av lenker til internettressurser.

3.6.3 Samvirke

Bibliotekene utveksler poster seg imellom gjennom sine standarder. Den globale kommunikasjonen åpner opp for utveksling av data med andre miljøer. Dette betyr både at bibliotekene selv må være i stand til å levere data i andre standarder og kunne motta data fra andre. Dette er ikke minst viktig i samband med etablering av systemer for digitale læremidler og interaktive læringsmiljøer. Dublin Core kan vise seg å få sentral betydning i interaksjon mellom systemer med ulike metadataformater.

3.7 FRBR-modellen

IFLA-modellen [82] – bedre kjent som *Functional requirements for bibliographic records* (heretter FRBR) – er resultatet av mange års arbeid og flere høringsrunder, for å gjennomgå katalogiseringsprinsippene.

Arbeidet startet etter IFLA-konferansen i Stockholm i 1990, og ble endelig godkjent på IFLA-konferansen i København i 1997. Sluttrapporten føyer seg inn i en historisk rekke som blant annet omfatter Cutters arbeid i 1904 [37] og Parisprinsippene i 1961 [84].

De sterke endringer i omgivelsene som har skjedd siden 1961, har reist behovet for en ny gjennomgang av katalogiseringsprinsippene og metodikken. Ikke minst har katalogteknologien endret seg betraktelig, uten at dette i særlig grad har virket tilbake på katalogiseringsreglene. Det er her nok å vise til utviklingen av edb-baserte kataloger og eksplosjonen i datakommunikasjonen. Dette stiller helt nye krav til katalogdataene for å få utnyttet den bibliografiske strukturen og for å få til internasjonalt samvirke. Det stiller også store krav til edb-systemene som opp til nå har vært bleke kopier av kortkatalogene hva angår bibliografisk struktur. Kortkataloger lever i et lukket univers, og selv om man bruker ISBD, fins det språklige formuleringer som "originaltittel", "illustratør" o.l. som fungerer lokalt, men ikke nødvendigvis i en annen katalog i et annet språkområde.

At reglene ikke er vesentlig endret, kan skyldes at målene og prinsippene er robuste, men det kan også skyldes at prinsippene er nedfelt på en slik måte i reglene at de ikke umiddelbart kan tas i bruk på en datamaskin (ikke formaliserte nok). Det er også en nokså tung prosess å snu et system med så mange millioner katalogposter verden over. Samtidig er det slik at når de bibliografiske strukturene ikke har vært gjenspeilt i edb-katalogene, så har meningen med å registrere dem forsvunnet. Dette har i visse tilfeller ført til at strukturelle komponenter i katalogiseringsreglene er blitt 'forenklet' bort ved anvendelse i ebd-systemene.

FRBR bringer disse og andre strukturer fram i lyset igjen og er formulert slik at muligheten åpnes for en sterkere grad av formalisering. Strukturene blir enklere å utnytte i datasystemene, og kan bli mer egnet til utveksling til fremme for det ultimative målet : universell bibliografisk kontroll (UBC).

De bibliografiske elementene og strukturene blir presentert i et rammeverk som kombinerer erfaring med bibliografiske beskrivelse og en analysemetode fra databasemodellering (såkalt entity-relationship-analyse).

I seg selv kan dette legge grunnlag for datasystemer som i større grad vil tillate brukerne å navigere i det bibliografiske univers.

Modellen har fått allmenn anerkjennelse for sin ryddighet og klarhet når det gjelder å presentere bibliografiske elementer og bibliografisk struktur. Ikke minst ser det ut til at modellen blir forstått i edb-miljøer, noe som lover godt for neste generasjons biblioteksystemer.

Det sies eksplisitt i rapporten at modellens intensjon er å være uavhengig av etablerte katalogiseringsregler og katalogformater. En allmenn aksept av modellen kan derfor ha som konsekvens endringer av både katalogiseringsregler og edb-systemer.

3.7.1 Målet med bibliografiske beskrivelser

Hensikten med beskrivelsene er å tilfredsstille visse forhåndsangitte ønsker fra brukerne når det gjelder å få tak i (i bred forstand) dokumenter. Brukerne i denne sammenhengen er en nokså bredt sammensatt gruppe: lesere, studenter, forskere, bibliotekansatte, forleggere, forhandlere, informasjonsmeglere, administratorer av opphavsrettigheter, osv.

Det er ikke første gang det blir formulert mål for den bibliografiske beskrivelsen: Cutter i 1904 og nevnte konferanse i Paris i 1961 gjorde det samme.

I kapitlet om krav til nasjonalbibliografiske tjenester i FRBR formuleres målene med beskrivelsen slik[6]: Den bibliografiske beskrivelsen skal sette brukeren i stand til:

- **å finne alle manifestasjoner som inneholder**
  - de verkene som en gitt person eller korporasjon er ansvarlig for
  - de ulike uttrykksformene som et gitt verk har
  - verk om et gitt emne
  - verk i en gitt rekke
  - en bestemt manifestasjon når tittelen er kjent
  - en bestemt manifestasjon når identifikatoren er kjent
- **å identifisere**
  - et verk
  - et verks uttrykk
- **å velge**
  - et verk

[6] I det innledende kapitlet formuleres det litt annerledes, vi kommer tilbake til det. Her er en kortversjon: **1**: å bruke registrerte data for å finne materiale som samsvarer med brukerens uttrykte søkekriterier; **2**: å bruke uthentete data for å identifisere det man er på jakt etter; **3**: å bruke dataene til å velge det som tilfredsstiller brukerens ønsker; **4**: å bruke dataene for å få tak i eller få adgang til det man er på jakt etter.

- et uttrykk
- en manifestasjon

- **å få tak i**
  - en manifestasjon

Allerede her er det en rekke ord og uttrykk som berører sentrale byggeklosser i modellen: *verk, uttrykk, manifestasjon, person, korporasjon, "er ansvarlig for", emne*. For å forstå modellen og målene må man gå nærmere inn på disse byggeklossene.

3.7.2 Modellens byggeklosser

Byggeklossene i modellen fordeler seg i 3 grupper.

Den første gruppa betegnes på engelsk som *entities*. Den norske oversettelsen av dette ordet er *entiteter*[7]. Det som menes i denne modellen, er de komponentene som har kjerneinteresse for brukerne av bibliografiske data, nemlig 1) produktene; 2) ansvaret for produktene, og 3) emnet for produktene.

Den andre gruppa av byggeklosser utgjøres av entitetenes *egenskaper*: et verks tittel, en forfatters navn, en emnebeskrivelse.

Den tredje gruppa utgjøres av *relasjoner* mellom entitetene, for eksempel at en forfatter *har skapt* et verk, at et verk *handler om* et angitt emne.

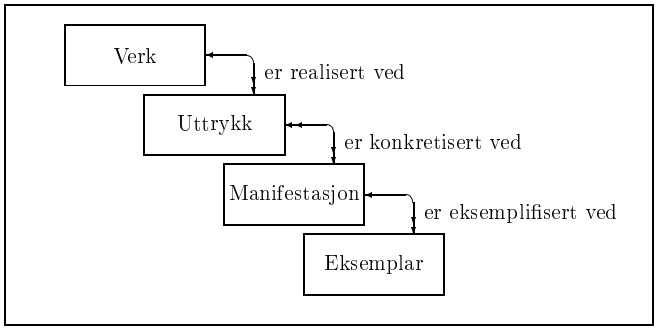
Entiteter

Gruppe 1: Produktene

Denne gruppa består av entiteter som er resultat av intellektuell eller kunstnerisk innsats. Dette omfatter både abstrakte og konkrete størrelser. Figur 3.2 viser de fire entitetene og deres innbyrdes sammenheng. Forbindelsen mellom verk og uttrykk forteller at et uttrykk bare kan være realisering av ett verk (én pilespiss), mens et verk kan uttrykkes på flere måter (to pilespisser). Tilsvarende kan en manifestasjon inneholde flere uttrykk, og et uttrykk kan få flere forskjellige manifestasjoner. Endelig kan en manifestasjon være eksemplifisert i flere eksemplarer, mens et eksemplar bare kan eksemplifisere én manifestasjon.

- **Verk**
  Verk er en abstrakt entitet som representerer et distinkt intellektuelt eller kunstnerisk produkt. Et verk konstateres ved hjelp av de ulike realiseringer - uttrykk - verket får, som en generalisering av dem.

[7] Norsk dataordbok (6.utg.) definerer det slik: Hvilken som helst konkret eller abstrakt ting som eksisterer, eksisterte eller som kunne eksistere, inklusive forbindelser mellom disse tingene. Eks.: person, objekt, begivenhet, id, prosess osv [ISO]
Andre definisjoner:
- Entity. A person, place, thing, or concept that has characteristics of interest to the enterprise. An entity is something about which we store data [123].
- The basic object that the ER model represents is an **entity**, which is a "thing" in the real world with an independent existence. [52]



Figur 3.2: Verk, uttrykk, manifestasjon og eksemplar og deres grunnleggende relasjoner.

Definisjonen av *verk* er problematisk fordi det kan være et skjønnspørsmål hvorvidt et nytt uttrykk representerer et nytt verk eller ikke. Graden av tilførsel av intellektuell eller kunstnerisk aktivitet vil avgjøre det. *My fair lady* er et nytt verk i forhold til *Pygmalion*, men filmversjonen av *My fair lady* må betraktes som et nytt uttrykk av *My fair lady*-verket (selv om det kanskje fornærmer filmprodusent og regissør). Det kan også eksemplifiseres ved gjendiktninger av poesi som kan være en ren oversettelse - dvs et nytt uttrykk ifølge modellen - eller en gjendiktning som skiller seg så mye fra originalen at det må betraktes som et nytt verk.

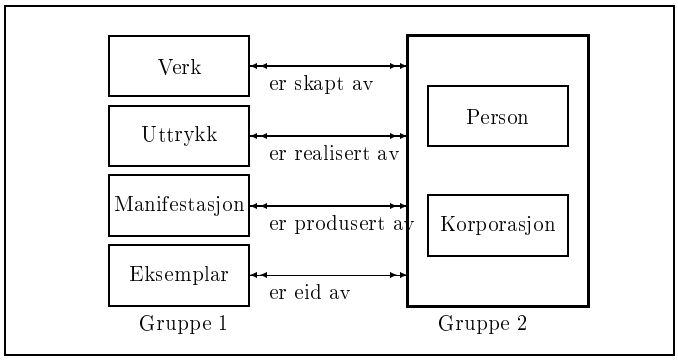
Å etablere den abstrakte entiteten *verk* gjør det mulig å samle alle uttrykk av et bestemt verk gjennom relasjoner dem i mellom.

- **Uttrykk**
  Et uttrykk er den intellektuelle eller kunstneriske formen et verk inntar hver gang det blir realisert. For eksempel vil forskjellige oversettelser representere forskjellige uttrykk av samme verk. Likeens forskjellige arrangementer av et musikkverk.

- **Manifestasjon**
  En manifestasjon er den fysiske konkretiseringen av et verks uttrykk. Et verks uttrykk kan utgis som bok, lydbok, som datafil; alle er forskjellige manifestasjoner av ett av et verks forskjellige uttrykk. En manifestasjon kan inneholde uttrykkene til flere verk (f.eks. i form av en antologi).

  En manifestasjon er både abstrakt og konkret, dels fordi det alltid er et eksemplar som utgjør den første forekomsten av en manifestasjon, dels fordi manifestasjonen representerer en generalisering av alle eksemplarene av en bestemt manifestasjon.

  FRBR understreker *manifestasjonens* abstrakte karakter, men setter likevel opp **å få tak i en manifestasjon** som mål. Det endelige målet



Figur 3.3: Ansvarsentitetene og deres grunnleggende relasjoner til produktentitetene.

må være å få tak **et eksemplar**, ikke den abstrakte manifestasjonen.

- **Eksemplar**
  Eksemplar er en konkret entitet. Det særegne ved *eksemplar*-entiteten framtrer først når eksemplar 2 tas opp til behandling.

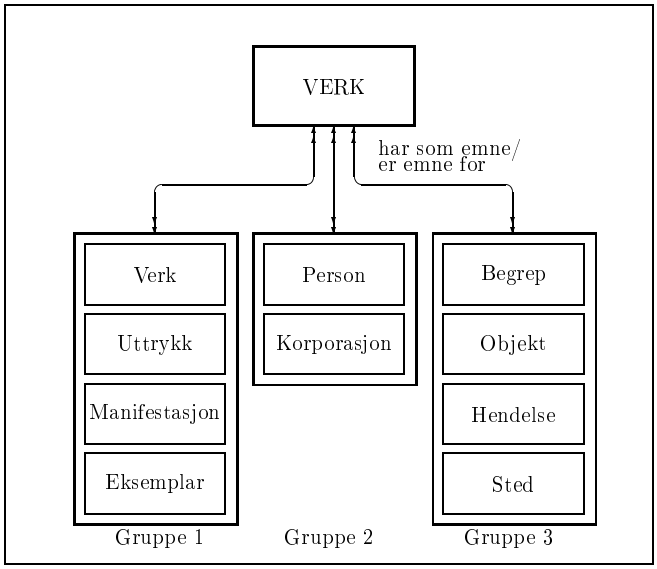
Det har vært reist en kritikk mot inndelingen i disse fire nivåene at den mangler et overordnet nivå. Den er ikke er i stand til å samle verk som hører sammen under samme paraply[8]. Et eksempel skal være forholdet mellom *Pygmalion* og *My fair lady*, et søk på det ene skal også gi treff på det andre for å gi et fullstendig bilde. Men her kan det innvendes at avledninger av verk er en sekundær opplysning som uansett vil komme fram når fullstendige opplysninger for verket presenteres.

Gruppe 2: Ansvaret
Denne gruppa omfatter de entitetene som kan være ansvarlig for de forskjellige produktentitetene. Figur 3.3 viser de grunnleggende ansvarsrelasjonene mellom de forskjellige produktentitetene og de to ansvarsentitetene (person og korporasjon). Legg merke til at et verk kan være skapt av flere (enten personer og/eller korporasjoner), og at en person eller korporasjon kan ha skapt flere verk. Tilsvarende gjelder for de andre forbindelsene (angitt ved to pilespisser).

- **Person**
  Ved å definere enkeltpersoner som entitet åpner modellen for å trekke relasjoner mellom verk eller uttrykk og den personen som er ansvarlig for verket eller uttrykket.

- **Korporasjon**
  En korporasjon er grupper av personer, sammenslutninger eller institusjoner som identifiseres ved et særskilt navn og som opptrer som en enhet.

[8]European Library Automation Group - ELAG - har reist en slik kritikk.



Figur 3.4: Et verks emner illustrert med verkets mulige emnerelasjoner til produkt-, ansvars- og spesifikke emneentiteter.

Ved å definere *korporasjon* som en entitet åpner modellen for å trekke relasjoner mellom verk eller uttrykk og den korporasjonen som er ansvarlig for verket eller uttrykket.

Gruppe 3: Emne
Denne gruppa omfatter de komponentene som brukes for å beskrive emnet for det intellektuelle eller kunstneriske produkt.

- **Begrep**
  I FRBR er *begrep* definert som entiteten som omfatter abstraksjoner som kan være emnet for et verk. Ved å definere begrep som entitet åpner modellen for å trekke relasjoner mellom verk og det som er verkets emne. I modellen er *begreper* entitet bare i den utstrekning de kan fungere som emne for et *verk*.

- **Objekt**
  Objekt er i denne modellen definert som en fysisk gjenstand som kan være emne for et *verk*.

- **Hendelse**
  Hendelse er i denne modellen uttrykk for historiske hendelser, epoker og tidsperioder som kan være emne for et *verk*

- **Sted**
  Sted er stedsnavn, både på og utenfor jorda, nåværende eller historis-

ke, geografiske posisjoner eller geo-politiske jurisdiksjoner som kan være emne for et *verk*.

I tillegg til at disse emneentitetene kan være emne for et *verk*, kan også de andre entitetene inngå som emner. Et verk kan for eksempel omhandle et annet verk, et uttrykk, en manifestasjon, et eksemplar, en person eller en korporasjon. En teateranmeldelse omhandler for eksempel en bestemt forestilling (eksemplar) av en bestemt oppsetning (manifestasjon) av den norske versjonen (uttrykk) av *My fair lady* (verk), og kan ha alle nivåer som emne.

Figur 3.4 viser hvordan et verk kan ha som emne både verk og andre entiteter fra de tre gruppene. Igjen betyr antall pilespisser på forbindelseslinjene at et verk kan omhandle mange emner (innenfor en gruppe), og at emnet kan være behandlet i mange verk.

Egenskaper

Alle entiteter er tilordnet en mengde egenskaper (attributter) som fyller bestemte funksjoner når det gjelder å nå målene for den bibliografiske beskrivelsen. Egenskapene kan tjene deler av målsetningene i større eller mindre grad. Noen egenskaper tjener gjenfinningsformål, andre målet med å identifisere entitetene eller skille dem ut fra andre liknende entiteter.

I modellen er det oppramset egenskaper for hver entitet og gjort en anlyse av hvilke mål de har betydning for oppfyllelsen av. Enkelte egenskaper er generelle for en entitet, det vil si at de kan være aktuelle for alle forekomster av en bestemt type entitet. Andre egenskaper gjelder bare spesielle forekomster av entiteten (f.eks. er *spilletid* en egenskap ved den typen av manifestasjoner som angår lydopptak).

Som eksempel kan her nevnes egenskapene til entiteten *verk*: **tittel, form, dato, andre skillende karakteristika, planlagt avslutning, målgruppe, verkets omgivelser, besetning (musikkverk), verknummer (musikkverk, opus), toneart (musikkverk), koordinater (kartverk), referanseår (stjernekartverk)**.

Som man forstår av lista, kan de 7 første av egenskapene (ikke nødvendigvis alle) brukes til å beskrive alle typer *verk*, mens de siste gjelder spesielle typer verk.

Legg merke til at skaperen av verket ikke er en egenskap ved verket, men at informasjon om ansvar skapes gjennom en relasjon til en entitet i gruppe 2 (person eller korporasjon).

Det er også viktig å merke seg at egenskaper blir arvet nedover nivåene fra *verk* til *eksemplar*. En fullstendig bibliografisk beskrivelse av *eksemplar* inkluderer egenskapene fra *manifestasjonen, uttrykket* og *verket*. De enkelte opplysningene vil spille forskjellig rolle på de forskjellige nivåene. For eksempel vil *tittel* på verknivået spille rollen som *standardtittel* eller *originaltittel* på uttrykksnivået.

Relasjoner

Barbara Tillett [166] analyserte i en artikkel i 1991 de grunnleggende bibliografiske relasjoner. Hun bygget på tidligere - ufullstendige modeller - og presenterte selv nedenstående 7 hovedgrupper av relasjoner hun på basis av empiriske studier av bibliografiske data fant i bibliografiske kataloger (Tillett hadde på dette tidspunkt ikke samme inndeling av produktentitetene som FRBR).

- **ekvivalens**
  for eksempel relasjoner mellom eksakte kopier av en manifestasjon eller mellom en reproduksjon og originalen, så lenge intellektuelt og kunstnerisk innhold er bevart og ansvaret beholdt.

- **avledning**
  relasjoner mellom en bibliografisk enhet og modifikasjoner basert på denne, for eksempel variasjoner eller versjoner av et verk.

- **beskrivelse**
  relasjoner fra en bibliografisk enhet som beskriver et annet verk, for eksempel gjennom en evaluering, en anmeldelse, en omtale, en annotert utgave.

- **helhet-del**
  relasjoner mellom et verk og dets enkelte deler, for eksempel fra enkeltbidrag i en antologi til antologien som helhet, eller til en samling eller serie.

- **vedlegg**
  relasjoner mellom en manifestasjon og materialet som følger manifestasjonen, for eksempel en diskett, et kartvedlegg.

- **sekvens**
  for eksempel relasjoner mellom enkeltverk i en serie.

- **delte karakteristika**
  relasjoner som skapes mellom bibliografiske enheter fordi de har opplysninger til felles, f.eks forfatternavn, klassifikasjonskode.

Problemet med mange av disse relasjonene i bibliotekkatalogene er at de gjennom katalogiseringsreglene får et språklig uttrykk som vanskeliggjør maskinell behandling for å aktivisere relasjonene i edb-systemene, men de fungerer bra på kortkatalogens premisser.

Gjennom å identifisere relasjonene og kategorisere dem, legges grunnlaget for en mer formell beskrivelse som også kan lette navigasjonen i det bibliografiske univers.

Tillets relasjonskategorier er godt dekket i FRBR. Vi har allerede presentert noen av relasjonene i modellen: at et verk *er realisert gjennom* et uttrykk som

Relasjonstype	**relasjon**	**invers relasjon**
Etterfølger	har etterfølger	er etterfølger av
Supplement	har supplement	er supplement til
Komplement	har komplement	er komplement til
Sammendrag	har sammendrag	er sammendrag til
Bearbeiding	har bearbeiding	er bearbeiding av
Omarbeiding	har omarbeiding	er omarbeiding av
Etterlikning	har etterlikning	er etterlikning av

Figur 3.5: Relasjoner fra *verk* til *verk*.

er konkretisert ved en manifestasjon som *er eksemplifisert ved* et eksemplar. Figur 3.3 viste de sentrale ansvarsrelasjonene, og figur 3.4 emnerelasjonen.

Denne første klassen av relasjoner kan sies å være på et grunnleggende nivå som sammen med egenskapene og entitetene *bærer* modellen.

Enhver relasjon impliserer også en invers relasjon som går den andre veien, den er den ene halvparten av et par. Ansvarsrelasjonsparet mellom verk og person blir derfor *har skapt/er skapt av*. Tar vi for oss relasjonen *er realisert ved* mellom verk og uttrykk, uttrykkes den andre delen av relasjonsparet som *er en realisering av*. Ved å følge denne relasjonen finner man verket, og fra dette kan man finne alle uttrykk av verket ved å studere alle verkets *er realisert ved*-relasjoner. Slik skapes indirekte relasjoner mellom alle uttrykk av et verk, og tilsvarende mellom alle manifestasjoner av et uttrykk og mellom alle eksemplarer av en manifestasjon.

Den andre klassen relasjoner beskriver relasjoner mellom forekomster av entiteter av samme type eller mellom entiteter av forskjellig type. Dette dreier seg for eksempel om relasjoner basert på opplysninger gitt i en entitet (opplag, utgave, sammendrag av, bind 2) eller basert på en analyse av entiteten. I FRBR er hver relasjon gitt et navn sammen med et relasjonspar.

Det er imidlertid viktig at slike relasjoner blir kodet (internt), og ikke språklig uttrykt slik de i mange tilfeller blir i følge katalogiseringsreglene i dag. Samtidig er det viktig som FRBR påpeker at relasjoner bare kan uttrykkes mellom entiteter, ikke ut i lufta (f.eks., ikke *basert på et skuespill av H. Ibsen*, men *basert på 'Et dukkehjem' av H. Ibsen*).

Som eksempel på slike andre nivås relasjoner gjengis relasjoner fra *verk* til *verk* i figur 3.7.2. I tillegg kan man altså også uttrykke emnerelasjoner mellom verk.

FRBR definerer også relasjonspar fra *uttrykk* til *uttrykk*, fra *verk* til *uttrykk*, fra *manifestasjon* til *manifestasjon*, fra *manifestasjon* til *eksemplar*, og fra *eksemplar* til *eksemplar*.

Sammensatte entiteter

Spesiell vekt blir det lagt på helhet/del-relasjonen mellom de respektive produktentitetene på hvert nivå.

Et viktig aspekt ved modellen er muligheten til å knytte sammen enkeltverk i større enheter gjennom disse helhet/del-relasjonene og likevel beholde øvrige relasjoner og egenskaper som selvstendige komponenter både for helheten og for de enkelte delene.

På verknivå i modellen kan for eksempel et antologiverk[9] settes sammen av de enkelte verkbidrag med sine egne relasjoner og egenskaper.

På manifestasjonsnivået koples flere fysiske enheter sammen ved helhet/del-relasjonen og det samme gjelder på eksemplarnivå. Enkeltverk i en monografiserie koples sammen i en serie, og enkeltartikler kan koples sammen til et tidsskrifthefte som igjen kan koples til en årgang (volum).

Det kan være forskjellige ansvarlige for å lage sammensatte entiteter. Det kan være produsenten som for eksempel knytter sammen forskjellige verk i en antologi, eller det kan være biblioteket som binder sammen flere hefter av et tidsskrift i ett bind.

3.7.3 Modellen

Bibliografiske poster

FRBR bruker gjennomgående uttrykket *bibliografisk post* om en bibliografisk beskrivelse. Det kan være vanskelig ut fra modellen å se hvor en bibliografisk post begynner og hvor den slutter. Modellen definerer en struktur der all informasjon i modellen henger sammen i sterkere eller svakere grad.

Dette kan være et problem når man skal utveksle bibliografiske data. Hva skal man utveksle? En rimelig antakelse vil være å ta utgangspunkt i den katalogiserte manifestasjon, og generere en post ut av egenskapene til de entitetene manifestasjon er forbundet med gjennom de grunnleggende relasjonene (på nivåene over) og med ansvars- og emneentitetene som hører til disse. Skal man utveksle til en samkatalogbase kan det også være aktuelt å ta med opplysninger om de ulike eksemplarene.

Grunnen til at modellen ikke definerer *bibliografisk post* nærmere kan være at eksisterende katalogiseringsregler og formater har ligget som en ubevisst forutsetning, og at disse er tenkt tilpasset slik at modellen kan inngå. Dette illustreres for så vidt gjennom rapportens appendiks som viser hvordan modellens egenskaper har paralleller til for eksempel UNIMARC. Implementeringen av modellen kan dermed sees på som en ren utvidelse av eksisterende systemer.

Det er viktig ikke å være bundet til en slik tankegang. Det kan nemlig føre til mye overflødig registrering av informasjon som ifølge modellen skal være normalisert gjennom eksistensen av entitetene. Et verks egenskaper må bare

[9] samling av enkeltverk, der utvalg og sammensetning representerer en intellektuell innsats som gjør det til et nytt verk.

registreres én gang, registrering av nye uttrykk av verket og nye manifestasjoner av uttrykkene må utnytte dette gjennom de grunnleggende relasjonene, det må være nok å peke på hvilket verk uttrykket er en realisering av for at uttrykket skal arve verkets egenskaper.

Mangler og problemer

I FRBR sies det eksplisitt at det fins områder der modellen langt fra er uttømmende og som må gjøres til gjenstand for videre studier og analyse. Eksempler på slike områder er en mer uttømmende kartlegging av egenskaper til forskjellige materialtyper, og ikke minst sies det at den dynamiske naturen til entiteter i digital form krever nærmere analyse.

Selve katalogiseringen kan være problematisk i denne modellen fordi den alltid vil være "baklengs". Utgangspunktet for enhver katalogisering vil alltid være et fysisk eksemplar. Å arbeide seg fra eksemplaret til det verket som ligger til grunn er ikke alltid like enkelt, når det kan dreie seg om omarbeidelser, filmatiseringer, lydbøker osv.

Åpenhet

FRBRs intensjon er å fungere som grunnlag for felles forståelse og videre diskusjon. Den gir seg ikke ut for å være siste ord i saken. Metodikken i modellen er i seg selv også åpen for å innføre nye entiteter, egenskaper og relasjoner.

Et eksempel på en slik åpenhet er muligheten for å kople litteratur om museumsgjenstander til museenes beskrivelser av gjenstandene gjennom entiteten *objekt*.

Åpenheten kan utnyttes til å kople bibliografiske systemer til læringsmiljøer der også andre tjenester og systemer inngår.

Modellens mål

Etter å ha gått gjennom de tre gruppene av modellens byggeklosser – entiteter, egenskaper og relasjoner – kan vi se på målformuleringen én gang til i fullstendig form:

- **å finne** de entitetene som korresponderer med brukerens uttrykte søkekriterier (det vil si å lokalisere enten en enkelt entitet eller en mengde entiteter i en fil eller database som et resultat av å bruke en entitets egenskaper eller relasjoner som søkekriterium).

- **å identifisere** en bestemt entitet (det vil si å verifisere at entiteten som beskrives korresponderer med den entiteten man søker, og å kunne skille mellom to eller flere entiter med liknende karakteristika).

- **å velge** en entitet som enten samsvarer med brukerens ønsker (det vil si å velge en entitet som tilfredsstiller brukerens krav med hensyn til innhold, fysisk format) eller som må forkastes som utilfredsstillende.

- **å få tak i** eller **få tilgang til** entiteten som er beskrevet (det vil si enten gjennom kjøp, lån, eller få tilgang til entiteten elektronisk ved en nettverkstilkopling til en maskin).

Sammenliknet med Cutters og Paris-prinsippenes mer konkrete mål, kan det se teoretisk ut, men går vi nærmere inn på målene så knytter for eksempel det **å finne** seg til *verk, uttrykk, manifestasjon* og *eksemplar*, slik målene ble formulert først i dette kapitlet på en mer konkret form.

Det er viktig nå å se hvordan entitetene er knyttet til det som ventes å være av kjerneinteresse for brukerne, for eksempel å finne alle verk om et bestemt emne, alle verk av en bestemt forfatter/korporasjon; å kunne skille mellom forskjellige uttrykk og manifestasjoner av et verk.

Katalogiseringsregler og edb-systemer

Tillett [166] sier i ovennevnte artikkel:

> This study is focused on bibliographic relationships and provides a taxonomy of relationship types as a consideration for future designers of cataloging rules and computerized systems for cataloging and catalogs. In order to develop future systems, we should first have a firm understanding of the theoretical framework upon which catalogs are built.

Det vil føre for langt å foreta en grundig studie av forholdet mellom katalogiseringsreglenes (AACRII) og modellens rammeverk. FRBR-rapporten har et appendiks som viser hvordan ulike egenskaper allerede har sitt uttrykk i internasjonale standarder (som ISBD, GARE, GSARE, UNIMARC[10]). Her skal bare følgende poenger nevnes:

- Skillet mellom verk, uttrykk, manifestasjon og eksemplar er uttrykt i dagens katalogiseringsregler, men siden reglene retter seg inn på beskrivelse av manifestasjoner, gjentas verk- og uttrykksegenskaper for hver enkelt manifestasjon. I FRBR-modellen vil dette kunne unngås.

- Det fins katalogiseringsregler for de fleste egenskapene i modellen.

- Mange av modellens relasjoner er også kjent fra katalogiseringsreglene, men har ofte en språklig formulering som er vanskelig å utnytte i edb-systemer ("Medforf", "Red", eller det er gitt en formulering i notefelt:

[10] **ISBD** = International Standard Bibliographic Description; **GARE** = Guidelines for Authority and Reference Entries; **GSARE** = Guidelines for Subject Authority and Reference Entries; **UNIMARC** = Universal machine readable cataloguing.

"Faksimileutg. Originalutg. Bergen, 1774"). Dette vanskeliggjør også full utnytting ved utveksling over språkgrenser. På samme måte som man er (noenlunde) enige om hvordan man koder bibliografiske egenskaper (245$a = hovedtittel), må man også bli enige om et kodesystem for relasjoner. Noe av dette er allerede gjort gjennom relasjons- og lenkefeltene i Normarc-formatet. (760-79X). Men fortsatt blir mange relasjoner uttrykt språklig i notefelt (5XX).

- Modellen har relasjoner som ikke er dekket av katalogiseringsreglene.

På bakgrunn av dette er det klart at det gjenstår betydelige studier og grunnlagsarbeid når det gjelder forholdet mellom modellen og katalogiseringsreglene for å fastslå hvor endringer må gjøres. Dette arbeidet er i gang i IFLA-regi, se http://www.ifla.org/VII/s13/frbr/isbd-chg.htm, der det inviteres til å komme med endringsforslag til ISBD(M) for å bringe det i samsvar med FRBRs grunnleggende krav til nasjonalbibliografiske poster.

Også edb-systemene må endres betraktelig for å kunne utnytte nye katalogiseringsregler bygget på FRBR. Dette gjelder ikke minst databasemodellen, men det gjelder også for systemenes virkemåte i katalogiseringsfasen og i grensesnittet for sluttbrukerne av katalogen.

Dersom modellen blir implementert i katalogiseringsregler og edb-systemer, vil det åpne for mye bedre navigasjonsmuligheter i det bibliografiske univers. En underkomite av IFLAs katalogseksjon har publisert et høringsutkast til retningslinjer for å organisere bibliografiske beskrivelser for katalogbrukeren [188]. Retningslinjene er basert på dagens katalogiseringsregler, men beskriver likevel utmerket hvor mye bibliografisk informasjon som ligger uutnyttet i katalogene, men dette kan være vanskelig å implementere med dagens data. Dersom FRBR tas i bruk, vil man kunne nå mye lenger, noe vi har forsøkt å illustrere i tillegg D.

Kapittel 4

Digital informasjon

I løpet av vår historie har en rekke teknologiske nyvinninger gitt oss muligheten til å formidle informasjon på nye måter, f.eks. boktrykkerkunsten, fotografiet, levende bilder, radio, fjernsyn og mange, mange flere. Digital informasjon formidlet over nett er i så måte ingen revolusjon, men bare et nytt trinn i denne utviklingen. Et generelt problem for digital informasjon er mangfoldet av løsninger for hvordan digital informasjon skal lagres, presenteres på skjerm, formidles over nett, etc. Dette skyldes både den raske utviklingen av informasjonsteknologi, de mange leverandøravhengige plattformene som finnes for informasjonsbehandling, og de mange formål informasjonen og systemene skal tjene.

I dette kapitlet beskrives noen utvalgte modeller og løsninger for digital informasjon som er relevant for digitale bibliotek.

4.1 CNRI-arkitekturen

En av de mer refererte artikler om rammeverk eller arkitektur for digitale bibliotek, er Kahn og Wilenskys *A Framework for Distributed Digital Object Services* [108]. I dette rammeverket fokuseres det på digital informasjon, og de viktigste begrepene som blir innført, er *digitalt objekt* (digital object), *hendel* (handle) og *lager* (repository). En videre bearbeiding av disse begrepene er gitt av William Arms i [11], og er utgangspunkt for *CNRI Digital Object Architecture Project*[1] [12]. Ideen om en hendel som et unikt, lokaliseringsuavhengig navn, er videreutviklet til CNRIs *the Handle System*[2].

Vi presenterer her et sammendrag av disse artiklene hvor vi fokuserer på de mest sentrale delene relatert til representasjon av digital informasjon. Selve modellen og de begrepene som introduseres, er noe forskjellig beskrevet i disse artiklene, da dette er mer presentasjoner av ideer under utvikling enn formelle spesifikasjoner. Vi har derfor tillatt oss å forenkle framstillingen av enkelte begreper.

[1] se: http://www.cnri.reston.va.us/doa.html

[2] se kap. 5.6.6 og http://www.handle.net

4.1.1 Fokus for arkitekturen

- **Relasjoner**
  Digitalt materiale er ofte relatert til andre enheter av digitalt materiale. Dette kan være strukturelle relasjoner som del-av/helhet (f.eks. består de fleste web-sider av html-koden i én fil og illustrasjoner og grafikk i egne filer), eller former for relasjoner som samlinger basert på likhet og relevans mellom informasjonsobjekter.

- **Formater**
  Digital informasjon kan finnes i mange formater. Ett og samme bilde kan lagres i mange forskjellige filformater. Enkelte av disse representerer nøyaktig samme informasjon, slik at det er mulig å konvertere mellom formatene, mens andre formater vil gi forskjellig representasjon av samme informasjon (f.eks. enkelte former for komprimering).

- **Versjoner**
  Digital informasjon er enkel å endre, og derfor vil det lett oppstå forskjellige versjoner av digitale informasjonsobjekter. Dette er ikke vesentlig forskjellig fra f.eks. forskjellige utgaver av en bok, men det spesifikke for digital informasjon er at versjonsproblematikken er mye mer fremtredende.

- **Rettigheter**
  De forskjellige enhetene av digital informasjon kan ha forskjellige bruks- og adgangsrettigheter knyttet til seg.

- **Tjenestekvalitet**
  Brukere har forskjellige behov for hvordan de vil ha tilgang til informasjon. Brukere som aksesserer informasjon fra et digitalt bibliotek over et høyhastighetsnett, vil ha andre behov enn de som benytter modem og har vesentlig mindre overføringskapasitet. Krav til kvalitet på informasjon kan derfor være forskjellig fra bruker til bruker.

4.1.2 Arkitekturens elementer

De forskjellige elementene som rammeverket består av er:

- **Digitale objekter**
  Den fundamentale enheten i arkitekturen kalles et digitalt objekt, og dette er en datastruktur bestående av nøkkelmetadata og digitalt materiale.

- **Nøkkelmetadata**
  Nøkkelmetadata er den informasjon som er nødvendig for å administrere det digitale objekt i et nettverksmiljø. Dette er først og fremst en identifikator (hendel), men kan også være informasjon om bruks- og adgangsrettigheter assosiert til det digitale objektet.

- **Digitalt materiale**
  Det digitale materialet er selve innholdet i et digitalt objekt, og har en struktur som kan bestå av elementer og pakker.

- **Element**
  Et element er en innkapsling av et elementært innhold. Elementene har en ID, attributter og et dataelement:

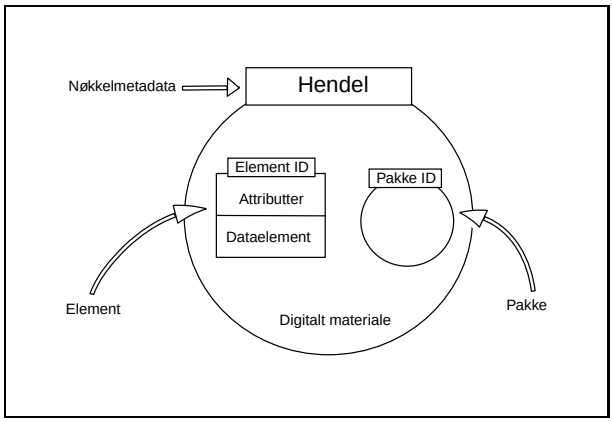
  - ID er en lokal og systemrelatert identifikator for elementet.
  - Attributtene er informasjon som er nødvendig for å kunne prosessere elementet. Dette inkluderer både en rollebeskrivelse og en typebetegnelse. Rollen sier hvilken funksjon dette elementet har i det digitale objektet, om det er metadata eller om det er innholds-data. Type gir informasjon om format, f.eks. at dette er ASCII-tekst, en Word-fil eller et JPEG-bilde.
  - Dataelementet er bit-sekvensen som utgjør innholdet, f.eks. et tekstdokument eller et bilde.

- **Pakker**
  Pakker brukes for å gruppere elementer og andre pakker. En pakke har en ID tilsvarende som for et element. Et digitalt objekt er også en pakke, men med den forskjellen at ID er en hendel.

- **Hendel**
  En hendel er et globalt unikt og varig navn som kan oversettes til lokaliseringsinformasjon, som adresse og protokoll.

- **Metaobjekter**
  Elementene i et digital objekt kan inneholde hendler som dataelement. Det digitale objektet blir da bare en aggregert enhet for referanser til andre digitale objekter, og kan slik brukes for å organisere relasjoner mellom digitale objekter. Attributtene i elementet kan benyttes for å beskrive hvilken rolle de forskjellige digitale objektene har i den aggregerte enheten (strukturelle metadata).

- **Lager**
  Informasjon som skal være tilgjengelig i et digitalt bibliotek, kan lagres og gjøres tilgjengelig på mange måter, f.eks. ved å bruke en web-tjener eller en ftp-tjener. I arkitekturen benyttes lager (repository) for et system for lagring av digitale objekter. Intensjonen med et lager er å utvikle et system for lagring og tilgang til informasjon hvor det kan taes spesielle hensyn til sikkerhet, og hvor egenskaper (metadata) og transaksjonslogger kan assosieres til de enkelte digitale objekter. Et lager har funksjoner for å lagre digitale objekter (deposit) og aksessere digitale objekter (aksess) via en Repository Access Protocol (RAP), men kan også kombineres med andre typer informasjon, tjenester og administrasjon.



Figur 4.1: Et digitalt objekt

4.2 Hypertekst

4.2.1 Om hypertekst

Hypertekst er et uttrykk som ble introdusert av Ted Nelson i 1965 for å beskrive et system for ikke-lineær tekst[3] [136]. I dataverdenen er hypertekst en metafor for informasjon presentert som et nettverk av noder hvor leseren er fri til å navigere mellom assosierte noder. Hypermedia er mer eller mindre synonymt med hypertekst, men i dette uttrykket legges det vekt på at informasjonen ikke bare er tekst, men også bilder, lyd, video o.l.

Kombinasjonen av HTML og HTTP er grunnlaget for hypertekst-informasjonsrommet World Wide Web, men det er også utviklet en rekke andre hypertekstsystemer. Et hypertekstsystem kan enten være lukket og kun brukt på en avgrenset informasjonsmengde, men det kan også være et globalt og åpent informasjonsrom hvor lenkene brukes for å koble sammen informasjon som er distribuert på mange maskiner i et nettverk, slik som for World Wide Web. Et viktig element i alle hypertekstsystemer er lenking og adressering.

Hypertekst (og hypermedia) er viktige elementer for digitale bibliotek bl.a. fordi:

- Denne måten å organisere informasjon på støtter andre typer informasjonsgjenfinning enn tradisjonell søking. Bruk av lenker for å organisere informasjon gir mulighet for assosiativ informasjonsoppdagelse, noe som er en velegnet strategi, f.eks. når en bruker ikke har et klart definert informasjonsbehov, men snarere er på jakt etter mulig relevant informasjon.

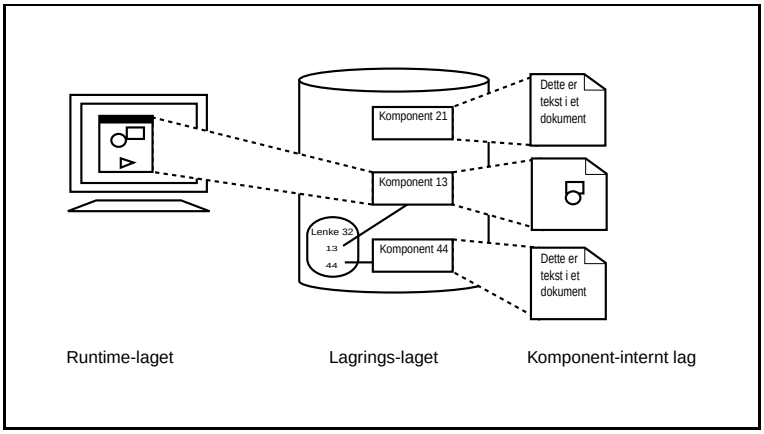
[3] I historiske omtaler av hypertekst er det også vanlig å henvise til Vannevar Bush og hans artikkel "As We May Think" fra 1945. Her beskrives et tenkt system – memex – som er en innretning for assosiativ lenking mellom dokumenter på mikrofiche [27].

- Hypertekst er et velegnet medium for formidling av informasjon f.eks. i en læringssituasjon. Dette gjelder både som overordnete informasjonsstrukturer for en mengde av dokumenter, og som presentasjonsmåte for enkeltdokumenter.

- Hypertekst kan også sees på som et brukergrensesnitt, ved at lenkene representerer overganger fra skjermbilde til skjermbilde, som i andre sammenhenger ivaretas av dialoger og menyvalg.

4.2.2 Dexter-modellen

Dexter-modellen – The Dexter Hypertext Reference Model – er et forsøk på å fange opp de viktigste abstraksjonene som finnes i hypertekstsystemer [69, 70]. Den ble utarbeidet gjennom en serie av workshops hvorav den første ble holdt i 1988[4]. Selv om modellen er relativt gammel og ble utarbeidet før World Wide Web, er den et viktig teoretisk grunnlag for forståelse av hypertekstsystemer og ofte referert til.



Figur 4.2: De tre lagene i Dexter-modellen

- **Tre lag**
  Dexter-modellen deler et hypertekstsystem i tre lag, *runtime-laget* (runtime layer), *lagrings-laget* (storage layer) og det *komponent-interne laget* (within-component layer). Hovedfokus for modellen er på lagrings-laget, som er en modell av den nettverksstrukturen av noder og lenker som er essensen av hypertekst. Lagrings-laget beskriver en konseptuell "database" bestående av et hierarki av *komponenter* som inneholder data[5].

[4] Referansemodellens navn er tatt fra The Dexter Inn i New Hampshire, som huset workshopen hvor dette initiativet startet.

[5] Med komponenter i Dexter-modellen menes beholdere for data. Dette er ikke samme bruk av begrepet komponent som vi benytter i kap. 8, og som er bundet sammen med lenker.

- **Komponenter**
  De fundamentale entiteter og adresserbare enheter i lagrings-laget er *komponenter*. Komponentene inneholder tekst, bilder, video eller annet som utgjør innholdet i hypertekstsystemet, og en komponent er å betrakte som en beholder for disse dataene. Dexter-modellen gjør ingen forsøk på å modellere innholdet i en slik beholder, og det komponent-interne laget er å betrakte som utenfor modellen. Hvis et bilde eller en tekstfil er datainnholdet i en komponent, er med andre ord dokumentets interne organisering ikke relevant i Dexter-modellen. En komponent er enten *atomisk*, en *lenke*, eller en *sammensatt komponent*.

  En atomisk komponent er den elementære byggeklossen i modellen. Den kan ha en intern struktur, men denne er komponent-intern og et anliggende for det komponent-interne laget. Atomiske komponenter er det som ofte kalles noder i hypertekstsystemer.

  Lenker er entiteter som representerer relasjoner mellom andre komponenter. De er i hovedsak en sekvens av to eller flere endepunkter (adresser) som hver refererer til en komponent eller del av en komponent i hyperteksten.

  Sammensatte komponenter består av andre komponenter, og kan utgjøre et hierarki begrenset til en "Directed Acyclic Graph" (DAG). I dette ligger det at en komponent kan inngå som subkomponent av andre komponenter, men en komponent kan ikke inneholde seg selv hverken direkte eller indirekte.

- **Komponentbeskrivelsen**
  Alle disse formene for komponenter er egentlig komplekse enheter ved at de er satt sammen av to deler, en *komponent-beskrivelse* og et *innhold*. Innholdet er selve dataene som utgjør komponentens innhold, og komponentbeskrivelsen er egenskaper til komponenter andre enn dens innhold. Dette kan for eksempel være informasjon om hvordan innholdet i komponenten skal presenteres i brukergrensesnittet, men det kan også være metadata som beskriver andre aspekter ved innholdet, som emneord og opphav. Innholdsbeskrivelsen kan også inneholde en typebeskrivelse som f.eks. forteller om dette er tekst, eller hvilket bildeformat innholdsdataene er lagret som.

- **Unike identifikatorer**
  Hver komponent har en globalt unik identitet (unique identifier - UID). UID'er er basis byggeklosser i modellen, og skal være unike i hele det systemet som hyperteksten berører.

- **Lenking**
  Lenking i Dexter-modellen er basert på spesifisering av to eller flere endepunkter. Et endepunkt kalles et anker (anchor), og er en form for in-

direkte adressering inn i en komponent. Et anker består av en anker-ID som identifiserer ankeret i komponenten, og en anker-verdi som brukes for å angi lokasjon, region, enhet eller substruktur en komponent. Anker-verdien er del av komponenten som inngår i lenken, og relatert til det komponent-interne laget. Anker-verdien vil kun bli behandlet f.eks. av det program som skal presentere komponenten i hypertekstsystemet. Et anker kan unikt identifiseres i hele hypertekstsystemet ved å kombinere anker-ID med UID for komponenten.

For å angi en lenke bygger man opp en komponent som består av en eller flere spesifikatorer (specifier). En spesifikator er en datastruktur som består av komponent-spesifikasjon (UID), en anker-ID, retning (direction), og presentasjonsspesifikasjon (presentation specification). Retning er informasjon om ankerets rolle i relasjonen (kan ha verdiene FROM, TO, BIDIRECT og NONE), og presentasjonsspesifikasjonen er en verdi som er del av grensesnittet mellom lagrings-laget og runtime-laget.

Modellen beskriver også en rekke funksjonelle aspekter ved hypertekstsystemer. For enkelthets skyld har vi utelatt dette aspektet og henviser til de originale artiklene som beskriver Dexter-modellen for de som evt. har interesse av å vite mer om dette.

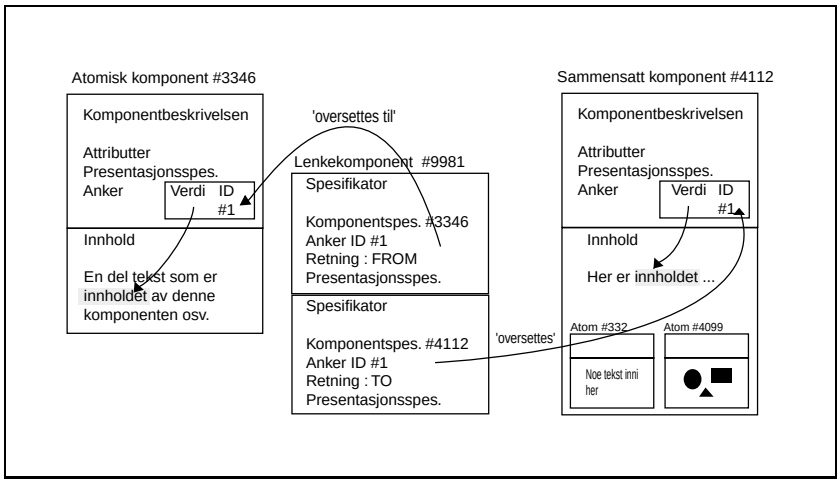
Dexter-modellen er kun en referansemodell for et hypertekstsystem basert på de erfaringer og krav man kjente til på 1980-tallet. Ideene fra Dexter-modellen er i løpet av 90-tallet videreutviklet i mange forskjellige prosjekter. Amsterdam-modellen [72] er en videreutvikling av Dexter-modellen med en bedre tilpasning til multimedia, og Grønbæk og Rigg beskriver i [67] en Dexter-basert modell for integrering av interne lenker slik vi finner de i HTML og lenkeobjekter, og vi finner Dexter-modellen som et av grunnlagene for utviklingen av både Resource Description Framework og XML Links (se kap. 4.3.3).

På WWW er HTML det rådende hypertekstspråket. HTML og lenkingsmulighetene vi har i dette hypertekst-formatet skiller seg fra Dexter-modellen ved at lenkene er en del av teksten og ikke i separate komponenter. Bruken av HTTP-protokollen og web-lesere avviker også noe fra Dexter-modellens beskrivelse av et runtime-lag, selv om det er mulig å finne likhetstrekk.

Det viktigste vi kan trekke fra Dexter-modellen er fremstillingen av komponenter, lenker og sammensatte komponenter som er en grunn-ide, og denne delen av modellen har klare likhetstrekk med CNRIs arkitektur for digitale objekter. Lenker som selvstendige objekter er noe vi finner igjen i en rekke andre løsninger og systemer, f.eks. Xlink og HyTime, og enkeltsystemer bl.a. for referanselenking mellom artikler.

4.2.3 HTML – Hypertext Markup Language

HTML er en SGML-applikasjon for hypertekstdokumenter på World Wide Web [176]. HTML-dokumenter er portable og kan benyttes på alle plattfor-



Figur 4.3: Komponenter og lenking i Dexter-modellen

mer. Det er utviklet web-lesere for de fleste maskintyper og operativsystemer, noe som gjør at HTML er blitt et universelt dokumentformat for utveksling av informasjon.

I HTML benyttes tagger eller merker, som for å formatere teksten i overskrifter og avsnitt, men HTML inneholder også merker for å spesifisere hypertekstlenker, inkludere bilder, lage skjema for inndata fra bruker, m.m. Ved hjelp av HTML kan vi:

- lage dokumenter med overskrifter, tekst, tabeller, lister.
- gjøre informasjonen interaktiv ved å inkludere klikkbare hypertekstlenker i dokumentene.
- lage skjema (ved hjelp av Forms) for inndata til tjenester som informasjonssøking og bestillinger.
- inkludere bilder, video, lyd, regneark direkte i dokumentene.

HTML er opprinnelig utviklet av Tim Berners-Lee, og ble populær blant annet på grunn av programmet Mosaic[6] som var den første grafiske web-leseren som ble brukt av et stort publikum. Spesifikasjonene for HTML utvikles av W3C, men det er også under utvikling en ISO-standard for HTML [103].

Selv om HTML og hypertekstprotokollen HTTP er komplementære, er dette i realiteten to teknologier som også kan benyttes uavhengig av hverandre. Det er f.eks. blitt vanlig å implementere hjelpe-systemer og presentasjoner i HTML, hvor filene er lagret lokalt og derfor kun hentes fra filsystemet.

[6] se http://www.ncsa.uiuc.edu/SDG/Software/Mosaic/NCSAMosaicHome.html

4.2.4 HyTime

HyTime (Hypermedia/Time-based Structuring Language) [96] er en ISO-standard for hypertekst og hypermedia. HyTime er en SGML-applikasjon som gjør det mulig å representere statisk og dynamisk informasjon for prosessering og utveksling i hypertekst- og multimedia-applikasjoner, ved hjelp av standardiserte måter for å spesifisere hyperlenker i og mellom dokumenter og andre informasjonsobjekter, og for å planlegge multimediainformasjon i tid og rom.

Selv om det er store fordeler knyttet til å basere et hypertekstsystem på en åpen og standardisert spesifikasjon, har ikke HyTime foreløpig oppnådd noen global suksess. Standarden er kompleks og definerer kun elementer og attributter og deres virkemåte. Det er foreløpig utviklet få HyTime-baserte løsninger, og det finnes lite programvare tilgjengelig som kan forenkle bruken av HyTime. Introduksjonen av XML og spesifikasjonene for lenking og adressering i XML, er løsninger som kan ivareta noen av de samme behovene som HyTime adresserer, og det er antagelig rimelig å anta at de XML-baserte løsningene som utvikles i regi av W3C vil dominere i fremtiden.

4.3 Markup-språk

Ideen om at strukturerte dokumenter kunne utveksles og manipuleres med, hvis de ble publisert i et åpent standardisert format, går helt tilbake til 1960-tallet. IBM utviklet GML (the Generalized Markup Language) for å håndtere sine publiseringsbehov, og dette var en av forløperne til SGML – Standard Generalized Markup Language – som ble en ISO-standard i 1986 [87].

Markup er alt i et dokument som ikke er innhold [10]. Den tradisjonelle betydningen av markup er den manuelle merkingen av teksten som ble gjort på manuskripter før de skulle typesettes for trykking, for å beskrive hvordan teksten skulle plasseres på en side, hvilken skrifttype og størrelse som skulle brukes, etc. Denne typen markup kalles for *prosedural markup* fordi den beskriver hvordan teksten skal se ut. *Generisk markup* beskriver derimot hensikten med teksten som merkes, i stedet for hvordan teksten fysisk skal se ut. Generisk markup kalles også deskriptiv markup.

Et sentralt konsept for generisk markup er at innhold skal separeres fra stil. Generisk markup identifiserer elementene i en struktur, som kapitler, avsnitt, innholdsfortegnelse osv.

SGML bringer generisk markup enda videre ved å spesifisere et språk for å sette opp hierarkiske dokumentmodeller, hvor elementene i dokumentet inngår i en logisk predefinert struktur. SGML er både en syntaks for markup av dokumentelementene og et rammeverk for markup av dokumenter. Ved hjelp av SGML kan dokumenter utvikles uavhengig av maskin- og programvare, og SGML kan gi opphav til et uendelig antall av dokumentstrukturer.

Et SGML-dokument kan brytes opp i tre komponenter; *struktur*, *innhold* og *stil*. Struktur beskrives ved hjelp av en DTD (dokumenttype-definisjon) som

er en definisjon av de elementene og elementstrukturene som dokumentet kan bestå av. Innhold er informasjonen i dokumentene, f.eks. selve teksten i en bok. Stil er en spesifikasjon av hvordan et dokument skal presenteres. Dette er ikke en del av SGML-standarden, men defineres via egne språk for å spesifisere presentasjon av SGML-dokumenter, f.eks. DSSSL [95].

SGML har vært opphavet til mange markup-språk, eller SGML-applikasjoner som de ofte kalles. I dette kapitlet ser vi spesielt på XML, mens HTML og HyTime er omtalt i relasjon til hypertekst i kap. 4.2. Både HTML og HyTime er SGML-applikasjoner, mens XML en forenklet utgave av SGML som igjen gir opphav til andre XML-applikasjoner.

Utgangspunktet for SGML var dokumenter og publisering, men vi ser markup også gir mulighet for å representere datastrukturer i dokumentene. Dette er et aspekt som gjør at SGML-baserte dokumenter ikke bare brukes til å spesifisere dokumentelementer som avsnitt og overskrifter, men vi kan bruke markup for å representere datastrukturer som er nødvendige for spesifikk funksjonalitet, f.eks. lenker i hypertekstdokumenter.

4.3.1 XML – Extensible Markup Language

XML – Extensible Markup Language – kan beskrives som en forenklet utgave av SGML. Formelt sett er XML et subsett av SGML hvor komplekse og lite brukte egenskaper ved SGML er tatt ut. Det sies om XML at det inneholder 90% av SGMLs funksjonalitet og 10% av dets kompleksitet. XML ble en W3C-anbefaling i 1998 [174], og har i løpet av kort tid oppnådd status som det fremtidig universelle språket for strukturerte dokumenter og data på World Wide Web.

Motivasjonen for å utvikle XML var behovet for noe midt imellom HTML og SGML. Kompleksiteten i SGML gjør det vanskelig både å lære språket og implementere løsninger som bruker SGML, mens HTML er for statisk og avgrenset til å kunne brukes på alle de områdene hvor det er behov for standardiserte måter for å formidle og utveksle strukturerte dokumenter og strukturert data.

Det er generelt tre områder hvor XML gir funksjonalitet vi ikke finner i HTML [19]:

- **Utvidbart.** Muligheten for å definere nye elementer og attributter.
- **Struktur.** Muligheten for "dype strukturer", nødvendig f.eks. for å kunne representere databaseskjemaer eller objektorienterte hierarkier.
- **Validering.** Muligheten for å kontrollere at strukturen er i overenstemmelse med gitte regler for hvilke elementer som kan inngå, og forholdet mellom disse m.m.

4.3.2 XML-dokumenter

Et tekstobjekt som følger XML-spesifikasjonene, er et *XML-dokument*. Et XML-dokument har både en logisk og en fysisk struktur:

- Fysisk sett består et XML-dokument av entiteter. En entitet kan referere til andre eksterne entiteter og dermed gjøre at disse inngår som del av XML-dokumentet. Et XML-dokument har bestandig en toppnivå-entitet – en rot. XML-dokumentet kan med andre ord være satt sammen av mange filer, som tilsammen utgjør XML-dokumentet.
- Logisk sett består et dokument av deklarasjoner, elementer, kommentarer, tegnreferanser og prosesseringsinstruksjoner, som alle er uttrykt gjennom eksplisitt markup.

Et XML-dokument må være *velformulert* (well-formed). Et velformulert XML-dokument kan i tillegg være *gyldig* (valid) hvis det er i overenstemmelse med en dokumenttype-definisjon (Document Type Definition – DTD).

- **Velformulerte XML-dokumenter:**
  - Hele XML-dokumentet følger XML-syntaksen. Dette gjelder også for entiteter det indirekte eller direkte refereres til.
  - Har ett eller flere elementer.
  - Har bare ett rot-element (dokument-elementet) som heller ikke inngår som innhold i andre elementer i dokumentet.
- **Gyldige XML-dokumenter** er velformulerte, men må i tillegg overholde en assosiert DTD som spesifiserer:
  - Hvilke elementer som kan benyttes i dokumentet og hvilket innhold disse skal ha, hvordan disse skal være strukturert, f.eks. hvilke elementer som er subelementer av andre, o.l.
  - Elementenes attributter og regler for disse: hvilke verdier de kan ha, om de er obligatoriske, o.l.
  - Entiteter og notasjoner (notations) som kan inngå i dokumentet.

Selv om XML er et subsett av SGML har det likevel en viktig egenskap som SGML mangler, nemlig suksess og spesiell innretning mot World Wide Web. Lanseringen av XML skjedde på et tidspunkt da mange hadde erfart begrensningene i HTML og var på utkikk etter bedre alternativer, noe som skapte store forventninger til XML. De forenklinger som XML gir i forhold til SGML, er også et vesentlig bidrag til XMLs suksess. Disse forenklingene har gjort det overkommelig både å lære og å bruke XML. Tilgjengeligheten av mange frie programmer for XML har også akselerert bruken av XML, i tillegg til de mange spesielle innretningene for XML som er under utvikling.

4.3.3 XML-baserte løsninger

I tillegg til basis-spesifikasjonene for XML utvikles det i regi av W3C også en rekke spesifikasjoner for bruk av XML, både til generelle og mer spesifikke behov:

XML Namespaces
Et XML-dokument kan bruke elementer og attributter som er assosiert med forskjellige DTD'er. For å forhindre overlappende vokabularer og for å presist kunne identifisere et element eller attributt, er det behov for å knytte navnene til et navnerom. Et XML-navnerom (XML namespace) er en samling av navn identifisert ved hjelp av en URI, som brukes i et XML-dokument som element- og attributtnavn [178].

XML Schema
Bruk av dokumenttype-definisjon – DTD – for å spesifisere gyldig struktur i XML-dokumenter, har klare begrensninger, i tillegg til at syntaks for DTD er forskjellig fra syntaks for resten av XML-dokumentet. "XML Schema" er en spesifikasjon under utvikling, hvor målet er å utvikle et språk for å spesifisere regler for struktur, dokumentinnhold m.m., basert på XML-syntaksen [173].

XML Query
XML Query er en felles overskrift for flere spesifikasjoner under utvikling. Målet er å utvikle et spørrespråk for XML, både for å søke i enkelte dokumenter og i samlinger av XML-dokumenter. Resultatet fra et søkeuttrykk skal kunne være både hele dokumenter, eller subdeler av dokumentene som tilfredstiller betingelser for struktur og innhold. Et søkeuttrykk vil også kunne resultere i nye dokumenter ut fra betingelsene i søkeuttrykket. I utviklingen av XML Query fokuseres det både på å støtte behovene for søking i forhold til "tradisjonelle" dokumenter, og på behovene i forhold til strukturerte data representert som XML, f.eks. datalager [172].

XLink – XML Linking Language
Denne spesifikasjonen definerer elementer som kan brukes i XML-dokumenter for å uttrykke og beskrive lenker mellom ressurser [185]. XML-syntaksen brukes for å uttrykke både enkle enveis lenker tilsvarende det som kan gjøres i HTML, men også mer avanserte lenker som toveis lenker separate fra ressursene det lenkes mellom (se eksempel i fig. 6.3).

XPath – XML Path Language
XPath er et språk for å adressere deler av et XML-dokument, utviklet for å brukes både av XSL og XPointer [180].

XPointer – XML Pointer Language
Denne spesifikasjonen definerer et språk for URI-basert adressering inn i den interne dokumentstrukturen til et XML-dokument. XPointer er basert på XPath, og støtter adressering basert på hierarkisk struktur, valg av indre part basert på elementtype, attributtverdi, innhold og relativ posisjon [186].

XSL – Extensible Stylesheet Language
XSL er et språk for presentasjonsinformasjon (stilsett) [182]. Det består av to deler: et språk for å transformere XML-dokumenter (XSLT) og et vokabular for å uttrykke formatering (presentasjon).

XSLT – XSL Transformations
XSLT er et språk for å spesifisere hvordan et XML-dokument skal transformeres til et annet XML-dokument [181]. XSLT er utviklet som en del av XSL, og selv om det også er utviklet for å kunne brukes uavhengig av XSL, er det ikke et generelt transformeringsspråk.

RDF – Resource Description Format
RDF er et rammeverk for maskin-forståelige metadata [179]. Med maskinforståelig menes at metadata skal kunne behandles automatisk av programvare. RDF er basert på en generell datamodell, men det er utviklet en XML-basert syntaks som implementerer modellen.

Grunnlaget for RDF er en modell hvor ressurser beskrives med navngitte egenskaper og en verdi. En verdi kan igjen være en ressurs og på denne måten støtter RDF rekursive beskrivelser av ressursene. RDF er uavhengig av metadataformat og modellen støtter integrering av metadataelementer fra forskjellige formater. I den XML-baserte syntaksen benyttes XML-navnerom for å relatere de forskjellige metadataelementer til sine respektive formater. RDF-modellen gir også mulighet for å aggregere ressurser, slik at en metadatapost kan beskrive alle sider på et web-sted o.l.

XHTML – Extensible HyperText Markup Language
XHTML er en reformulering av HTML 4 som en XML-applikasjon [184]. Fordelene ved å bruke XHTML i stedet for HTML er at XHTML-dokumenter er i overenstemmelse med XML-spesifikasjonen og derfor kan presenteres, redigeres og valideres med XML-verktøy. Samtidig vil XHTML-dokumenter fungere like godt i eksisterende web-lesere som støtter HTML 4.

MathML – Mathematical Markup Language
MathML er et XML-basert språk for matematiske uttrykk. Målet er å gi bedre støtte for matematikk på World Wide Web, både til formidling og for prosessering, tilsvarende det HTML har gjort mulig for tekst [177].

SMIL – Synchronized Multimedia
SMIL er et XML-basert språk for multimedia-presentasjoner på World Wide Web [175]. SMIL-presentasjoner kan derfor skrives ved hjelp av en vanlig tekstbehandler. I et SMIL-dokument er det mulig å:

- Integrere et sett av uavhengige multimediaobjekter til en synkronisert multimedia-presentasjon.
- Beskrive de tidsrelaterte egenskapene i presentasjonen.
- Bestemme layout for presentasjonen.
- Assosiere hyperlenker til mediaobjektene.

SVG – Scalable Vector Graphics
SVG er et språk for todimensjonal vektorgrafikk eller grafikk som kombinerer vektorgrafikk og rastergrafikk [183].

4.3.4 Stilsett

Det finnes flere standarder og spesifikasjoner for stilsett:

- DSSSL (Document Style Semantics and Specification Language) er en ISO-standard for å spesifisere formatering og transformering av SGML-dokumenter [95]. Formålet med DSSSL er primært å formatere SGML-dokumenter for papir eller elektroniske media-presentasjoner, og å transformere SGML-dokumenter mellom markup-skjemaer som er definert i forskjellige DTDer. Hoveddelene i DSSSL er:
  - Et språk og en prosesseringsmodell for å transformere SGML-dokumenter til andre SGML-dokumenter.
  - Et språk for å spesifisere hvordan SGML-dokumentet skal formateres.
  - Et spørrespråk for å identifisere deler av et SGML-dokument; SDQL (Standard Document Query Language).
- XSL (Extensible Stylesheet Language) er et stilsett-språk for XML som benyttes for å spesifisere presentasjon av XML-dokumenter (se s. 83).
- CSS (Cascading Style Sheets) er en spesifikasjon utviklet av W3C, opprinnelig med HTML som primært bruksområde, men CSS kan også benyttes på f.eks. XML [119]. Flere web-lesere støtter bruk av CSS nivå 1.

  For CSS nivå 1 er det mulig å definere presentasjonsregler for skrifttype og størrelse, farge, justering, linjeavstand o.l., boksegenskaper som innramming, margstørrelse og tekstflyt rundt rammer, samt andre egenskaper

som det å skjule tekst og spesifisere bakgrunn. CSS nivå 2 er siste versjon som har status som W3C anbefaling, og i denne spesifikasjonen utvides stilsettet med stilegenskaper for tekst-til-tale systemer, stilegenskaper for utskrift, langt mer avansert fonthåndtering, og enkel tekstgenerering. nivå 2 er til nå dårlig støttet av web-lesere. Neste generasjon av CSS (nivå 3) er under utvikling.

4.4 Medietyper

I digital lagring eller formidling av medier som tekst, bilder, lyd eller film, er disse representert som avgrensete sekvenser av binære data. Til dette formålet er det utviklet mange lagrings eller representasjonsformater, og for å kunne identifisere disse medietypene, formatene og den assosierte programvaren som kan presentere formatet, er det behov for en typebetegnelse. I enkelte operativsystemer benyttes filens etternavn som en indikasjon på hvilket program som er assosiert med filen, men også andre teknikker benyttes for å spesifisere dokumentformatet på andre plattformer.

På Internett brukes ofte *medietyper* som en typebetegnelse for informasjonen. Medietyper er definert i MIME-spesifikasjonene (Multipurpose Internet Mail Extensions), som opprinnelig er en protokoll for bruk av multimedia i epost. [58, 59]. MIME-protokollen anvendes også i mange andre sammenhenger, f.eks. HTTP, selv om denne protokollen ikke følger MIME-standarden fullt ut.

I MIME defineres en rekke meldingsoverskrifter, deriblant *Content-type* som benyttes for å spesifisere type data som følger i innholdet. Dette gjøres ved hjelp av toppnivå-medietype, en subtype, og eventuelle parametre som gir ytterligere informasjon om innholdet.

Toppnivå-medietype brukes for å spesifisere generell type som tekst eller bilde, mens subtype angir et spesifikt format. Dette gjør det mulig for et program, f.eks. en epost-leser eller web-leser, å vurdere presentasjon av innholdet, uten at det spesifikke formatet som er brukt behøver å være kjent. MIME-standarden spesifiserer de forskjellige toppnivå-medietypene som kan benyttes, og noen basis subtyper. Andre subtyper registreres ved IANA [60, 77]. Leverandørspesifikke formater har subtype-navn med prefiks "vnd".

Parametrene som kan benyttes, er relatert til medietype. Enkelte er spesifiserte for toppnivå-medietype og kan brukes for alle subtyper av denne, andre er kun assosiert med en spesifikk subtype.

MIME-standarden spesifiserer også en rekke regler for hvordan et program skal tolke innholdet hvis subtypen er ukjent o.l.

De forskjellige toppnivå-medietypene er :

- **Text**
  Medietypen "text" indikerer tekstlig informasjon. Subtypen "plain" indikerer at dette er helt ren tekst, uten noen formatering, som skal presen-

teres slik den er. Andre subtyper som benyttes er "enriched". For "text" er det også registrert en rekke subtyper som "html", "xml", "sgml". Parameteren "charset" kan benyttes for å indikere tegnsett.

```
Content-type: text/plain
Content-type: text/xml
```

- **Image**
  Medietypen "image" indikerer at innholdet er et bilde. Subtyper under "image" er f.eks. "jpeg" og "gif", som er vanlige formater for bilder på Internett.

  ```
  Content-type: image/gif
  Content-type: image/jpeg
  ```

- **Audio**
  Medietypen "audio" indikerer audio data som skal presenteres som lyd.

  ```
  Content-type: audio/basic
  ```

- **Video**
  Medietypen "video" indikerer levende bilder, evt. med lyd. Subtyper under denne er videoformater som "mpeg" og "quicktime".

  ```
  Content-type: video/mpeg
  Content-type: video/quicktime
  ```

- **Application** er en medietype som brukes om avgrensede data som ikke passer inn i de øvrige kategoriene, og spesielt benyttes denne kategorien for data som må prosesseres av spesifikk programvare.

  ```
  Content-type: application/postscript
  Content-type: application/vnd.ms-powerpoint
  Content-type: application/vnd.ms-word
  ```

- **Model**
  Medietypen "model" er kommet inn som et tillegg til de medietyper som er definert i MIME-standarden [135]. Dette er en medietype som indikerer at innholdet er en modell – en representasjon av strukturer som er satt sammen av ett eller flere objekter, f.eks. Virtual Reality Modelling Language (subtype "vrml").

  ```
  Content-type: model/vrml
  ```

4.5 Tegnsett

Koding av tekst
Tekst er den dominerende måten vi bruker for å kommunisere informasjon, og de basisenheter som tekst kan deles inn i kalles skrifttegn[7]. I den digitale verden håndterer vi egentlig binære tall, og skrifttegn lagres derfor som et nummer. Antallet bit[8] vi bruker for å representere et enkelt nummer, avgjør hvor mange forskjellige skrifttegn vi kan representere. Brukes 7 bits kan vi ha 128 forskjellige skrifttegn, men brukes 8 bits – en oktett eller byte – kan vi ha 256 forskjellige skrifttegn.

Et avgrenset og definert sett av skrifttegn kaller vi et *tegnsett*, og det finnes et meget stort utvalg av forskjellige tegnsett definert for å representere tekst. Dette skyldes en kombinasjon av to forhold:

- Representasjon av skrifttegn har tradisjonelt vært basert på færrest mulig antall bits (f.eks. 8 eller 7 bits), og de fleste av disse er basert på ASCII-standarden [7].
- For å imøtekomme behovet for å representere forskjellige skriftspråk er det utviklet forskjellige standarder innrettet mot de enkelte skriftspråkene.

Med andre ord har tekst til nå vært håndtert på en lite universell måte, noe som skaper inkompatibilitet på flere områder. I informasjonsteknologien må vi kunne håndtere:

- Utveksling av tekst mellom maskiner, operativsystem og programvare.
- Lagring av tekst over tid.
- En verden som består av mange språk:
  - Tekst-dokumenter må kunne være på forskjellige språk.
  - Flere språk må kunne kombineres i samme tekstdokument.

Unicode [167] og ISO-standarden *ISO/IEC 10646* [104] adresserer disse problemene ved å utvikle standarder hvor alle verdens språk kan være representert i det samme tegnsettet.

4.5.1 ASCII

American Standard Code for Information Interchange er en 7-bits standard for skrifttegn som er utgangspunktet for de fleste moderne tegnsett [7]. Standarden

[7] Den engelske betegnelsen for dette er "character".
[8] En bit kan være enten 0 eller 1.

er tilpasset engelsk/amerikansk uten støtte for skrifttegn som er spesifikke for andre språk[9].

Med utgangspunkt i ASCII ble det utviklet en rekke nasjonale standarder tilpasset andre språk, hvor spesialtegn i ASCII benyttes til skrifttegn som er spesifikke for de enkelte språk, f.eks. den norske standarden NS 4551-1, den tyske standarden DIN 66003, og den danske standarden DS 2089.

ASCII-standarden og de nasjonale variantene ble definert som internasjonal standard gjennom ISO/IEC 646. Ved en revisjon i 1991 ble denne standarden revidert og de nasjonale variantene ble tatt ut. ISO/IEC 646 [88] er derfor nå synonymt med ASCII-standarden.

4.5.2 Andre tegnsett

På begynnelsen av 1980-tallet eksisterte det i tillegg til de ASCII-baserte, en rekke forskjellige tegnsett som var utviklet av forskjellige leverandører og som ble brukt på spesielle plattformer. Flere av disse var varianter av ASCII, men hvor det ble brukt 8-bits representasjon, noe som gav mulighet for å bruke de 128 tegnene ut over ASCII, til språkspesifikke skrifttegn og andre symboler.

Operativsystemet MS-DOS hadde en rekke tegnsett som ble kalt "code pages". CP437 inneholdt ASCII-tegnene, men også et utvalg fremmedspråklige skrifttegn og grafiske symboler. Det ble også utviklet en rekke andre tegnsett som var tilpasninger til ISO/IEC 8859 standarden, men hvor en rekke grafiske tegn fra CP437 ble beholdt.

For Windows-operativsystemet ble det på 1990-tallet utviklet nye tegnsett som CP1252 og CP1250. Disse var stort sett kompatible med de forskjellige ISO/IEC 8859-standardene, men likevel ikke helt identiske.

Vi finner også en rekke andre leverandører med egne tegnsett, og det finnes en rekke standardiserte multi-byte tegnsett for asiatiske språk

4.5.3 ISO/IEC 8859

ISO/IEC 8859 er en familie standardiserte, flerspråklige 8-bits tegnsett, hvorav de første ble utviklet på 1980-tallet av ECMA (the European Computer Manufacturer's Association) og godkjent av ISO. Disse tegnsettene er basert på ASCII, men med utvidelser for forskjellige nasjonale tegn. Standardene dekker forskjellige språk og geografiske områder, og flere av standardene er overlappende.

ISO/IEC 8859-1 [99] (latin-1) dekker de fleste vest-europeiske språkene som norsk, svensk, dansk, engelsk, fransk, tysk, spansk m.fl. ISO/IEC 8859-1 var tidligere definert som basis-tegnsett for HTML, før HTML ble internasjonalisert [137].

Disse ISO-standardene gjør det mulig å kombinere enkelte språk i samme tekst. Begrensningen som ligger i bruk av 8-bits gjør likevel at ikke alle språk

[9] Mye av informasjonen i dette avsnittet er hentet fra: http://czyborra.com/

kan kombineres ved hjelp av ett og samme tegnsett – de vel 15 forskjellige tegnsettene som er definert, er en pragmatisk løsning på det flerspråklige problemet, som ikke løser de fundamentale problemene for informasjonsutveksling på global basis.

4.5.4 UNICODE og ISO/IEC 10646

Unicode [167] og ISO/IEC 10646 [104] er standarder som har som mål å forenkle flerspråklig og plattform-uavhengig utveksling av tekst. Begge disse standardene er basert på bruken av flerbyte-verdier for å representere skrifttegn, noe som gir mulighet for å representere alle verdens skrifttegn i ett og samme tegnsett. Unicode er fullt ut kompatibel med ISO/IEC 10646, og inneholder alle de samme tegnene som ISO-standarden[10]. Unicode-standarden inneholder i tillegg informasjon om tegnene og deres bruk.

I Unicode benyttes en 16-bit verdi for å representere hvert skrifttegn, noe som gir rom for over 65.000 tegn. Hvert skrifttegn får et unikt nummer og et navn. Selv om 65.000 skrifttegn gir rom for de fleste av verdens språk, er det i Unicode plass for ytterligere en million skrifttegn gjennom en utvidelsesmekanisme. Dette er tilstrekkelig for all verdens språk, både eksisterende og historiske, samt vitenskapelige symboler og andre relevante tegn.

ISO/IEC 10646 definerer transformeringsformatene UTF-8 og UTF-16, som i hovedsak er regler for å konvertere skrifttegnenes verdier til byterepresentasjoner. De første 256 tegnene i Unicode og ISO/IEC 10646 har identisk byte-verdi med ASCII og ISO/IEC 8859-1 (Latin-1), noe som er basis for UTF-8, et komprimert format for Unicode og ISO/IEC 10646. For å representere ASCII-tegn er det da tilstrekkelig å benytte én byte pr. tegn. Skrifttegn ut over ASCII fra Latin-1 og de fleste andre ikke-ideografiske tegn, kan representeres ved hjelp av 2 byte, mens andre Unicode-tegn, f.eks. asiatiske, må representeres ved hjelp av enn 3 eller 4 byte. Dette er med andre ord et komprimeringsformat tilpasset vestens språk. UTF-8 er definert som basis-tegnsett for XML.

[10] ISO/IEC 10646 tillater bruken av to forskjellige representasjoner, en to-byte form og en 4-byte form, hhv. UCS-2 og UCS-4. Unicode-standarden kan beskrives som en profil av ISO/IEC 10646, som er basert på to-byte formen. Ved hjelp av en utvidelsesmekanisme (extended characters) er Unicode ekvivalent med UTF-16.

Kapittel 5

Identifikatorer

5.1 Identifikatorer i digitale bibliotek

For å kunne forvalte, organisere og gjenfinne informasjonsobjekter, har vi behov for å identifisere disse på en presis og utvetydig måte. Dette er likt med det vi til daglig gjør når vi snakker om personer eller steder og bruker navn for å referere til disse. Navnene gjør at vi presist kan kommunisere om nøyaktig den personen eller det stedet. På samme måte som vi bruker navn for å organisere den verden som omgir oss og kommunisere om denne, er det behov for å bruke identifikatorer for å organisere og kommunisere om informasjonsobjekter i digitale bibliotek. Vi kan definere *identifikator* på følgende måte:

> En identifikator er et entydig navn, kjennetegn eller merke som brukes for å identifisere noe.

Dette er en generell og pragmatisk definisjon, men den gir en innledning til hva vi legger i begrepet. Identifikatorer, navn og adresser er et stort område som vi kan tilnærme oss på mange måter. I denne rapporten skal vi primært se på identifikatorer med informasjonsforvaltning og digitale bibliotek som kontekst. De identifikatorene vi fokuserer på er representert som sekvenser av bokstaver, tall eller andre tegn, og det vi identifiserer er informasjonsobjekter.

Identifikatorer og systemer for dette, er et viktig fundament i digitale bibliotek. Mye av den funksjonaliteten vi ønsker å legge i en arkitektur for digitale bibliotek, er avhengig av identifikatorer – identifikatorene gjør at vi kan kommunisere om informasjonsobjektene.

> Names are a vital building block for the digital library. Names are needed to identify digital objects, to register intellectual property in digital objects, and to record changes of ownership. They are required for citations, for information retrieval, and are used for links between objects [11].

De forskjellige løsningene og systemene for identifikatorer som er relevante i en arkitektur for digitale bibliotek, har sitt utspring i mange forskjellige miljøer.

Forlag og bibliotekverdenen har sine identifikatorer basert på behovet for å identifisere litteratur og andre åndsverk. Nettverksteknologien har sine løsninger for identifikasjon og navngiving av ressurser som maskiner og tjenester i et distribuert miljø. Selv for disse to ganske så forskjellige bruksområdene finnes det forskjellige løsninger tilpasset mer eller mindre spesifikke bruksområder og behov. Målet vi må ha for de løsninger som utvikles eller velges for identifikasjon av informasjonsobjekter i digitale bibliotek, er å kunne integrere etablerte systemer og standarder med nye løsninger som utvikles.

Identifikatorer og navn er ikke bare brukt på informasjonsobjekter. Vi finner identifikatorer og navn i mange sammenhenger enten det er organisering av personer og eiendommer i offentlige register eller det er maskiner og tjenester i et nettverk. I denne rapporten skal vi primært se på identifikatorer for informasjonsobjekter, selv om beskrivelser og teori også vil være gyldig for identifikatorer og navn brukt på andre områder.

5.2 Terminologi

De mange bruksområdene for identifikasjon og navngiving har naturlig nok ført til en lite entydig terminologi. Vi skal her se på noen generelle begreper som *navn*, *identifikator*, *adresse* og *lokator*. I tillegg finnes det mange spesielle termer knyttet til enkeltsystemer f.eks. hendel, doi, urn osv., men for en forklaring på disse begrepene henviser vi til beskrivelsene av de forskjellige identifikatorsystemene, kap. 5.6.

Det er stor variasjon i hva som legges i disse ordene, og noen entydig eller felles definisjon finner vi ikke. Navn og identifikatorer er termer som brukes om hverandre. Enkelte definerer navn som en type identifikatorer, mens andre definerer identifikator som en type navn:

> An identifier is an unambiguous label which specifies an entity. In computer science terms, an identifier is a name; the entities named occupy a specific domain of application, the namespace, and identify points in that namespace [146].

> A process that requires access to a resource which it does not manage must process a name or identifier for it. We shall use the term name to refer to names that can be interpreted by users or by programs and the term identifier to refer to names that are interpreted and used only by programs [36].

Det er likevel noen aspekter ved disse begrepene som er forskjellige. Mens *navn* indikerer noe som er lesbart eller forståelig for mennesker, assosieres *identifikator* med noe unikt og systemteknisk. Ved å bruke begrepet identifikator fremheves at dette er en unik verdi som identifiserer noe på en presis og utvetydig måte. I motsetning til *navn* henspeiler ikke *identifikator* på at dette skal være

noe menneskelig forståelig. En identifikator kan like gjerne være basert på tall eller andre tegn som ikke umiddelbart gir mening når de leses av en person. I mange sammenhenger vil likevel *navn* og *identifikator* være synonyme begreper fordi de begge brukes om kjennetegn som identifiserer noe.

Hvilken term som foretrekkes er avhengig av bruksområde. Bibliotek og forlag bruker ofte *identifikator*, fordi det er lite intuitivt å bruke *navn* på publikasjoner. Bruken av identifikatorer på informasjon har et annet utgangspunkt enn navnebruken i informasjonssystemer. Her er det primære formålet å gi publikasjoner eller andre informasjonsobjekter kjennemerke som er globalt unikt, entydig og varig - og som kan brukes for å effektiv identifisere eller referere til dette informasjonsobjektet.

> Bibliographic identifiers function as names for objects that exist both in print and, increasingly, in electronic format [120].

Identifikatorsystemer som ISBN (International Standard Book Numbering) og ISSN (International Standard Serial Number) bruker begge nummersekvenser som vanlige brukere ikke har behov for å kunne tolke eller forstå betydningen av annet enn at de er unike nummer for en publikasjon. Her brukes *nummer* synonymt med *identifikator*. I nettverksteknologien er det derimot en generell trend å bruke *navn*. Dette skyldes at navn brukes for å gi ressurser et kjennemerker som er mer egnet for menneskelig kommunikasjon. Maskiner, nettverk og andre ressurser gies navn fordi det er tungvint og lite mnemoteknisk å basere seg på f.eks. adresser i form av IP-nummer.

Med utgangspunkt i World Wide Web er det utviklet en egen termbruk - URI, URN, URL - og egen arkitektur for navn og adresser på Internett. Her brukes identifikator (Uniform Resource Identifier) som overordnet term:

> An identifier is an object that can act as a reference to something that has identity. In the case of URI, the object is a sequence of characters with a restricted syntax [15].

Navn og lokator (URN, URL) er forskjellige former for URI. URN (Uniform Resource Name) er et varig og globalt unikt navn som gir en utvetydig identifikasjon av en ressurs:

> The term "Uniform Resource Name" (URN) refers to the subset of URI that are required to remain globally unique and persistent even when the resource ceases to exist or becomes unavailable [15].

URL (Uniform Resource Locator) er identifikatorer i form av adresser eller lokaliseringsinformasjon til ressurser (f.eks. web-dokumenter).

> The term "Uniform Resource Locator" (URL) refers to the subset of URI that identify resources via a representation of their primary access mechanism (e.g., their network "location"), rather than identifying the resource by name or by some other attribute(s) of that resource [15].

Et viktig skille mellom alle disse termene går mellom *navn* og *identifikatorer* på den ene siden og *adresser* og *lokatorer* på den andre siden.

> The term name refers to a human-readable (and meaningful) identifier which refers to an entity, whereas address tell where the entity is located. In some circumstances it is useful to give an entity a location dependent name, but in general location independent names are preferable to achieve location independence [155].

Mens *navn* og *identifikatorer* er kjennemerker for en ressurs uavhengig av lokalisering eller instans, f.eks. eksemplar, er *adresse* eller *lokator* informasjon om hvor denne ressursen befinner seg. Vi kan selvsagt bruke lagringsplass eller lokaliseringsinformasjon for å referere til ressursen, men dette blir fort et problem hvis ressursen flyttes, forsvinner eller erstattes med en annen ressurs.

På World Wide Web er skillet mellom navn og lokatorer diffust fordi den eneste etablerte måten å referere til en web-side har vært å bruke URL. En URL er bygd opp av informasjon om hvor web-siden befinner seg i nettverket og hvor f.eks. denne filen er lagret i en katalogstruktur under tjener, og dette er ikke noe godt utgangspunkt som navn.

5.3 Formålet

På samme måte som bruken av navn er en viktig del av den menneskelige kommunikasjonen, har bruken av navn og identifikatorer en fundamental funksjon for håndtering av informasjonsobjekter. For informasjonsobjekter i digitale bibliotek oppfyller identifikatorer de basale funksjonene **identifisering** og **referering** i tillegg til at identifikatorer brukes spesifikt på en rekke områder.

Identifisering
Identifisering er en form for gjenkjenning hvor vi sammenligner to enheter for å finne ut om den ene er lik den andre. For informasjonsobjekter kan vi gjøre dette på flere måter. Vi kan sammenligne to instanser av samme enhet og sjekke om de er identiske, vi kan identifisere ved hjelp av data som beskriver informasjonsobjektet (metadata), eller vi kan identifisere informasjonsobjektet ved hjelp av en identifikatorer.

Det å sammenligne to instanser av samme informasjonsobjekt er ofte lite praktisk mens det å bruke metadata i mange tilfeller er tilstrekkelig, men har klare begrensninger. Det kan være vanskelig å finne et fast sett av attributter som til sammen garantert gir en unik "signatur", og dette kan gi lange og sammensatte signaturer som blir ineffektive. Løsningen på dette er ofte å opprette et nytt attributt kun beregnet til identifikasjon, og gi hvert informasjonsobjektene en unik verdi. Fordelen ved å bruke egne identifikatorattributter er at vi får en presis og pålitelig identifikasjon basert på minst mulig data.

Referering
Som en konsekvens av at vi kan identifisere et informasjonsobjekt har vi også mulighet til å referere til dette informasjonsobjektet. Vi kan vise til informasjonsobjektet som logisk enhet uten å kjenne til faktisk lokalisering, og vi kan på denne måten behandle informasjonsobjektene uten å aktivere det, kun ved å bruke identifikatoren.

5.4 Problemområder

Navn og identifikatorer har i de senere årene vært tema for mange utviklings- og utredningsprosjekter, og det finnes mange systemer for identifikatorer. Internett og World Wide Web har resultert i store mengder av informasjon som nå gjøres tilgjengelig for et globalt publikum. Behovet for systemer for å håndtere denne informasjonen på en automatisk måte har aktualisert behovet for identifikatorer. Både for Internett generelt og for digitale bibliotek spesielt, er håndtering av identifikatorer et krevende problem av flere grunner:

- Det finnes i dag ikke noe globalt og universelt identifikatorsystem som er velegnet alle typer informasjonsobjekter, men det er mange forskjellige identifikatorsystemer som er mer eller mindre tilpasset spesifikke områder.

- Vi må kunne håndtere et mangfold av identifikatorsystemer. Forskjellige områder vil ha forskjellige krav til identifikatorer, og det er derfor behov for mange systemer - ikke bare ett. Det meste av informasjon som er publisert via forlag har allerede identifikatorer (ISBN, ISSN), og det er behov for løsninger som kan innlemme eksisterende navnesystemer.

- Den globale informasjonsdeling i et nettverksmiljø som vi går mot, stiller krav til ny funksjonalitet, og mange av de eksisterende måtene å navngi informasjonsobjekter på har ikke de kvaliteter vi har behov for. Informasjon som gjøres tilgjengelig på Internett identifiseres ofte kun ved hjelp av en URL, og dette har ført til mange referanser på World Wide Web som er feil eller som viser til web-sider som ikke lenger er tilgjengelige (nettråte).

- Mange informasjonsobjekter er uten identifikator som er kjent for omverdenen, eller som omverdenen kan bli kjent med (f. eks. metadataposter i en database, artikler som hentes ut ved hjelp av cgi-skript o.l.). Interne identifikatorer er ustabile i den forstand at de er relatert til og avhengig av et spesielt system.

5.5 Aspekter ved identifikatorsystemer

Gjennom en rekke prosjekter og spesifikasjoner de siste årene begynner vi å få en litt klarere teori og modell for navn og identifikatorer. Det er mange innfallsvinkler til identifikatorer, men vi skal her se på de mest sentrale aspektene og forskjellige begreper som er viktige. *Identifikatorsystem* brukes her som en samlebetegnelse på det sett av regler og karakteristiske egenskaper som omgir en spesifikk form for identifikator; både selve identifikatoren slik den fremtrer, og de formelle sidene ved det å opprette, administrere og bruke identifikatoren.

5.5.1 Identifikatoren

Identifikatorskjema
Identifikatorskjema benyttes om alle de regler som gjelder for en identifikator, både syntaks og administrative sider. I enkelte sammenhenger brukes skjema synonymt med det vi her har definert som identifikatorsystem.

Syntaks
I et navnesystem vil det være regler for hvordan en identifikator skal formuleres. Dette kan være regler for hvilke bokstaver, tall eller andre tegn som kan benyttes, og andre strukturelle regler for hvordan selve identifikatoren skal bygges opp. Et eksempel på dette er et ISSN som skal bestå av 8 siffer.

Unikhet – entydighet
Unik betyr at det bare finnes en. Med unike identifikatorer menes at hver identifikator skal være entydig og referere til en ting. For informasjonsobjekter betyr unike identifikatorer at en identifikator bare skal peke til ett informasjonsobjekt. Unikhet kan være global, og da vil en og samme identifikator representere samme enhet på tvers av system og organisasjoner. Vi kan også ha unikhet innen ett system eller en organisasjon, men da vil identifikatoren nødvendigvis være meningsløs utenfor det system eller organisasjon den er gyldig i.

Lesbarhet
Mange identifikatorer skal brukes av mennesker. For at identifikatoren skal være lett å kommunisere, skrive ned eller huske, kan struktur og bruk av tegn inrettes mot dette. Skal identifikatorene utelukkende håndteres av programmer, er det andre krav som vil dominere, f.eks. kompakthet.

Smarte og dumme identifikatorer
Vi kan si at identifikatorer er smarte eller dumme. Smarte identifikatorer bærer i seg informasjon, mens dumme identifikatorer er helt uten informasjon.

Baserer vi identifikatorene på fortløpende nummer av bokstaver eller tall, som ikke har annen hensikt enn å gi informasjonsobjektet en unik identifikator, kan vi kalle dette for dumme identifikatorer. For å knytte informasjon til slike dumme identifikatorer er vi avhengig av å bruke egne databaser eller register.

En smart identifikator er en identifikator som ikke bare er et unikt navn, men hvor vi også kan avlede meningsfull informasjon fra de tegnene som er brukt i identifikatoren. Eksempler på slik informasjon kan være å legge årstall inn i identifikatoren. ISSN er et eksempel på en dum identifikator, mens ISBN er et eksempel på en identifikator som er litt smartere. Fra et ISBN kan vi avlede både land og forlag hvis vi har kjennskap til de forskjellige kodene som brukes for denne informasjonen.

Sjekk-tall
Som et middel for å sikre korrekt gjengivelse av en identifikator kan vi bruke sjekk-tall. Et sjekk-tall er en matematisk beregnet verdi basert på de tegn som er brukt i identifikatorstrengen. Som oftest ønsker vi et minst mulig sjekk-tall, og vi kan bruke beregninger som produserer et ensifret eller tosifret tall[1]. Et sjekktall kan innlemmes som del av identifikatoren. Sjekktallet kan brukes bl.a. for å sikre at identifikatoren blir gjengitt riktig ved registrering. I et ISSN er det 8. sifferet et sjekktall. Dette betraktes som en del av ISSN og regnes ut på basis av de øvrige tall etter en metode som kalles "modulus 11 med vektene 8 til 2". Blir sjekktallet 10, brukes romertall X [101].

Kontekst og navnerom
En identifikatorer er alltid relatert til en *kontekst* fordi den bare har mening når vi vet hva slags identifikator dette er. Vi må kjenne kontekst for å kunne bruke eller forstå en identifikator. Nummersekvensen "7-709265-1" er meningsløs hvis vi ikke vet hva slags nummer dette er. Oppgir vi ISBN som prefiks eller av andre grunner vet at dette er et ISBN, er vi i stand til å tolke eller nyttiggjøre oss dette nummeret, f.eks. til å bestille den boka nummeret representerer. Her er ISBN konteksten som gir rammen for hvordan dette nummeret skal tolkes eller brukes.

I en del tilfeller vil en kontekst være implisitt bestemt mens i andre tilfeller er det behov for å uttrykke kontekst eksplisitt. ISBN som trykkes på publikasjoner, trykkes alltid med prefiks ISBN i henhold til standarden, slik at det eksplisitt vises at dette er et ISBN. I sammenhenger hvor det er mulig å bruke forskjellige typer identifikatorer på samme sted, for eksempel Dublin Core-metadata hvor identifikator-feltet kan brukes til forskjellige typer av identifikatorer, er det et nødvendig å kunne uttrykke konteksten til identifikatoren f.eks. ved hjelp av en kvalifikator [49].

Navnerom er delvis synonymt med *kontekst* ved at begge er rammer for identifikatoren. Mens kontekst henspeiler på referanserammen for det å gjenkjenne/identifisere forskjellige typer av identifikator, defineres navnerom som det sett av navn/identifikatorer vi har eller som kan finnes i en kontekst. Eksempelvis er alle de forskjellige kombinasjonene av tall som et ISBN kan ha, navnerommet til ISBN.

[1] ISO standard 7064 inneholder en rekke sjekktall-systemer som blant annet er brukt i ISBN og ISSN.

Kontekst kan være basert på en flat modell eller være hierarkisk oppbygd. Med flat modell menes at alle identifikatorer er relatert til samme kontekst og at identifikatorer ikke deles i undergrupper under denne konteksten. Alle identifikatorer er relatert til samme kontekst og er unike innenfor denne felles konteksten. ISSN er et eksempel på bruk av en flat modell. Et ISSN er en fortløpende numerisk identifikator som tilordnes tidsskrifter, uten videre inndeling.

Identifikatorer med hierarkisk kontekst er basert på oppdeling og bruk av undergrupper. Toppnivå-konteksten er felles for alle identifikatorer, men denne kan inndeling i forskjellige undergrupper – subkontekster – som igjen kan være inndelt i nye undergrupper. Identifikatorer som hører til en og samme subkontekst må være unike innen denne konteksten, mens identifikatorer fra forskjellige subkontekster kan være like. For å oppnå at identifikatorene er unike innen hele navnesystemet, er det da nødvendig at navn eller koder for de forskjellige subkontekster også inngår i identifikatoren. ISBN er basert på bruk av land-/språk-koder og under disse er det egne koder for forlag eller utgiver. Dette kan sees på som et hierarkisk oppbygd identifikatorsystem. Toppnivåkontekst er ISBN-standarden, mens land er subkontekst. Forlag eller utgiver er subkontekst under land.

For de som oppretter nummer, er bruk av hierarkisk kontekst ofte en fordel. ISBN-kontoret i Norge har fått tildelt landkoden *82* sammen med retten til å opprette subkontekster for forlag og utgivere i Norge. De forskjellige forlag og utgivere får tildelt forlagskoder fra ISBN-kontoret.

5.5.2 Administrering

Administrering er et viktig aspekt ved et identifikatorsystem. Med administrering mener vi organisatoriske forhold rundt det å opprette, tilordne og bruke identifikatorer. Det er verdt å merke seg at det ofte er et samspill mellom tekniske løsninger og organisatorisk modell. Navnesystemet vi velger å bruke eller utvikler må være i overensstemmelse med den verden identifikatoren skal brukes i.

Et enkelt identifikatorsystem kan for eksempel bestå av et løpende nummer som tilordnes informasjonsobjektene etter tur, men til og med for et slikt enkelt system vil det være behov for administrative ordninger. Selv om vi ikke trenger å kjenne hvilke identifikatorer som er tilordnet hvilke informasjonsobjekter, må vi holde rede på hvor langt vi er kommet i nummerrekken, og vi må administrere oppdatering av nummeret. Allerede for et slikt enkelt system er det et spørsmål om hvem som skal ha lov til å utstede identifikatorer. Den organisasjon som er autorisert til å opprette identifikatorer, kalles ofte for en *navneautoritet*, og de underorganisasjoner som navneautoriteten evt. delegerer videre, til kalles *subnavneautoriteter*.

Bygger vi ut identifikatorsystemet med mer funksjonalitet og hukommelse, blir det enda flere aspekter som skal administreres. Med lagring av informasjon

om hvilke informasjonsobjekter som har fått tildelt hvilke identifikatorer, oppstår det ansvar for å administrere data over tid. Vi må også kunne garantere at dette er informasjon som ikke forsvinner hvis organisasjonen avvikles.

Vi kan generelt si at det er to motpoler av administrative modeller for identifikator, sentralisert og distribuert:

- **Sentralisert**
  En sentralisert administrering av navn har vi når en enkelt organisasjon eller enhet er ansvarlig for å utstede navn fra ett felles navnerom.

- **Distribuert**
  En distribuert administrering av navn har vi når mange organisasjoner er autoriserte til å utstede navn. Navnerommet er da ofte hierarkisk oppbygd.

For bruk i en enkelt administrativ enhet vil det være få problemer knyttet til det å opprette og tilordne unike navn. Her kan det være fordelaktig å bruke en sentralisert modell basert på et flatt navnerom. Også hvis vi ønsker stor grad av kontroll med de identifikatorene som utstedes, hvilke informasjonsobjekter som får identifikatorer og hvordan identifikatoren utformes, kan dette være enklest å realisere ved hjelp av en sentralisert administrering av navn.

Hvis mange aktører, organisasjoner eller enheter skal bruke et felles navnesystem, kan en sentralisert flat modell basert på et flatt navnerom fort bli rigid og tungvint. Denne modellen gir ingen fleksibilitet i hvordan identifikatorer kan utformes siden den er basert på et felles navnerom og overordnete felles regler for utforming av identifikatorer. En mer fleksibel administrering av navn kan oppnåes ved å ta i bruk en distribuert modell kombinert med hierarkisk oppbygde navnerom. Navnesystemet må da ha en toppnivåautoritet som kan delegere rettigheter til underautoriteter. Hierarkiet kan være definert slik som vi finner det i ISBN, eller det kan være udefinert og kun basert på prinsippet om at en subautoritet kan opprette nye underautoriteter og delegere rettigheter videre nedover i et nærmest uendelig hierarki. Dette kan også brukes for å gi lokale enheter stor frihet i formulering av identifikatorer ved at det felles er spesifisert overbyggende regler for syntaks som alle må overholde, mens de forskjellige subautoriteter kan legge til egne regler for sitt sub-navnerom kun gjeldende for identifikatorer i dette navnerommet.

Identifikatorer og administrering av disse har også en organisatorisk og økonomisk side. For identifikatorsystemer som bare er i bruk av en enkelt virksomhet, vil drift og ansvar for identifikatorsystemet være oppgave som uten videre kan inngå som del av annen drift, men når et identifikatorsystem deles av mange organisasjoner, oppstår det behov for egne organisasjoner som kan koordinere og administrere bruken av identifikatorsystemet. For ISO-standardene er det definert toppnivå-organisasjoner som har oppgaven med å koordinere standardene, med varierende oppgaver i henhold til identifikatorsystemet. Avhengig av organisasjonen som koordinerer identifikatorsystemet og de tjenester som

er knyttet til identifikatorsystemet, kan det også være behov for en fordeling av kostnadene knyttet til drift og administrering.

5.5.3 Bruksområde

Et identifikatorsystem gir ofte rammer for både type informasjonsobjekt som kan tilordnes en identifikator og for hva som skal regnes for en enhet som kan identifiseres i identifikatorsystemet. Enkelte identifikatorsystemer er rimelig spesifikke på dette punktet mens andre er mer generelle. En differensiert begrensning i bruksområde kan også knyttes til navnerom i et hierarkisk oppbygd identifikatorsystem.

Et identifikatorsystem eller navnerom kan være avgrenset til spesifikke typer av informasjonsobjekter enten dette er tekst, lyd, bilder eller andre kriterier. ISO-standardene er spesifikke og har klare definisjoner av hva som kan tilordnes en identifikator, mens flere av identifikatorsystemene som utvikles for informasjon på Internett (DOI, The Handle System og PURL) er mindre stringent og gir få eller ingen begrensninger i hva som kan tilordnes en identifikator.

Granularitet er også et relevant aspekt ved tilordning av identifikator. De fleste informasjonsobjekter kan naturlig deles i mange mindre enheter. Et tidsskrift kan deles i hefter, et hefte kan videre deles inn i de forskjellige artikler, og artiklene kan igjen deles inn i nye logiske enheter, slik at vi i realiteten har et hierarki av komponenter. Muligheten for å tilordne identifikatorer til deler av f.eks. et tidskrift eller en bok har vært dårlig, men vi ser at nyere standarder som SICI [6] og den foreslåtte BICI standarden [122] fokuserer på nettopp dette behovet.

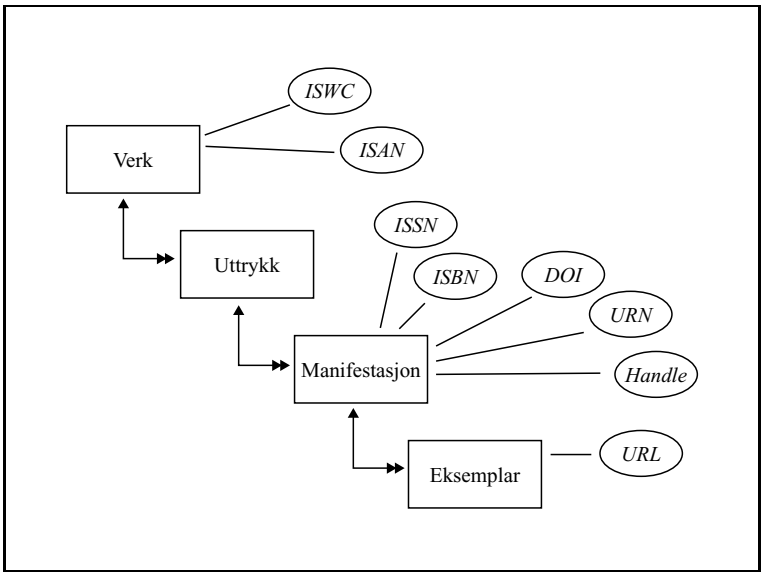
Versjoner er et annet område som kan være problematisk. Særlig gjelder dette digital informasjon som enkelt kan endres og korrigeres. Et sentralt spørsmål er når informasjonsobjektet er så endret at det er behov for en ny identifikator for å skille det fra tidligere versjoner.

Forholdet mellom logiske og fysiske enheter er også en annen problemstilling. Et eksemplar er en faktisk enhet mens et verk er en abstrakt enhet. FRBR-modellen opererer med verk, utrykk, manifestasjon og eksemplar. Alle disse kan i en eller annen sammenheng være berettiget en egen identifikator. Et eksempel på hvordan forskjellige identifikatorer peker på de forskjellige enheter er vist i fig. 5.1.

5.5.4 Identifikatortjenester

I et identifikatorsystem kan det være tjenester for oppretting, administrering og bruk av identifikatorene.

Generering av identifikatorer
Identifikatorer kan genereres manuelt eller vi kan utvikle systemer som utfører denne oppgaven. For identifikatorsystemer som består av et endelig sett med nummer (som ISBN og ISSN), er dette et spørsmål om å tilordne disse sifrene til



Figur 5.1: Identifikatorer og korresponderende produktentitetene fra FRBR-modellen

publikasjoner, enten direkte eller via å tilordne delmengder av de mulige sifrene til underautoriteter. Identifikatorer kan også tilordnes automatisk enten til et begrenset antall autoriserte brukere eller fritt som den generering av URN som de skandinaviske nasjonalbibliotekene tilbyr [140]. For identifikatorer som ikke er basert på et endelig sett av mulige identifikatorer, men som har uendelige antall på grunn av variabel lengde, er dette en noe mer komplisert oppgave fordi man ikke har oversikt over mulige identifikatorer. Unikhet kan i disse tilfellene sikres ved å opprette unike prefiks og distribuere ansvaret for å kontrollere og generere unike identifikatorer under disse prefiks (hierarkisk kontekst).

Identifikator-registre
Det er ofte behov for å registrere og lagre metadata knyttet til identifikatoren. Vanligvis er dette metadata om informasjonsobjektet som identifiseres. Informasjon om tidsskrifter som er tilordnet et ISSN, er lagret i det internasjonale ISSN-registeret i et MARC-basert format – en database også eksterne brukere kan få tilgang til mot betaling. For DOI (Digital Object Identifier) er det også planer om å utviklet et lignende format og register.

Oversetter-tjenester
Særlig for digitale informasjonsressurser tilgjengelig på nett er det behov for tjenester som automatisk kan oversette fra navn til lokaliseringsinformasjon for automatisk innhenting av et informasjonsobjekt. Slike tjenester er en viktig del av de identifikatorsystemene som utvikles for Internett. DOI og The Handle System bruker slike tjenester for å oversette fra identifikatorer til URL, slik at

vi kan benytte disse identifikatorene i web-dokumentenes klikkbare lenker på samme måte som vi bruker URL. Forløpig er det behov for egne programmer (plugins) for å få denne funksjonaliteten i vanlige web-leser, men det brukes også HTTP-baserte løsninger hvor identifikatoren oversettes av en web-tjener. Også for ISSN er det utviklet en plugin slik at man kan bruke ISSN i web-dokumentene. Når vi klikker på en slik lenke, resulterer det i et oppslag i ISSN-registeret slik at brukeren kommer direkte til f.eks. et tidsskrift som er digitalt tilgjengelig. Foreløpig er slike løsninger midlertidige i påvente av et bedre og globalt system for bruk av URN (se kap. 5.6.4).

5.6 Identifikatorsystemer – eksempler

Det finnes en rekke identifikatorsystemer som er i bruk eller er under utvikling. Vi skal her se på et utvalg av de som er mest relevante for digitale bibliotek. Her er det fokusert på følgende:

- **ISO-standardene**
  ISBN, ISSN, ISMN, ISRN, ISRC, ISWC, ISAN
- **Identifikatorsystemer fra forlagsverdenen**
  SICI, BICI
- **Identifikatorsystemer for Internett**
  URI, URL, URN, The Handle System, DOI, PURL, DNS

5.6.1 ISO-standarder

ISO-standardene er viktige fordi dette er internasjonale standarder og fordi de fleste av disse er vel etablerte, har stor oppslutning, og allerede er brukt på store mengder av tradisjonelt publiserte informasjonsobjekter.

En gjennomgang av de forskjellige ISO-standarder for nummerering av informasjonsobjekter viser at selv om det er likheter mellom de forskjellige standardene, er det også store forskjeller både med hensyn til administrative modeller og hvordan identifikatorene bygges opp. Fellestrekk er at alle disse identifikatorene i hovedsak er basert på tall.

De forskjellige ISO-standardene tar utgangspunkt i forskjellige typer materiale. Informasjonsobjektene kan likevel få nummer fra flere standarder, f.eks. kan enkelte informasjonsobjekter både karakteriseres som trykt bok og periodikum.

ISBN
ISBN er en forkortelse for *International Standard Book Numbering* og er en identifikator som kan tilordnes trykte bøker og andre monografiske utgivelser. ISBN er definert i ISO-standard 2108 [89, 106].

Vi kan karakterisere ISBN som en delvis intelligent identifikator. Hvis vi kjenner hva de forskjellige deler av et ISBN representerer, kan vi avlede informasjon om hvilket land en bok er utgitt i og hvilket forlag som har utgitt boka. Et ISBN består av 10 siffer som inneholder følgende deler:

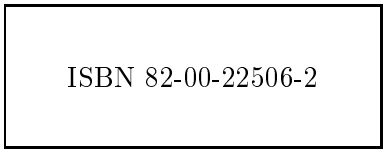
- **Gruppe-identifikator**
  Første del av et ISBN representerer en gruppe, f.eks. nasjonale, geografiske eller språkrelaterte grupper. De fleste ISBN er basert på nasjonale grupper hvor tallet Norge identifiseres med 82. Gruppeidentifikatorer allokeres av det Internasjonale ISBN-byrået og varierer i lengde basert på forventet produksjon av titler innen en gruppe.

- **Utgiver- eller produsent-identifikator**
  Dette er et tall som representerer en utgiver eller produsent. De fleste forlag i Norge har sine egne utgiver-identifikatorer. Tallet allokeres internt i gruppen eller av en organisasjon som har autoritet til å gjøre dette for en gruppe.

- **Tittel-identifikator**
  Utgiver eller produsent tilordner boken en tittelidentifikator som må være unik.

- **Sjekk-tall**
  Det siste sifferet av et ISBN er et sjekk-tall.

Når et ISBN presenteres, skal det ha prefiks "ISBN", og de forskjellige deler skal være adskilte med mellomrom eller bindestrek.



Figur 5.2: Eksempel på et ISBN

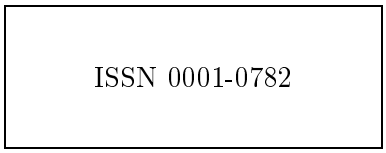
Øverste myndighet for ISBN er det internasjonale ISBN-byrået, men autorisasjon til å utstede utgiver- og produsent-nummer distribueres til underbyråer. I Norge er det ISBN-kontoret ved Nasjonalbiblioteket som har denne autoriteten.

ISSN

ISSN er en forkortelse for *International Standard Serial Number* og er definert i ISO standard 3297 [101, 106]. Hvert ISSN identifiserer et spesifikt periodikum, for eksempel et tidsskrift eller en avis.

Et ISSN er en "dum" identifikator hvor tallene ikke representerer noen informasjon.

ISSN er basert på 8 siffer hvor det siste er et sjekk-tall. Ved presentasjon skal et ISSN prefikses med "ISSN" og et mellomrom fulgt av tallene som to grupper på 4 siffer med bindestrek mellom.

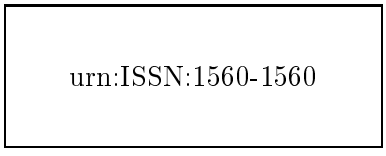


Figur 5.3: Eksempel på et ISSN

ISSN administreres som et internasjonalt nettverk av nasjonale eller regionale sentra med et overordnet internasjonalt senter i Paris. De regionale sentrene er ansvarlige for å tilordne ISSN og registrere periodika utgitt i eget land eller region, og overfører denne informasjonen til det internasjonale senteret. Det internasjonale senteret er ansvarlig for å tilordne ISSN og registrere publikasjoner som er internasjonale eller de som ikke er dekket av et regionalt senter. Det internasjonale senteret er også ansvarlig for å utstede blokker av nummer til de regionale sentrene og for å drifte og utgi et internasjonalt register over ISSN. Dette registeret finnes også tilgjengelig på World Wide Web mot betaling.

ISSN har rimelig strenge definisjoner på relasjonen mellom et ISSN og et tidsskrift, og forskjellige media skal ha forskjellige ISSN. Disse reglene er med på å sikre utvetydig identifikasjon, men dette problematiserer også forholdet mellom trykte og digitale utgaver fordi samme periodikum kan ha flere identifikatorer avhengig av medium.

I det internasjonale registeret over ISSN legges det inn informasjon om de enkelte periodika. Det benyttes et MARC-basert format som også kan inneholde URL til tidsskriftet. Denne informasjonen gjør at det er mulig å oversette fra et ISSN til en URL. Forholdene er derfor til stede for å utvikle en ISSN-navnetjeneste. Et pilotprosjekt har implementert en slik navnetjeneste med et plugin-program for web-lesere slik at det er mulig å bruke ISSN som lenke i et web-dokument (se fig. 5.4).



Figur 5.4: Eksempel på et ISSN som URN

Dette er fortsatt på eksperimentstadiet og regnet for å være en midlertidig løsning frem til en mer global løsning på oversetting av URN til URL er realisert.

Andre ISO-standardiserte identifikatorsystemer
I tillegg til ISBN og ISSN finnes det også en rekke andre ISO-standarder for nummerering og identifisering av annet materiale.

- **ISMN**
  International Standard Music Number er et identifikatorsystem for trykte musikkutgivelser [91, 105]. Et ISMN består av fire elementer: Kjennetegnet "M", utgiveridentifikator som identifiserer utgiver, tittelidentifikator som identifiserer en tittel, og et sjekk-tall. ISMN administreres på tre nivåer: det internasjonale nivå, det nasjonale/regionale nivå, og forleggernivå. En inndeling som er tilsvarende den for ISBN. Den internasjonale administrasjonen av systemet er ivaretatt av *The International Standard Music Number Agency*, som har et råd bestående bl.a. av representanter fra ISO, musikkforlegger- og musikkbibliotekforeninger. De nasjonale/regionale ISMN-kontorene sørger for tildeling av ISMN til forleggere, og i Norge ivaretas dette av ISMN-kontoret Norge som er tilknyttet Nasjonalbiblioteket.

- **ISRN**
  International Standard Technical Report Number er et identifikatorsystem for tekniske rapporter; dokumenter som beskriver resultater fra forskning, undersøkelser og andre studier [93]. Et ISRN består av maksimum 36 alfanumeriske tegn og er inndelt i tre segmenter; rapportkode som identifiserer organisasjon som gir ut rapporten, sekvensinformasjon som kan være kombinasjonen av år, et unikt nummer og versjonsidentifikator hvor bare nummeret er obligatorisk, og siste del er en tobokstavs kode for land. Det er også mulig å bruke et lokalt suffiks til ISRN, men dette er ikke å regne som del av standardnummeret. ISRN administreres av et internasjonalt byrå som overvåker bruken av ISRN og koordinerer tilordning av unike rapportkoder.

- **ISRC**
  Internation Standard Recording Code er en identifikator for musikkopptak og musikkvideo-opptak [86, 83]. Hver ISRC er en unik og permanent identifikator for et spesifikt opptak som kan inngå i et produkt som et permanent digitalt fingeravtrykk. Slike innkodede ISRC kan være et middel for automatisk å identifisere opptak for betaling ved bruk av opptaket. Også ISRC er bygd opp av segmenter.

- **ISAN**
  International Standard Audiovisual Number er en ny ISO-standard under utarbeidelse [163]. ISAN er en dum identifikator på 16 siffer som skal gi permanent og unik identifisering av audiovisuelle verk. Med audiovisuelt verk menes bevegelige bilder som video eller film. Et ISAN vil identifisere et verk, og vil derfor ikke være knyttet til spesifikke formater eller det som

kalles manifestasjoner i henhold til FRBR-modellen (kap. 3.7). En ISAN vil derfor være den samme uavhengig av om en video er på DVD-plate eller en VHS- kassett.

- **ISWC**
  International Standard Musical Work Code er også en ny ISO standard under utarbeidelse [164]. Denne tar utgangspunkt i musikk-verk og behovet for identifikatorer for disse. Tilsvarende som for ISAN er ISWC en identifikator for verk og ikke for uttrykk eller publikasjoner.

5.6.2 SICI - Serial Item and Contribution Identifier

SICI er en identifikator av variabel lengde som skal gi unik identifikator for periodika (serial items) og de forskjellige logiske deler av et periodikum – bidrag (contribution). Et tidsskrifthefte eller artikler i et hefte, er enheter som kan navngis ved hjelp av SICI.

SICI ble definert av SISAC (Serials Industry Advisory Commitee) i 1991 og revidert i 1996. Den er akseptert som en ANSI/NISO-standard, og siste revisjon av standarden er Z39.56-1996 [6].

Formålet med utviklingen av SICI var å gi en felles lenke for forfattere og utgiveres orginalverk, referanser, abstract og indeks-databaser, uavhengig av hvilket format verket har.

Bak utviklingen av SICI ligger det noen strategiske valg i form av designkriterier:

- Begrense område for identifikator til "serial items and contributions".
- Dekke mest mulig av periodika, uavhengig av form.
- Gjøre mulig uavhengig utledning av SICI identifikatoren basert på verket selv, eller en referanse til det, uavhengig av om periodikumet er utgitt eller om utgiver har trykt identifikator på serialen.
- Gi kortest mulig kode konsistent med unik identifikasjon.
- Være konsistent med og basere seg på eksisterende standarder som ISSN.
- Interoperabilitet mellom SICI koder fra forskjellige kilder

For å identifisere periodikumet bruker SICI periodikumets ISSN. De øvrige elementene, med unntak for lokal identifikator, er generert ut i fra heftet og bidraget.

SICI er et eksempel på en intelligent identifikator. Dette er en identifikator som bare indirekte er avhengig av administrative tilordning. Når vi sier delvis, skyldes dette at SICI er basert på ISSN, noe som gir SICI en avhengighet av et identifikatorsystem som er basert på administrativ tilordning.

Fordeler med SICI er at de kan genereres av andre enn forvaltere av publikasjonen. Så lenge periodikumet har et ISSN, kan også alle bidrag i et hefte ha SICI eller tilordnes en SICI av alle de som har behov for slike identifikatorer. En SICI-identifikator som genereres av én organisasjon, er ikke nødvendigvis identisk med identifikatorer som er generert av en annen organisasjon. Dette gjelder når SICI baseres på lokale identifikatorer.

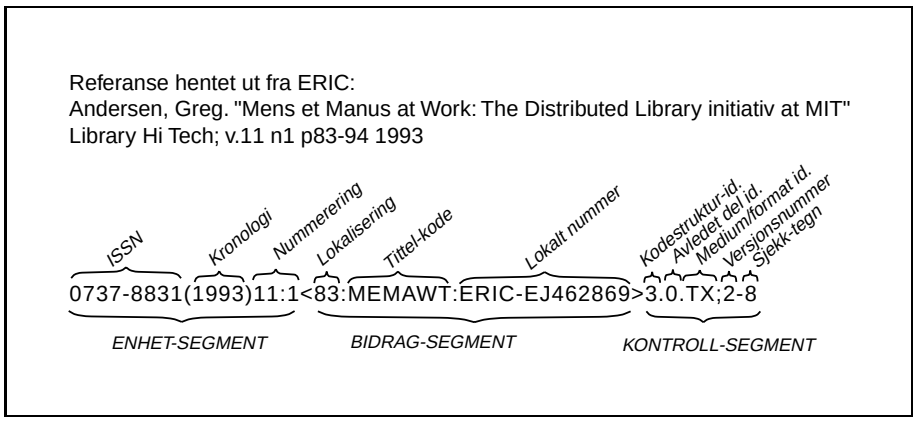
SICI kan brukes på tre forskjellige måter eller nivåer. Hvilket nivå som er brukt, identifiseres via en delkomponent av SICI-identifikatoren – kodestruktur-identifikator – CSI (Code Structure Identifier):

- **CSI = 1** Brukes for å identifisere enkeltutgivelser i et periodika, basert på ISSN og kronologi.
- **CSI = 2** Brukes for å identifisere bidrag i en enkeltutgivelse av et periodika, basert på startside og første bokstav fra tittelord.
- **CSI = 3** Brukes for å identifisere bidrag i et hefte, basert på nummer fra et lokalt system (PII, databasenummer o. l.), eventuelt kombinert med startside og første bokstav fra tittelord.

En SICI-identifikator har en strukturell modell bestående av tre segmenter. Hvilke komponenter som skal være med, ikke være med, bare være med hvis de finnes, avhenger av hvilket SICI-nivå (SCI) man velger.

- **Enhet-segment** består av dataelementer som identifiserer periodikumet:
  - *ISSN.*
  - *Kronologi* som identifiserer en spesifikk dato, f.eks. dato som gjengis på omslaget. Dette må brukes hvis det finnes.
  - *Nummerering* som identifiserer en spesifikk utgave av periodikumet, f.eks. volum og nummer.
- **Bidrag-segment** er dataelementer som beskriver bidrag i et periodikum:
  - *Lokalisering* er startsted for bidraget, f.eks. sidenummer for første side av en artikkel i et hefte.
  - *Tittelkode* som er første bokstav fra ordene i tittel, maks seks tegn og konvertert til store bokstaver.
  - *Lokale nummer* er alternative nummereringer som brukes av enkelte organisasjoner eller i enkelte systemer.

- **Kontroll-segment** er dataelementer for validering av koden, versjon og formatet til koden. Dette er et nødvendig segment fordi det gir informasjon om hvordan de andre elementene skal tolkes og prosesseres.
  - *Kodestruktur-identifikator* identifiserer de tre forskjellige nivåene SICI kan brukes på.
  - *Identifikator for avledet del* gir en metode for å identifisere om SICI-nummeret identifiserer deler av en serial som ikke er et bidrag eller identifiserbar del av et bidrag, f.eks. innholdsfortegnelse, indeks, sammendrag.
  - *Medium/format identifikator* gir mulighet for å presisere presentasjonsmedium, f.eks. trykt tekst.
  - *Versjonsnummer* identifiserer hvilken versjon av SICI som er brukt for å generere identifikatoren.
  - *Sjekk-tall* som genereres for SICI-identifikatoren



Figur 5.5: Eksempel på et SICI-nummer

Selv om SICI bruker begrepet "unik identifikator" er det likevel en teoretisk mulighet for at bidrag kan ende opp med samme identifikator. Sjansen for at dette skal inntre er likevel så liten på grunn av måten en identifikator bygges opp, at dette må være akseptabelt. Dette er gjort for å få den rette balansen mellom unikhet og kompakte identifikatorer.

5.6.3 BICI - Book Items and Contributions Identifier

På samme måte som det er behov for å identifisere et bidrag i et serial er det også i en del tilfeller behov for samme muligheter for bøker. BICI er et forslag til en standard for identifisering av enkeltdeler i en bok utviklet av *Book Industry Communications* [122]. Utgangspunktet for denne identifikatoren er behovet

for å kunne identifisere deler av en bok; et kapittel eller sekvens av sider og lignende.

SICI er kun beregnet for periodika, og den eneste lignende standarden for å kunne identifisere deler av en bok har vært *Biblid*, som er trukket tilbake på grunn av manglende bruk [22].

Formålet med BICI er i stor grad håndtering av opphavsrett. Det å kunne referere til deler av en bok er nødvendig for å kunne forvalte opphavsrett, eksempelvis når kompendier settes sammen av deler fra flere bøker og lignende. BICI muliggjør også en presis kommunikasjon om deler av bøker for andre formål, selv om det er opphavsrettslige aspekter som i følge utkastet til en standard er hovedformålet.

BICI følger stort sett den samme oppbygning som SICI, men er basert på ISBN og har en del elementer og definisjoner som er tilpasset bøker.

BICI er foreløpig kun et utkast til en standard, og vi skal derfor ikke gå detaljert inn på oppbygning av denne identifikatoren, men henviser til det foreløpige utkastet til standarden.

5.6.4 Identifisering og adressering på Internett

Da World Wide Web ble introdusert i 1990, var en av intensjonene å skjule forskjellige protokoller og aksessmetodene for brukerne, bak en felles og uniform syntaks kalt URI (Universal Resource Identifier). Intensjonene ved URI er godt illustrert ved undertittelen til den opprinnelige RFC som definerte og beskrev URI [17]:

> A Unifying Syntax for the Expression of Names and Adresses of Objects on the Network as used in the World-Wide Web

URI er fortsatt det samlende begrep for identifikatorer og adresser på Internett, men i nyere spesifikasjoner er "Universal" byttet ut med "Uniform" slik at URI i dag er en forkortelse for *Uniform Resource Identifier*. URI-begrepet er ikke bare knyttet til hypertekstformatet HTML og protokollen HTTP. Det er etter hvert blitt en felles identifisering og addresseringssyntaks også for andre deler av Internett, og utviklingen av URI foregår nå i regi av Internett-organisasjonen IETF.

De mest kjente formene for URI er de som vi vanligvis kaller URL (Uniform Resource Locators). Dette er identifikatorer som har det til felles at de identifiserer en web-ressurs ved hjelp av lokaliseringsinformasjon. Når vi bruker

```
http://www.bibsys.no/index.html
```

for å adressere et web-dokument tilgjengelig fra en web-tjener er dette en URL. Et annet velkjent eksempel på URL er

```
ftp://server.bibsys.no/pub/dok1.doc
```

som vi bruker for å hente filer ved hjelp av filoverføringsprotokollen FTP.

Allerede tidlig i utviklingen av World Wide Web var det klart at det å identifisere en ressurs ved hjelp av lokaliseringsinformasjon ikke var en god nok løsning for alle typer av informasjonsobjekter. Ressurser kan skifte plass, web-tjenere skifter navn og adresse, og ressurser fjernes fra nettet. En URL fungerer bare som identifikator så lenge ressursen fortsatt finnes og er lagret på samme adresse. URL har derfor klare svakheter som identifikator for informasjonsobjekter når det stilles krav til varige, lokaliseringsuavhengige og eksemplaruavhengige identifikatorer. Løsningen på disse problemene er en annen subtype av URI som kalles URN (Uniform Resource Name). En URN kan karakteriseres som globalt unik, lokaliseringsuavhengig og varig. Selv om URN-begrepet har vært mye omtalt og er beskrevet allerede i de tidlige spesifikasjonene for URI, er det lite som foreløpig er realisert på dette området. Med den stadig økende bruken av Internett som informasjonsformidler har URN fått stadig større aktualitet, og det er i dag stor interesse for løsninger på dette området.

Både URL og URN er undergrupper av URI. Dette betyr at vi like gjerne kan kalle en URL for en URI, eller vi kan kalle en URN for en URI. URI er bare en generell fellesbetegnelse for identifikatorer, mens URL og URN sier noe om hva slags type identifikatorer det er. De mer formelle definisjonene på hva som menes med henholdsvis URI, URL og URN finner vi i [15]:

- **URI (Uniform Resource Identifier)** er en kompakt streng av tegn som identifiserer en abstrakt eller fysisk ressurs. En URI kan videre klassifiseres som en lokator eller et navn, eller begge deler.
- **URL (Uniform Resource Locator)** refererer til det subset av URI som identifiserer ressurser via en representasjon av deres primære aksessmekanisme (for eksempel nettverkslokalisering), i stedet for ved hjelp av ressursens navn eller andre attributter ved ressursen.
- **URN (Uniform Resource Name)** refererer til det subsett av URI som kreves å være globalt unike og varige, selv om ressursen slutter å eksistere eller ikke lenger er tilgjengelig.

URI

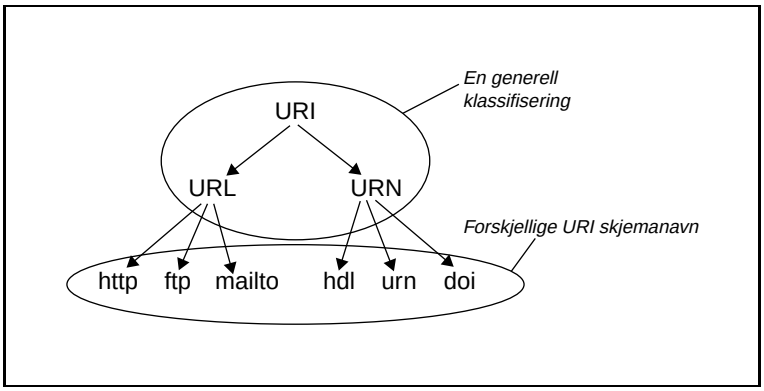
URI-spesifikasjonene er ment som en overbyggende standard for alle former for URI, enten dette er URL eller URN. Spesifikasjonene definerer en syntaks som skal være felles. Dette er først og fremst hvilke komponenter en URI består av og hvilke tegn som er lovlige i en URI. En URI er todelt med skjemanavn og skjemaspesifikk del adskilt av kolon som vist i fig. 5.6. I figuren er *ftp*, *http* og *mailto* navn på forskjellige skjemaer, og det som kommer etter kolon er en skjemaspesifikk streng. Skjemanavn er ofte basert på protokollnavn, f.eks. http og ftp, men dette er ikke en allmenngyldig regel i URI. Det finnes allerede mange etablerte URI skjemaer, og organisasjonen som registrerer nye skjemanavn og

Syntaks:
<skjema>:<skjema-spesifikk-streng>

Eksempler:
http://www.bibsys.no/index.html
ftp://ftp.sunet.se/pub/www
news:comp.os.linux.announce
mailto:Trond.Aalberg@idi.ntnu.no

Figur 5.6: URI

sørger for at disse er unike, er IANA (Internet Assigned Numbers Authority). IANA har to registre for henholdsvis URI og URL [80, 78]. W3C har en mer omfattende, men uformell liste, som også inneholder skjemanavn som ikke er registrert av IANA [170].
URI-spesifikasjonen inneholder regler for hvilke tegn som er tillatte og hvordan man kan representere andre tegn ved hjelp av deres heksadesimale verdi, men definerer ingen andre krav til den skjemaspesifikke delen. URI krever ingen generell struktur eller betydning felles for alle URI, men en del av disse skjemaene deler likevel en felles syntaks bl.a. annet for å representere hierarkier f.eks. filkataloger. Målsettingen er at alle skjemaer i fremtiden skal defineres i egne spesifikasjoner, mens URI kun definerer den universelle syntaksen de forskjellige skjemaer deler.



Figur 5.7: Forholdet mellom URI, URN, URL og de forskjellige URI-skjemanavnene

URL

Den dominerende måten for å identifisere ressurser på World Wide Web, er og har hele tiden vært, ved hjelp av URL (Uniform Resource Locator). En URL er en subtype av URI hvor vi identifiserer og navngir en ressurs ved hjelp av lokaliseringsinformasjon eller ressursens adresse. I dette ligger det at en URL bare er en spesiell type URI. Det er derfor like riktig å omtale en web-adresse som URI, som det er å kalle den for URL, men når vi bruker termen URL indikerer vi at ressursen identifiseres ved hjelp av dens lokalisering.

URL ble opprinnelig definert i en egen RFC [16], men utviklingen går mot å ha de generelle reglene som en del av URI-spesifikasjonen og de skjemaspesifikke reglene dokumentert i egne spesifikasjoner. Med andre ord vil det ikke finnes noen generell spesifikasjon for URL. Dette er allerede innført for noen URL-typer, f.eks. finner vi den skjemaspesifikke syntaksen for HTTP-URL definert i HTTP-spesifikasjonen [56], som vist i fig. 5.8.

Syntaks:
http_URL = "http": "//" host [":" port] [abs_path "?" query]

Eksempler:
http://www.bibsys.no/index.html
http://wgate.bibsys.no/gate1/FIND?bd=metadata

Figur 5.8: URL-syntaks for HTTP-adresser.
En URL for HTTP starter med strengen "http://", deretter følger en tjeneradresse, evt. med et portnummer, "abs-path" er stien til ressursen som adresseres, og til slutt kommer et evt. søkeuttrykk.

URN

Helt siden World Wide Web ble introdusert har det vært erkjent at lokatorbaserte identifikatorer var utilstrekkelig for mange bruksområder. Dette førte til at URN ble introdusert på et tidlig stadium av World Wide Web. Vi har innledningsvis omtalt URN som en generell type identifikatorer som identifiserer en ressurs ved help av et globalt unikt og varig navn. Dette er egentlig bare en side av URN, fordi begrepet er brukt på to forskjellige måter [35]:

- På den ene siden er URN brukt som generell betegnelse for en type URI, de som er globalt unike og varige og som har en organisasjon som er ansvarlig for disse egenskapene. Dette er den måten URN-begrepet har vært brukt i de tidligere dokumentene som beskriver URI, URL og URN.
- Den andre siden ved URN er et rammeverk for URN som en IETF-arbeidsgruppe jobber med [171]. Spesifikasjonene for dette er kommet

rimelig langt, og det er publisert både informative funksjonelle krav og et standardiseringsforslag for syntaks. I dette rammeverket er det spesifisert at URN skal ha en egen syntaks og ta i bruk det felles skjemanavnet *urn*.

Disse to retningene og bruken av URN-begrepet kan virke noe forvirrende. URN-begrepet ble innledningsvis introdusert som en generell betegnelse for en subtype av URI. I arbeidet med URN har man bestemt seg for et felles skjema og rammeverk for Uniform Resource Names under URI skjemanavn "urn" (se fig. 5.7). Når vi i tillegg ser at det også er forvirring rundt bruken av URI kontra URL, er det naturlig å stille spørsmål om det ikke er behov for en forenkling av typologien på dette området.

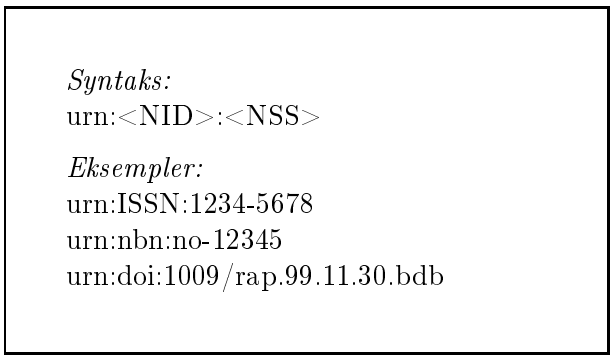
Funksjonelle krav til URN
En rekke dokumenter beskriver og spesifiserer URN. Det ble allerede i 1994 utarbeidet en informativ RFC [158] som både definerte URN og beskrev krav til funksjonelle egenskaper ved URN:

- **Global dekning**
  En URN er et navn med global dekning som ikke antyder noen lokalisering av ressursen. Den har samme betydning over alt.
- **Global unikhet**
  Samme URN må aldri bli tilordnet to forskjellige ressurser.
- **Varighet**
  Det er intensjonen at livstiden til en URN skal være uendelig. Med dette menes at en URN skal være globalt unik i evig tid, og at den skal kunne brukes som en referanse til en ressurs ut over livstiden til ressursen selv eller til navneautoriteten som tilordnet navnet.
- **Skalerbar**
  Alle ressurser som blir tilgjengelig på nett skal kunne tilordnes en URN, flere hundre år fremover.
- **Støtte for eksisterende navnesystemer**
  URN skjemaet må kunne støtte eksisterende navnesystemer så lenge de tilfredstiller de øvrige krav til URN.
- **Utvidbarhet**
  Alle skjemaer for URN må tillate fremtidige utvidelser til skjemaet.
- **Uavhengighet**
  Det er den autoritet som oppretter navn som avgjør hvilke betingelser som skal gjelde for å opprette/tilordne et navn.
- **Oversetting**
  En URN vil ikke forhindre oversetting av navn til lokator. Mer spesifisert betyr dette at for en URN som har en korresponderende URL må det være en passende mekanisme for å oversette en URN til en URL.

IETF arbeidsgruppe for URN
IETF har etablert en egen arbeidsgruppe som skal koordinere og initiere utviklingen av URN konformt med de funksjonelle kravene i [158]. Denne arbeidsgruppen har hatt som mål å utvikle et rammeverk for Uniform Resource Names (URN) og et sett av komponenter som passer i dette rammeverket [171].

Resultatene fra denne arbeidsgruppen er et sett av RFC-dokumenter som definerer forskjellige aspekter ved URN. De mest relevante av disse er en spesifikasjon for URN-syntaks [129], en arkitektur for URN-resolution [157], og en mekanisme for definering og registrering av URN-navnerom [38].

URN syntaks
En URN består av prefikset "urn:" etterfulgt av en navneromsidentifikator (NID), et kolon og en navneromsspesifikk streng (NSS) [129] (fig. 5.9).



Figur 5.9: URN

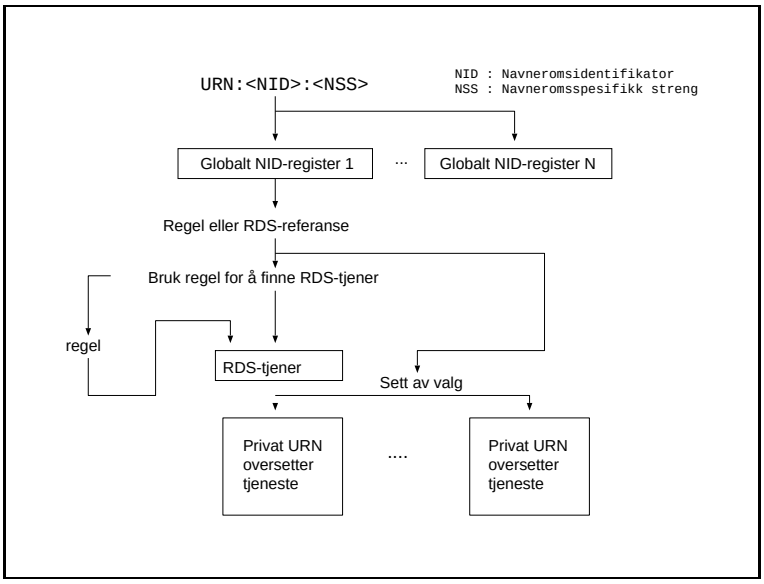
En navneromsidentifikator (NID) identifiserer den navneromsspesifikke strengen for en bruker eller et system. Eksisterende identifikatorsystemer som ISBN og ISSN er eksempler på aktuelle NID. URN-syntaksen definerer hvilke tegn det er tillatt å benytte i en NID, som er mer restriktiv syntaks enn den generelle syntaksen i URI. Den navneromsspesifikke delen av en URN (NSS) inneholder selve identifikatorstrengen, for eksempel selve ISBN. Syntaks for NSS er den samme som URI generell syntaks.

Rammeverk for URN-oversetting
Selv om en URN identifiserer et logisk informasjonsobjekt, er det ofte behov for å kunne finne frem til et eksemplar av informasjonsobjektet som identifiseres. På Internett vil dette være å kunne gå fra en lokaliseringsuavhengig og varig URN til en URL. Denne formen for tjeneste kaller vi oversettelse (eng. Resolution Service – RS) fordi den oversetter fra en identifikator til en lokator[2]. Det er allerede nå tydelig at det vil finnes mange forskjellige navnerom (NID)

[2]På engelsk brukes termen resolution – "oppløsning", men dette er et ord som ikke fungerer like godt på norsk. Vi har derfor brukt "oversetting".

med tilhørende oversettersystemer, og det vil være behov for en overbyggende arkitektur for å kunne velge oversettertjeneste som støtter det navnerom (NID) en identifikator tilhører. En slik arkitektur er beskrevet i [157]. URL som det oversettes til, trenger ikke nødvendigvis å vise til informasjonsobjektet direkte, men kan også vise til metadata for informasjonsobjektet, eller en påloggings-side o. l.

I grove trekk er denne arkitekturen basert på en "oversetter-oppdagelsestjeneste" (eng. Resolution Discovery Service - RDS) som inneholder et globalt NID-register med informasjon om hvilke oversettertjenester som støtter hvilke NID. Denne tjenesten returnerer en referanse til en oversettertjeneste, slik at klient er i stand til å kontakte riktig tjeneste for denne spesifikke URN. Dette er en fleksibel mekanisme som ikke impliserer at det skal være et 1:1 forhold mellom navnerom og oversettertjeneste, eller 1:1 forhold mellom URN og URL. Oversetting kan like gjerne resultere i metadata om informasjonsobjektet som en URL til et eksemplar tilgjengelig på nett, og vi kan også se for oss at en URN kan oversettes av flere forskjellige oversettertjenester.



Figur 5.10: RDS

Definering og registrering av URN navnerom

I URN er det stor fleksibilitet for definering av nye navnerom. De mer formelle forhold rundt dette er beskrevet i [38]. Navneromsidentifikatorer tilordnes eller godkjennes/registreres av IANA.

Noen kritiske bemerkninger til URN

Utviklingen av URI og URN drives av forskjellige IETF-arbeidsgrupper, og spesifikasjoner utvikles separat og uavhengig av hverandre i tid. URN-syntaksen

er eksempelvis utviklet i 1997, mens den siste URI-spesifikasjonen er utviklet i 1998, noe som blant annet har resultert i uoverensstemmelser i terminologien.

Det kan være vanskelig å se hvordan de forskjellige komponenter av en URN er relatert til de komponenter en URI består av. En naturlig tolkning er å se prefikset "urn:" som et URI-skjemanavn med det etterfølgende skilletegn, og den resterende delen av en URN som det URI-spesifikasjonen definerer som skjemaspesifikk streng.

Vi finner også enkelte uoverensstemmelser i syntaks. Et URI-skjemanavn skal være i små bokstaver, mens prefikset "urn:" er definert som case-insensitive, noe som ikke vil være i overensstemmelse med URI[3].

Denne mangelen på relasjon og presis konformitet til URI-spesifikasjonene er en ulempe så lenge begge spesifikasjoner berører hvordan identifikatorer skal kodes og parses av programvare som skal ha automatisert håndtering av identifikatorer. Mangel på konformitet vil forvanske innføringen av URN og støtte for URN i programvare. Et mer logisk utgangspunkt hadde vært å definert en URN-syntaksen i forhold til URI-syntaksen og presisert relasjonene mellom disse to spesifikasjonene.

Det er også vanskelig å tolke hvordan identifikatorsystemer som the Handle System og DOI skal plasseres i dette landskapet. Pr. i dag brukes *hdl* og *doi* som egne URI-skjemaer, men de er også brukt som eksempler på navnerom i URN. De vil sannsynligvis kunne eksistere som begge deler.

NBN - NID for nasjonalbibliotek

For å stimulere til bruk av URN, og for å tilby varige identifikatorer i overensstemmelse med URN-syntaksen, er det opprettet en egen navneromsidentifikator (NID) og tjeneste for automatisk generering av URN fra de nordiske nasjonalbibliotekene [140, 68]. Navneromsidentifikatoren[4] er *NBN* som står for *National Bibliographic Number*. Selve identifikatorstrengen prefikses av en landkode (se eksempel i fig. 5.11).

NBN-baserte URN er foreløpig kun et unikt nummer, og har ingen tjenester eller informasjon knyttet til seg. Overordnete retningslinjer for tilordning og bruk av slike identifikatorer og overordnet administrativ instans for dette navnerommet er beskrevet i [68]. Hovedmålet med denne identifikatoren er foreløpig kun å tilby unike navn som f.eks. kan legges i dokumenter for å navngi dette med en unik identifikator.

5.6.5 PURL og virtuelle URL

PURL er en tilnærming til URN som er utviklet av OCLC [154]. Problemet PURL adresserer er at de URL som brukes i web-dokumentene ikke er stabile eller varige. Et dokument på en web-tjener får en ny URL hvis dokumentet

[3]Dette er reelt sett ikke et problem siden URI-spesifikasjonene også sier at skjemanavn med store bokstaver skal håndteres som små, for å sikre bakover-kompatibilitet

[4]Denne navneromsidentifikatoren er derimot enda ikke registrert ved IANA [79].

Generell syntaks:
urn:NBN:<landkode>-<nummer>

Eksempel på norsk URN:
urn:NBN:no-123456

Eksempel på finsk URN:
urn:NBN:fi-fe976238

Figur 5.11: NBN

skifter plass i filsystemet eller flyttes til en annen maskin. Stabile og varige URL kan likevel oppnås ved å bruke en ekstra URL som mellomledd – en *virtuell URL*. Dennes fungerer som en lokaliseringsuavhengig identifikator, selv om den er basert på adressering.

Dette er en teknikk som gir fleksibilitet. Web-dokumentene kan adresseres med et lokatorskjema som web-lesere støtter (f.eks. http:), men det er likevel mulig å endre den faktiske adressen uten at dette har effekt for alle eksterne lenker som finnes til dokumentet. En slik løsning forutsetter selvsagt at det finnes informasjon og en tjeneste som kan brukes for å oversette fra virtuell URL til faktisk URL. En PURL forvaltes av en ansvarlig organisasjon, og dette er en viktig forutsetning for at et slikt system skal kunne fungere over tid – det sikrer at denne informasjonen taes vare på. Selv om den faktiske adressen oppdateres, forblir virtuell URL den samme.

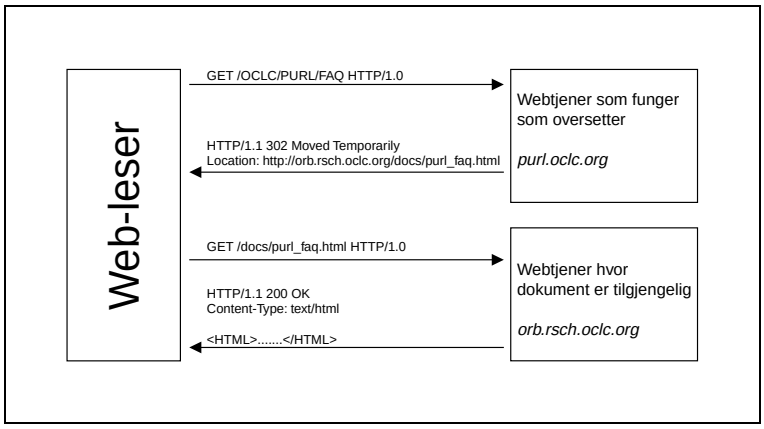
Oversetting fra virtuell URL til faktisk URL kan implementeres på flere måter. PURL er basert på den støtte for videreformidling av forespørsler som er del av HTTP standarden [56]. Når en bruker aktiverer en PURL-lenke i en web-leser, sendes det en forespørsel om dokumentet til en PURL-oversettertjener (en web-tjener). PURL-tjeneren returnerer en responsmelding som forteller hvor denne ressursen finnes[5]. Se eksempel i figur 5.12 for bruk av følgende PURL:.

`http://purl.oclc.org/OCLC/PURL/FAQ`

hvor "purl.oclc.org" er en web-tjener som fungerer som en oversetter-tjener og sender tilbake en omdirigeringsrespons, mens stidelen er det logiske navnet til dokumentet. I tilfellet over oversettes dette til den faktiske URL:

`http://orb.rsch.oclc.org/docs/purl_faq.html`

[5]Dette skjer ved hjelp av statuskoden 302 "Moved Temporarily" og meldingsoverskriften "Location" som inneholder den faktiske URL.



Figur 5.12: Sekvensen av HTTP meldinger for aksess til et dokument via PURL

PURL er ikke konform med URN, men kan likevel karakteriseres som delvis kompatibel. En PURL vil inneholde nok informasjon for en eventuell senere maskinell konvertering til URN-skjemaet. Dette kan derfor være en aktuelle mellomløsning når behovet primært er lokaliseringsuavhengige web-adresser.

PURL-systemet er fritt tilgjengelig programvare, og er basert på en tradisjonell web-tjener og en database som brukes for å oversette en PURL til en URL.

Fordelen med en slik løsning er at den er basert på teknologi som er i bruk og som er kompatibel med den måten vi allerede uttrykker identifikatorer på. Ulempen er at en PURL likevel ikke vil være mer varig enn domemenavnet til den maskinen som skal oversette mellom virtuell URL og faktisk URL.

Essensen i PURL er bruken av en et mellomledd eller virtuell URL. Dette gjør at den URL som publiseres ikke lenger er lokaliseringsinformasjon, men et lokaliseringsavhengig navn med URL-syntaks. Det er også mulig å implementere former for videresending eller oversetting av virtuelle URL på tjenersiden[6]. BIBSYS har brukt denne teknikken i det som kalles BIBSYS-hendler. Dette er et lokalt system BIBSYS benytter for å oppnå lokaliseringsuavhengig navngiving av nettdokumenter, og systemet er brukt på digitaliserte DKNVS skrifter.

5.6.6 The Handle System

The Handle System er et identifikatorsystem utviklet av CNRI [161, 162]. En hendel[7] defineres som et unikt navn for et digitalt objekt eller andre Internettressurser.

Utgangspunkt for The Handle System finner vi i et prosjekt for å utvikle et rammeverk for den underliggende infrastrukturen i digitale bibliotek (se kap. 4.1). Den første implementasjon ble gjort av CNRI i 1994, og systemet

[6] Enkelte web-tjenere som Apache har slik funksjonalitet implementert.
[7] Vi har oversatt "handle" til "hendel".

er i dag tatt i bruk av flere større aktører som Library of Congress og DOI Foundation.

The Handle System er et distribuert informasjonssystem utviklet for å gi effektive, utvidbare og sikre navnetjenester i et nettverk. The Handle System består av en åpen protokoll for å administrere, registrere og oversette hendler. Til en hendel kan det være knyttet en eller flere URL, eller andre former for lokatorer for en ressurs.

Oppbygning
The Handle System spesifiserer en generell syntaks for en hendel som er uavhengig av de systemene som bruker en hendel. En hendel er todelt og består av *navneautoritet* og *lokalt navn*, også kalt *prefiks* og *suffiks*.

- **Navneautoritet** identifiserer den administrative enheten for en hendel. En slik navneautoritet er globalt unik og varig når den først er opprettet. For navneautoriteter har tegnene "/" og " ." spesiell betydning. Tegnet "/ " brukes til å skille mellom navneautoritiet og lokalt navn, mens tegnet " ." skiller mellom segmenter i en navneautoritet. Navneautoriteter kan være hierarkisk oppbygd, men når de er etablert er en navneautoritet å regne for en samlet streng.

- **Lokalt navn** er en unik identifikator under en navneautoritet. Det er få syntaksregler for lokale navn, og alle utskrivbare tegn fra Unicode kan brukes.

The Handle System er et generelt navnesystem, men på Internett er det mest aktuelle å bruke hendler som URI. Dette kan gjøres på to måter. Enten ved å bruke et eget URI-skjemanavn som er spesifikt for The Handle System, eller ved å bruke URN og en egen navneromsidentifikator for The Handle System (se fig. 5.13). Uttrykt som URI eller URN må tegn ut over ASCII (samt noen spesialtegn) kodes som heksadesimale verdier.

Administrering av hendler og navneautoriteter
The Handle System er utviklet for å kunne tilfredstille en informasjonsverden med mange administrative enheter. Dette gjenspeiler seg i en todeling av navnene i lokale navn og navneautoriteter. Få begrensinger i syntaks for lokale navn gir stor fleksibilitet for innpassing av eksisterende navn i systemet.

Navneautoritetene er oppbygd i et hierarki, hvor oppretting av subautoriteter er et distribuert ansvar. En navneautoritet har selv ansvar og rett til å opprette navn uten andre krav enn de generelle reglene for syntaks. Denne modellen for navngiving gir også lokale enheter stor frihet i tilordning av navn til ressurser.

Til The Handle System er det også knyttet et administrativt system. Hver hendel kan ha egne administratorer eller administratorgrupper. Sammen med en autentifiseringsprotokoll gjør dette at hendler kan administreres på en dis-

Hendel
Generell syntaks:
<Hendel navne autoritet>/<Hendel lokalt navn>
Eksempel:
1009/rap.99.11.30.bdb

Hendel uttrykt som URI, URI skjemanavn = "hdl"
Syntaks hdl:<Hendel navne autoritet>/<Hendel lokalt navn>
Eksempel:
hdl:1009/rap.99.11.30.bdb

Hendel uttrykt som URN, NID = "hdl"
Syntaks
urn:hdl:<Hendel navne autoritet>/<Hendel lokalt navn>
Eksempel:
urn:hdl:1009/rap.99.11.30.bdb

Figur 5.13: The Handle System

tribuert måte av autoriserte administratorer, noe som er spesielt godt egnet for distribuerte miljøer.

CNRI har utviklet programvare som implementerer the Handle System[8]. Dette er fritt tilgjengelig programvare, både en Java-basert tjener for oversetting, programvarebiblioteker som forenkler utvikling av systemer som bruker hendler, en plugin for Netscape og Microsofts web-lesere og en egen web-leser Grail [33] som kan håndtere hendler kodet som URI-skjemaet "hdl".

En spesifikasjon for et URI-skjema for the Handle System er under utarbeidelse, men selv om URI-skjemaet "hdl" er tatt i bruk, er det enda ikke formalisert [161].

5.6.7 DOI Digital Object Identifier

DOI [145] er et identifikatorsystem som ble initiert av Association of American Publishers (AAP). Utgangspunktet var et behov for bedre muligheter for, og kontroll over, nettbasert handel med åndsverk. AAP opprettet en "Enabling Technologies Committee" i 1994, og målsettingen var å standardisere på et sett av teknologier for identifikasjon, beskrivelse, adgangsrettigheter, presentasjonsformater, beskyttelsesmekanismer og betalingsinformasjon [153]. De fleste

[8] Se: http://www.handle.net

av disse målene ble lagt til side eller vurdert som ivaretatt av eksisterende løsninger, og komiteen valgte å fokusere på identifikasjon. Dette ble vurdert som det mest fundamentale behovet, men dette valget var også strategisk fordi en felles standard for identifikasjon kunne fungere som kime til samarbeid også på andre områdene. Krav som ble stilt til identifikatorer av AAP var:

- Identifikatoren skal være så "dum" som mulig, uten noen innebygd mening.
- En identifikator skal unikt identifisere innhold, og det skal ikke være duplisering.
- Det skal kunne være et uendelig antall identifikatorer.
- Identifikatorsystemet skal kunne fungere som et metaskjema for allerede eksisterende navnesystemer som ISBN og ISSN.
- Identifikatoren skal kunne brukes på alle medier.
- Identifikatoren skal kunne brukes uavhengig av granularitet, både på flervolums bind og på enkeltavsnitt og illustrasjoner. Avgjørelse vedrørende granularitet overlates til utgiver.
- En identifikator skal referere til innhold uavhengig av lokalisering og eierforhold, selv om en av disse blir overført til andre utgivere.

Det identifikatorskjema som ble utviklet fikk navnet *Digital Object Identifier*, og er i overensstemmelse med de funksjonelle kravene til URN. The Handle System ble valgt som teknologisk plattform, og en prototyp ble utviklet og demonstrert i 1997.

DOI har utviklet seg ut over AAP's område, og i 1997 ble *The International DOI Foundation* opprettet for å overta den videre utvikling av DOI. Dette er en ikke-profitt organisasjon basert på medlemskap.

DOI-identifikatoren

Fordi DOI er basert på the Handle System, har DOI derfor samme strukturelle oppbygning og regler for tegn som andre hendler. DOI-syntaksen er fremmet som forslag til ANSI/NISO-standard Z39.84 [138]. Tilsvarende som for en hendel består en doi av et *prefiks* og et *suffiks*. *Prefiks* for en DOI er todelt og består av *Directory Code* og *Registrant Code* separert med et punktum. Directory code er toppnivå navneautoritet, og er knyttet til et *directory* som er en *hendel-tjener*. Foreløpig er bare "directory code" 10 etablert og tatt i bruk[9], men det er ingen hindringer for å etablere nye Directory Code når det oppstår behov for det. Registrant Code er et nummer (subautoritet) som tilordnes utgivere. Suffiks er en streng som utgivere tilordner sine objekter.

[9] Dette er i realiteten en navneautoritet under The Handle System.

DOI
Generell syntaks:
<Prefiks>/<Suffiks>
<DIR>.<REG>/<DSS>
Eksempel:
10.1006/rwei.1999.0001

DOI uttrykt som URI, URI skjemanavn = doi"
Syntaks:
doi:<Prefiks>/<Suffiks>
Eksempel:
doi:10.1006/rwei.1999.0001

DOI uttrykt som URN, NID = doi"
Syntaks
urn:doi:<Prefiks>/<Suffiks>
Eksempel:
urn:doi:10.1006/rwei.1999.0001

Figur 5.14: DOI syntaks

DOI-katalogen
DOI er mer enn bare en identifikator, DOI er også et system for å oversette en DOI til en lokator. Siden DOI er basert på The Handle System, er funksjonaliteten knyttet til en DOI tilsvarende som for en hendel. Oversettertjenesten fungerer som en videreformidler mellom bruker og utgiver. I enkelte tilfeller oversettes en DOI direkte til et dokument tilgjengelig på web, i andre tilfeller oversettes en DOI til en web-side som inneholder informasjon om dokumentet, utfører adgangskontroll og lignende. Et minimumskrav i DOI-systemet er at en DOI skal oversettes til en "respons-side" som gir brukeren relevant informasjon.

Oversetting mellom DOI og det en DOI oversettes til, er transparent for brukeren. Den samme plugin for web-leseren kan brukes både for URI-skjemaene *hdl* og *doi*, og en proxy-server for HTTP-basert bruk av DOI er også tilgjengelige.

DOI-metadata
DOI Foundation fokuserer også på metadata og har som mål at det også skal være metadata assosiert til en DOI. For å definere struktur og form på de nødvendige metadata, samarbeider DOI Foundation med prosjektet *The Inte-*

roperability of Data in E Commerce Systems (INDECS). INDECS er et forsøk på å samordne flere initiativ for å komme frem til en felles metadatamodell relatert til handel [147]. Det er anbefalt at ingen DOI skal registreres uten at også metadata registreres.

5.6.8 DNS (Domain Name Service)

DNS er en navnetjeneste som er i bruk over hele Internett [130, 131, 4]. Dette er en distribuert database med informasjon om domenenavn. Informasjonen er ikke bare lagret på en maskin, men er fordelt i et nett av samarbeidende maskiner. Disse er organisert i et hierarki, og når vi sender en forespørsel til en DNS-tjener vil denne sende forespørselen videre til andre tjenere hvis informasjonen om domenenavnet ikke finnes lokalt.

Når vi bruker Internett bruker vi så å si alltid DNS i bakgrunnen. De fleste adresser på World Wide Web (og resten av Internett) er basert på domenenavn som "www.bibsys.no" eller "ftp.nvg.ntnu.no". Web-lesere og annen programvare som bruker Internett, er derimot avhengige av å kjenne IP-adressen – det globalt unike nummeret som er den egentlige adressen til nettverk og maskin. For å finne denne adressen foretar programmet et oppslag i DNS-systemet som oversetter fra domenenavn til IP-adresse. En DNS-tjener kan også lagre annen informasjon assosiert med domenenavnet som informasjon om prosessor og operativsystem. DNS har også en viktig funksjon for email ved at en DNS inneholder informasjon om hvilken server i et nettverk som skal motta inngående epost - hvilket system som er "lokalt postkontor".

Objekter som er navngitt i DNS er primært datamaskiner som er representert ved domenenavn. En ressurs-post (resource record) inneholder de data som er assosiert med et domenenavn. Det finnes flere klasser av slike poster relatert til nettverkstype eller programvare. Den dominerende klassen er innlysende nok for Internett. Innenfor hver klasse er det forskjellige poster beregnet på de forskjellige typene av data som kan lagres i domenets navnerom. De fleste klasser har en datatype kalt adresse, som inneholder IP-adressen som er assosiert med domenenavnet.

DNS er et etablert system for oversetting av domenenavn, men i kraft av at dette er et etablert system som også kan brukes til å gjøre oppslag også for andre typer av informasjon, er dette systemet også testet ut for URN oversetting og som RDS-tjeneste [39].

Kapittel 6

Relasjoner og lenker

6.1 Relasjoner

Informasjonsobjekter eksisterer ikke uavhengig av hverandre eller sine omgivelser. Hvordan informasjonsobjekter er relatert til hverandre og til resten av verden, er opplysninger vi har bruk for både som brukere og som forvaltere av informasjon.

Bibliografiske metadataformater fanger opp en del av konteksten som omgir et informasjonsobjekt. Attributter som forfatter og korporasjonsnavn relaterer dokumenter til personer og virksomheter, og emneord og klassifikasjonsnummer er attributter som brukes for å relatere informasjonsobjekter tematisk. Det er likevel begrensninger på hva det er realistisk å legge i metadata, og selv de rikeste metadata-formater kan vanskelig fange opp alle relasjoner det er et mulig behov for.

Vi finner også relasjoner uttrykt i selve innholdet i informasjonsobjektene. I litteratur er det en etablert tradisjon for referanser, henvisninger og litteraturlister som en del av teksten, og de forskjellige temaer som behandles i en tekst kan relateres til andre deler av teksten eller andre tekster.

I informasjonsgjenfinning er relasjoner av spesiell betydning fordi de gjør det mulig med andre informasjonsgjenfinnings-teknikker enn den tradisjonelle hvor brukerne formulerer et søkeuttrykk og får returnert en treffliste.

I administrering av informasjon har vi også et stort behov for å kjenne slektskapet mellom informasjonsobjektene. Et metadata-format beskriver dokumenter, og henviser enten spesifikt til eksemplarer eller generelt til informasjonsobjektet uavhengig av eksemplar. Vi har også behov for relasjoner fra informasjonsobjektene til metadata. For mange informasjonsobjekter kan det finnes forskjellige former for metadata lagret i mange systemer. For å kunne gjenbruke disse metadata på tvers av systemer, er det behov for å relatere informasjonsobjektene til metadatapostene.

En generell tendens er at koblingen mellom informasjonsobjekter i ett og samme informasjonssystem under en virksomhets organisering, kan være god

fordi denne virksomheten kjenner informasjonsmengden og enkelt kan opprette og administrere relasjoner mellom informasjonsobjektene. For kobling mellom informasjonsobjekter på tvers av virksomheter og systemer er det derimot mange problemer knyttet til det å identifisere, administrere og uttrykke relasjoner.

Terminologi for relasjoner
Relasjoner er et sentralt element i data- og informasjonsmodellering, og det finnes mange formelle notasjoner for å modellere og beskrive relasjoner. ER-modeller (Entity-Relationship) er en mye brukt modelleringsmetode for databaser hvor verden beskrives som entiteter og relasjoner. I den opprinnelige artikkelen av Cheng som beskriver ER-modellen [31], finner vi denne definisjonen av hva en relasjon er:

> A relationship is an association among entities

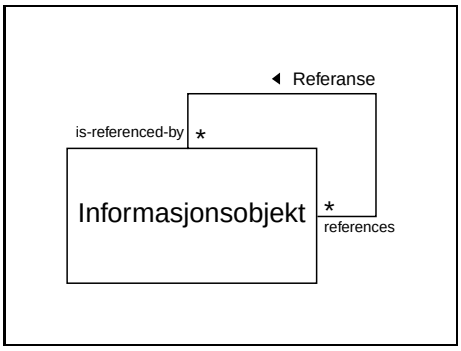
I ER-modellen er en entitet en enhet som distinkt kan identifiseres, mens en relasjon er en assosiasjon mellom entiteter. Entiteter er enheter som bok, person og forlag, mens relasjoner er assosiasjoner mellom disse, som "bok *er skrevet av* person", eller "forlag *gir ut* bok". Relasjoner er ikke noe som er spesifikt for dette modelleringsspråket, men et element vi finner igjen i alle modelleringsspråk.

En relasjon har ofte *retning*, og den kan være *enveis* eller *toveis*. I uttrykket "bok *er-skrevet-av* person" bruker vi en relasjon som peker fra bok til person, men vi kan også gå den andre veien ved å si "person *har-skrevet* bok". Relasjoner som bare peker fra en enhet til en annen er enveis, mens en relasjon som peker begge veier kalles toveis.

En relasjon kan være mellom to enheter, og da sier vi at den er *binær*. Relasjoner mellom flere enn to enheter kalles *multinære* (N-ary).

I modellering av relasjoner er det ofte også behov for å si noe om mengdeforholdene i en relasjon – *kardinalitet*. En forfatter kan ha skrevet mange bøker, og en bok kan ha mange forfattere, er et eksempel på et mengdeforhold hvor det er n antall enheter som kan inngå på begge sider av relasjonen. Sier vi derimot at et forlag gir ut mange bøker, mens en bok bare er gitt ut av ett forlag, har vi et mange-til-en forhold mellom enhetene som inngår i relasjonen. En annen formell beskrivelser av relasjoner er *deltagelse* – om en enhet må inngå i en relasjon eller om enheten kan eksistere i systemet uten at den deltar i en relasjon. Eksempler på dette er hvis vi sier at en bok bestandig har en forfatter, mens en forfatter ikke trenger å ha skrevet en eneste bok. Disse aspektene er viktige i modellering av data i databaser, fordi de gir oss opplysninger som er relevante for hvordan vi bør utforme selve databasen, men mindre viktige for relasjoner mellom distribuerte informasjonsobjekter, fordi det der er vanskelig å stille krav til mengdeforhold og deltagelse i en relasjon.

Hvis en relasjon er mellom to entiteter av samme type, kalles dette for en *refleksiv relasjon*. Et eksempel på dette er en artikkel som refererer til en annen artikkel. I fig. 6.1 har vi modellert en refleksiv relasjon kalt referanse, som også viser de andre aspektene som er diskutert.



Figur 6.1: En relasjon mellom to informasjonsobjekter uttrykt i UML
Et informasjonsobjekt (f.eks. en artikkel) refererer til et annet informasjonsobjekt, og pilen viser retningen på relasjonen. Rollene "references" og "isreferencedby" er hentet fra Dublin Core-kvalifikatorene, og forteller hvilken rolle de to artiklene har i relasjonen. I dette tilfellet kan et informasjonsobjekt referere til 0 eller flere informasjonsobjekter, og tilsvarende kan et informasjonsobjekt bli referert til av 0 eller flere informasjonsobjekter (dette uttrykkes ved bruk av stjernetegnet "".)*

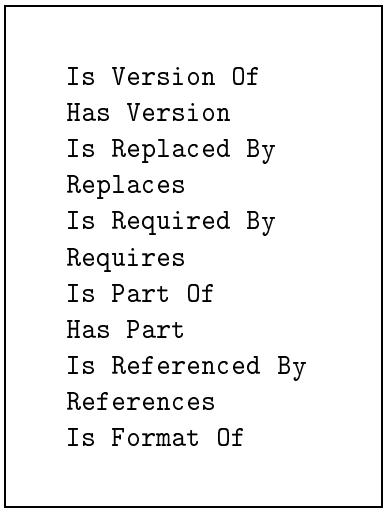
Relasjonstypologi
Relasjoner kan klassifiseres i tre allmenngyldige hovedtyper:

- **Aggregering** er å sette sammen deler til en større helhet. Dette forholdet mellom en helhet og de forskjellige deler av helheten er en karakteristisk type relasjon som ofte kalles *is-part-of* eller partitive relasjoner. Et kapittel er en del av en bok. Et verk kan bestå av mange bind. Aggregering er en relasjonstype som gir oss mulighet til å spesifisere strukturelle relasjoner, noe det er behov for fordi informasjon ofte organiseres som deler av større enheter. Strukturelle relasjoner kan være basert både på fysisk struktur og logisk struktur. Fysisk struktur er spesielt relevant for digital informasjon, fordi dette mediet ofte fragmenterer informasjonen basert på filformater. Et informasjonsobjekt kan være satt sammen av mange forskjellige filer av forskjellig type; teksten kan være en html-fil, mens bildene er i bildefiler (se fig. 2.2).

- **Subtyping** er en annen hovedtype relasjon som uttrykker et typologisk forhold mellom enheter. En katt er et rovdyr, og en mus er en gnager. Dette er relasjoner som sier noe om slektskapet mellom enheter. Det å kunne klassifisere omgivelsene og operere med typologiske slektskap, er en viktig del av den menneskelige abstraheringsevnen. Slike relasjoner kalles ofte *is-a* eller generiske relasjoner.

- **Assosiering** er en siste hovedtype relasjon som må være med for å fange opp alle de relasjonene som de to foregående kategoriene ikke favner. Dette er ingen spesifikk type relasjon, men heller en helt generell kategori av relasjoner, der vi kan sette våre egne betegnelser på relasjonene, som i "bok *er-produsert-av* forlag" eller "bibliotek *eier* bok". Assosiering kalles også brukerdefinerte relasjoner. Bruk av emneord og klassifikasjonsnummer er en måte å relatere informasjonsobjekter til hverandre, basert på tematisk likhet.Tematiske relasjoner kan kategoriseres som assosiering.

Innenfor spesifikke områder er det videre mulig å spesifisere andre hovedtyper i forhold til hvilke relasjoner det er behov for å uttrykke. I utviklingen av Dublin Core-metadataformatet har man etablert en liste over spesifiserte typer av relasjoner for bruk i relation-attributtet [44, 49]. De fleste av disse er bruksrelaterte assosiasjoner, men *is-part-of* og *has-part* er strukturelle relasjoner tilsvarende aggregering (se fig. 6.2).



Figur 6.2: Relasjonstyper som kan brukes i Dublin Core

6.2 Lenker

Lenker og relasjoner er uttrykk som benyttes om hverandre. På World Wide Web og i andre hypertekstbaserte informasjonsrom benytter vi heller *lenke* enn *relasjon* når vi omtaler de koblingene som finnes mellom enhetene. Vi finner også uttrykket lenke brukt i mange andre datarelaterte sammenhenger. I en terminologi for World Wide Web som er utviklet av en W3C-arbeidsgruppe, finner vi denne definisjonen på hva en lenke er [114]:

> A link expresses one or more (explicit or implicit) relationships between two or more resources.

En lignende definisjon finner vi i HyTime-standarden [96]:

> An information structure that represents a relationship among two or more objects.

I Dexter-modellen [70] defineres lenker ved:

> Links are entities that represent relations between other components.

Denne definisjonen er preget av modellens fremstilling av lenker som en komponent på linje med andre komponenter, f.eks. tekstdokumenter eller bilder.

Selv om vi forlater hypertekstverdenen, finner vi at lenkebegrepet har omtrent samme betydning. I modelleringsspråket UML defineres lenker slik [142]:

> Link is an instance of an Association. ..... A Link defines a connection between Instances.

Lenker og relasjoner er med andre ord to sider av samme sak. En relasjon er en abstrakt enhet, mens lenker er konkret uttrykte relasjoner, tilsvarende forholdet mellom klasse og objekter i objektorientering. Vi kan si at en lenke er en implementert eller instansiert relasjon.

Implisitte og eksplisitte lenker

Lenker assosieres vanligvis med statisk informasjon som er eksplisitt uttrykt i informasjonsobjektene, for eksempel hypertekstlenker i web-dokumenter, men en lenke kan også være dynamisk generert, f.eks. lenking mellom dokumenter som er basert på forekomster av like ord i tekstdokumenter. Mens eksplisitte lenker er uttrykt på forhånd, ofte på en formalisert måte, er implisitte lenker basert på informasjon som i utgangspunktet ikke var ment for lenking. Det å uttrykke lenker eksplisitt for et stort informasjonsrom er en tidkrevende oppgave, og bruken av dynamisk genererte lenker basert på implisitt lenkingsinformasjon kan være et aktuelt alternativ i enkelte situasjoner. Vi kan for eksempel ta utgangspunkt i metadata for en artikkel og på grunnlag av disse dynamisk generere en SICI-identifikator, som så kan brukes som del av adresse til denne artikkelen i en online dokumentbase. På denne måten har vi lenket metadatapostene til et eksemplar av dokumentet, uten at lenkene var eksplisitt uttrykt på forhånd.

Det er en glidende overgang mellom implisitte og eksplisitte lenker. I HTML benyttes et definert element med en gitt struktur for å uttrykke lenken (en datastruktur). Dette gjør at programvaren (web-leseren) enkelt kan identifisere hvilke deler av et HTML-dokument som er en lenke. Tilsvarende er det for XML/XLink og HyTime definert syntaks og semantikk for lenke-elementer. Lenking kan også være basert på andre strukturer, f.eks. har denne rapportens litteraturliste en fast struktur for hver referanse, noe som kunne vært utgangspunkt for lenker. Lenking er med andre ord ikke helt avhengig av en

datastruktur som er utviklet med lenking som formål, men det å basere seg på datastrukturer eller ustrukturerte data kompliserer lenking og gjør lenkingen usikker.

Lenker som del av informasjonsobjektene
I HTML-dokumenter har vi muligheten til å spesifisere en lenke, f.eks.:

```
<a href="http://www.bibsys.no">BIBSYS </a>
```

Denne lenken er en del av dokumentet og peker til en annen ressurs, og dette er den eneste formen for lenking vi har mulighet til i HTML. Lenker som del av informasjonsobjektene (embedded links) er en enkel lenkingsteknikk fordi lenkene ikke gir opphav til nye informasjonsobjekter som skal administreres. Administrativt er dette en effektiv teknikk fordi ansvar for lenkene ligger på de som skaper og vedlikeholder informasjonen. Begrensningene er at dette kun støtter enveis lenker. Vi finner enveis lenker implementert også i andre dokumentformater, f.eks. Microsoft Word og Adobe's PDF-format.

Også i metadataformater kan det være felter og informasjon som fungerer som lenker. I Dublin Core er det definert et generelt felt som til tross for attributtnavnet "relasjon" kan defineres som en lenke.:

```
<meta  name = "DC.Relation.references"
       content = "http://purl.org/dc/documents/rec-dces-19990702.htm">
```

I MARC-formatet er 700-feltene eksempler på felter som kan fungere som lenker til relaterte dokumenter, og felt 856 kan benyttes for adresser og tilgangsinformasjon til nettdokumenter. At 856-feltet kan være en lenke er tydelig, når vi ser at informasjonen i dette feltet benyttes for å generere en HTML-lenke når posten presenteres for brukere i det web-baserte grensesnittet.

Lenker som egne informasjonsobjekter
I tillegg til at lenker som del av dokumenter eller metadata har den ulempen at de er enveis, er det heller ikke mulig å spesifisere relasjoner mellom flere enn to objekter for slike lenker. Toveis lenker kan likevel opprettes hvis vi sørger for at alle informasjonsobjekter som inngår i relasjonen peker til hverandre, men i et miljø av informasjon som er distribuert over mange forskjellige virksomheter og systemer, er det lite realistisk å opprette og vedlikeholde en slik lenkingsstruktur.

En alternativ løsning er å operere med lenkene som selvstendige informasjonsobjekter, en type metadata. Mens HTML har et relativt enkelt språk for å uttrykke lenker, som kun støtter lenker i dokumentene, finner vi vesentlig bedre støtte for lenker som egne objekter i XML/XLink. XLink definerer et format som støtter både enkle lenker tilsvarende det som finnes i HTML, og mer avanserte lenker hvor lenkene kan være eksterne i forhold til de informasjonsobjektene de lenker mellom (se fig. 6.3). RDF kan også fungere som

lenkeobjekter, fordi det er mulig å aggregere ressurser og beskrive den aggregerte enheten. En annen aktuell standard for slike løsninger er HyTime, som grovt sett er overlappende med Xlink når det gjelder lenking.

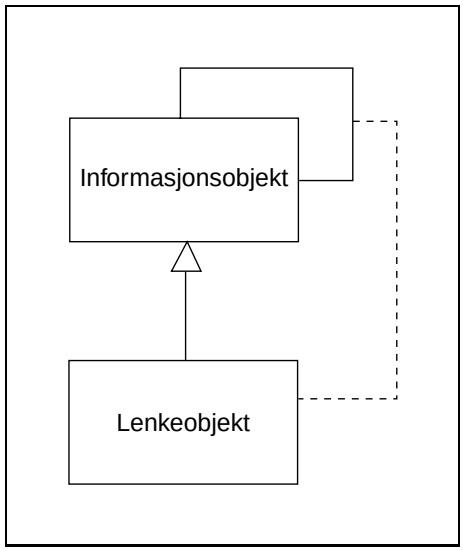
```
<relasjonsobjekt xlink:type="extended" xlink:title="referanser">

 <adresse xlink:type="locator"
   xlink:href="http://artikkel.1.html"
   xlink:label="FraArtikkel
   xlink:role="http://www.example.com/linkprops/References"/>

 <adresse xlink:type="locator"
   xlink:href="http://artikkel.2.html"
       xlink:label="TilArtikkel"
       xlink:role="http://www.example.com/linkprops/isReferenced"/>

      <adresse xlink:type="locator"
   xlink:href="http://artikkel.3.html"
       xlink:label="TilArtikkel"
       xlink:role="http://www.example.com/linkprops/isReferenced"/>

 <retning xlink:type="arc"
       xlink:from="FraArtikkel"
       xlink:to="TilArtikkel"/>

</relasjonsobjekt>
```

Figur 6.3: Et lenkeobjekt i Xlink

Figur 6.4 viser hvordan lenkeobjekter formelt kan modelleres i UML, med lenkeobjektet som en klasse som beskriver assosiasjonen mellom informasjonsobjekter. Siden et lenkeobjekt også kan defineres som et informasjonsobjekt, definerer vi det som en subtype av informasjonsobjekt. Relevante attributter for et slikt lenkeobjekt er en typebetegnelse for relasjonen, relasjonens retning og rollene de forskjellige informasjonsobjekter har i relasjonen. Videre er det behov for identifikatorer eller adresser til de informasjonsobjektene det lenkes mellom, og evt. mekanismer for å adressere inn i informasjonsobjektene (span-to-span lenking) til delene det lenkes mellom.

Referanselenking

Lenking mellom dokumenter er et sentralt tema for mange dokumentorienterte informasjonssystemer, og det finnes en rekke prosjekter for referanselenking. Dette er en type lenking som er basert på at referanser som forekommer i dokumentene skal være interaktive ved hjelp av lenker til de dokumentene det refereres til. Relevante eksempler på dette er The Open Journal Project [75], SFX [159] og BibRelEx [21].



Figur 6.4: Relasjoner/lenker som egne informasjonsobjekter

6.3 Bibliografiske relasjoner

Vi har i avsnittet om FRBR behandlet bibliografiske relasjoner på et overordnet plan, sett på hva slags typer relasjoner som fins. Men vi har sagt lite om hvordan de er representert i dagens systemer og hvilke muligheter som fins i det som allerede er registrert. Den følgende framstillingen er basert på innlegg Bernhard Eversberg[54] skrev til en epost-diskusjonsliste i forkant av en konferanse om AACR2 i 1997.

Et passende utgangspunkt kan være en post fra Norsk bokfortegnelse (NBF), se figur D.1 på side 190 i Appendiks D. Her er det en mengde relasjoner: mellom verk, mellom verk og uttrykk, mellom uttrykk, mellom personer/korporasjoner og verk, uttrykk, manifestasjoner og eksemplarer, alt samlet i samme MARC-post. I visse tilfeller relateres det til entiteter som ikke fins i katalogen fra før og som heller aldri vil forekomme der (engelsk original og svensk oversettelse fins ikke i NBF). I figur D.5 på side 192 er det gjort et forsøk på å representere posten etter FRBR.

Posten i NBF er katalogisert med utgangspunkt i et eksemplar av en manifestasjon, og det ligger mye arbeid i å spore opp original og andre entiteter som posten bygger på for å gjengi deres egenskaper. En del informasjon kan hentes fra dokumentet selv, men ofte må man konsultere tidligere registreringer.

Går man posten etter i sømmene i MARC-formatet (se figur D.1), vil man se at noen av relasjonene er formelt uttrykt gjennom 7XX-felt (biinnførsler), mens andre er uttrykt gjennom fritekstlige uttrykk dels hentet fra publikasjonen, dels hentet fra standardformuleringer i katalogreglene. Endelig er enkelte relasjoner etablert ved å hente informasjon fra autoritetsregister (personnavn).

Alle virkelige lenker etableres ved å kopiere identiske data. Det kan være tekstlige uttrykk som kan forstås umiddelbart, eller det kan være identifikatorer som må tolkes av systemet. Det er systemet som avgjør hvorvidt lenken kan brukes til noe.

Når man skal lage lenker mellom bibliografiske entiteter, kan dette altså gjøres på flere måter:

- **Tekstlige lenker**. Kopierer man en tekst – f.eks. fra en autoritetspost – inn i den posten man er i ferd med å lage, etableres lenker til autoritetsposten og til poster som har brukt samme autoritet. Det etableres også en relasjon fra autoritetsposten til de postene som har brukt autoritetsopplysningen. Slik kopiering må gjøres, ikke ved gjentatt skriving, men ved systemmessig kopiering. Fordeler ved en slik type lenking er for det første at den forstås umiddelbart og det er ingen problemer med utveksling av data, alle opplysninger følger posten. Ulempen er at slike lenker lett kan bli brutt. Den minste forandring i autoritetsposten, eller motsvarende opplysning i posten selv, fører til brudd. Et system med tekstlig lenking basert på kopiering av autoritetsopplysninger må derfor inkludere et system for å oppdatere alle poster som bruker autoritetsopplysningen, dersom autoritetsposten blir endret.

- **Identifikatorlenking**. Slik lenking skjer ved å kopiere en opplysning som identifiserer det tekstlige uttrykket. Istedenfor å legge inn forfatternavn i en post, legger man inn en identifikator til et autoritetsregister for navn der opplysningen kan slås opp. Fordelen ved dette er at det er enkelt å endre autoritetsopplysningen. En endring vil umiddelbart slå inn i alle poster fordi identifikatoren beholdes, det er bare opplysningen som identifikatoren peker på, som er forandret. Ulempen er at identifikatoren er meningsløs for leseren, den må alltid ved presentasjon erstattes av det tekstlige uttrykket. En annen ulempe er at utveksling av data blir komplisert. Autoritetsposten må følge posten selv om man ikke kan anta at mottakeren har tilgang til de samme autoritetspostene med samme identifikator. Dersom det etableres et globalt system med standard unike identifikatorer, vil mottaker av den bibliografiske posten selv kunne hente autoritetsopplysninger ved behov.

I de fleste bibliotekkataloger som presenteres i World Wide Web, blir flere bibliografiske opplysninger presentert som hypertekstlenker (klikkbare). I mange tilfeller mangler den andre enden av lenken. Dette skyldes at den bibliografiske opplysningen i de fleste tilfeller er etablert for å vise tilbake til posten, ikke til andre poster eller verk. Unntaket er standardtitler og lenking via autoritetsopplysninger som kommer nær virkelig lenking mellom verk.

Har man etablert en lenke den ene veien, vil systemet også alltid kunne generere en lenke den andre veien. Spesielt enkelt blir dette ved bruk av iden-

tifikatorer. En lenke fra *identifikator 1* til *identifikator 2* kan lett snus i en indeks og dermed gi en oversikt over alle poster som viser til *identifikator 2*.

Sammenlikner man figur D.1 og figur D.5, forstår man raskt at det vil være vanskelig å få til en automatisk overgang fra MARC-strukturen til FRBR-strukturen. Hvordan kan man innenfor dagens systemer likevel få til virkelig lenking? Eversberg foreslår å bruke identifikatorlenking med utgangspunkt i MARC-formatets felt 787 ("ikke spesifisert relasjon"). Gjennom MARC-indikatorer angis relasjonstype (som grovt sett omfatter Tilletts kategorier, supplert med noen nye). Lenken angis i delfeltet $w som en identifikator for lenkemålet, og kan gis et tekstlig uttrykk ved data i et annet delfelt – $g. Istedenfor å gjenta delfelt (slik det er definert i NORMARC), foreslår Eversberg at 787 gjentas for hver relasjon, dette gir grunnlag for en ryddigere datamessig behandling.

Kapittel 7

Infrastruktur

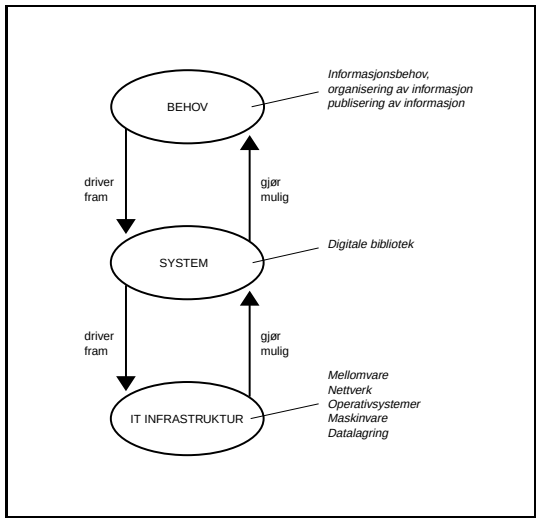
For å realisere digitale bibliotek er det behov for en underliggende informasjonsteknologisk *infrastruktur*. Vi har bruk for maskiner, operativsystemer, datalager og nettverk til å kjøre de forskjellige deler av systemet, og vi har bruk for verktøy og metoder for å utvikle disse systemene. Dette er generell teknologi som er uavhengig av bruksområde. Den samme infrastrukturen kan brukes til mange formål enten dette er et banksystem, system for reservering av flybilletter, eller digitale bibliotek.

De systemene vi lager er både et resultat av de behovene vi har og de teknologiske mulighetene infrastrukturen gir. Dette er likevel ikke et statisk bilde. Behovene våre er pådrivere for å utvikle systemer, og de systemene vi lager (eller ønsker å lage) er pådrivere for den teknologiske infrastrukturen. Dette er en påvirkning som også går den andre retningen ved at infrastrukturen gir oss muligheter til å lage systemer, og disse systemene er igjen med på å skape nye behov i samfunnet (se fig. 7.1).

Det finnes et bredt spekter av infrastrukturelle teknologier og standarder som er relevante for digitale bibliotek. Utfordringen for fremtiden er å velge en plattform av infrastruktur og standarder, videreutvikle og tilpasse teknologi, og ikke minst utvikle nye løsninger basert de spesielle behovene som finnes i dette domenet. En annen viktig utfordring er å kunne lage stabile løsninger som vil fungere selv om den teknologiske infrastrukturen endres.

7.1 Åpne systemer

Vi karakteriserer ofte informasjonssystemer som *åpne* (eller lukkete) og dette er en egenskap som sier noe om hvor utvidbart et system er. Åpne systemer er noe vi oppnår ved å spesifisere og dokumentere systemets grensesnitt. Ved hjelp av slike spesifikasjoner er det mulig å utvide systemet med ny funksjonalitet og legge til eller bytte ut komponenter. Det er denne åpne egenskapen som for eksempel gjør at vi kan sette sammen datamaskiner av deler fra forskjellige leverandører. Uavhengige leverandører kan produsere enheter som "passer



Figur 7.1: Infrastruktur, applikasjon og behov

sammen" fordi de kjenner grensesnittet enhetene skal kommunisere via.

Muligheten for kommunikasjon mellom programmer som kjører på forskjellige maskiner i et nettverk, gjør det mulig å dele ressurser mellom maskiner og programmer, f.eks. ved å tilby tjenester til andre. Slike systemer som deler ressurser over nettverk, gjennom dokumenterte grensesnitt, kalles ofte for *åpne distribuerte systemer*.

7.2 Klient/tjener og protokoller

Klient/tjener-konseptet brukes for å beskrive rollefordelingen mellom separate, men kommuniserende dataprogrammer (eller mer korrekt datamaskinprosesser), hvor den ene tilbyr tjenester og utfører disse på forespørsel fra en klient (tjener eller server), mens den andre er oppdragsgiver og konsument av disse tjenestene (klient). Klienten er ofte den som tar seg av interaksjonen med brukeren (brukergrensesnittet), mens tjeneren er den som kontrollerer en ressurs som deles, f.eks. en database eller en filtjener. En tjener må kunne betjene mange klienter samtidig, mens en klient ofte bare kommuniserer med en tjener i gangen. På World Wide Web er web-leseren vår en klient som kommuniserer med web-tjenere og får oversendt web-dokumenter.

Selv om klient eller tjener er begreper som ofte brukes for å kategorisere programvare, er klient eller tjener mer en rollefordeling for datamaskinprosesser enn spesifikke typer programmer. En prosess kan ha rollen som klient i forhold til en annen prosess, men samme prosess kan også tilby tjenester for andre og da ha rollen som tjener. Vi kan også ha distribuerte systemer uten denne rollefordelingen, f.eks. "peer to peer"-systemer hvor datamaskinprosessene som

kommuniserer, fungerer både som klient og tjener.

For at prosesser (f.eks. klient og tjener) skal kunne kommunisere, må de være enige om hvilke meldinger de kan sende seg imellom, hvordan disse meldingene skal være formatert og hva de betyr. *Protokoll* er en term som ofte benyttes for å referere til et sett av regler og formater som kan brukes i kommunikasjonen mellom klient/tjener-prosesser for å utføre en oppgave [36]. To viktige deler av en protokoll er:

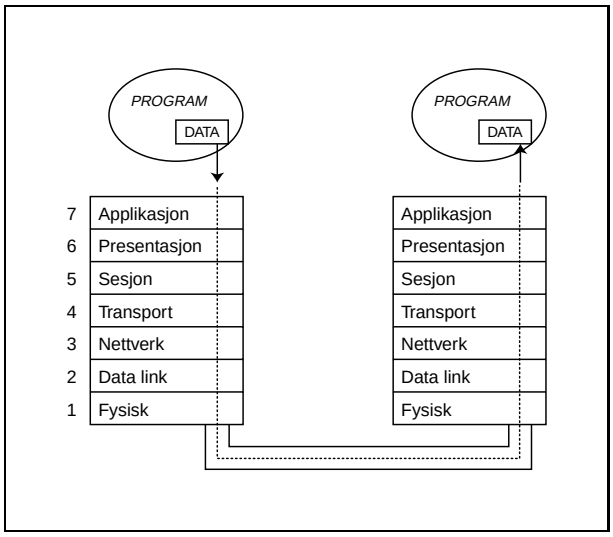
- En spesifikasjon av meldingene og sekvensen av meldinger som må utveksles.
- En spesifikasjon av formatet for data i meldingene.

I et nettverksbasert system bruker vi ikke bare en protokoll, men flere protokoller som er organisert i lag (ofte kalt en protokoll-stakk) hvor de laveste lagene tar seg av transport av pakker av data mellom maskiner, mens de høyere lagene er mer fokusert på håndtering av meldinger, logiske forbindelser eller spesifikke applikasjoner. Det er en slik lagdeling som gjør av vi kan kommunisere via Internett, enten det er ved hjelp av analog telefonforbindelse, ISDN, Ethernet eller radiobasert trådløs forbindelse.

Referansemodellen for *Open System Interconnections* (vanligvis bare kalt OSI-modellen) ble utviklet som en åpen arkitektur for å fremme åpne kommunikasjonsstandarder [92]. Den er strukturert i 7 lag som representerer den logiske sekvensen av funksjoner som utføres når meldinger konstrueres for overføring og sendes. De lavere lag i arkitekturen håndterer funksjonaliteten i nettverket, mens de høyere lag gir funksjonalitet i forhold til applikasjoner (programmene vi bruker) (se fig. 7.2).

- **Lag 7**. *Applikasjonslaget* er protokoller utviklet for å møte kommunikasjonskravene til spesifikke applikasjoner. De definerer ofte grensesnittet mot en tjeneste. Eksempler på slike protokoller er FTP [150], Telnet [149] og HTTP[1] [56].
- **Lag 6**. *Presentasjonslaget* er protokoller ansvarlige for å konvertere fra datarepresentasjoner som er spesifikke for applikasjoner, og til uavhengige dataformater for overføring i nettverket. Kryptering er et eksempel på dette laget.
- **Lag 5**. *Sesjonslaget* kontrollerer oppretting, håndtering og avslutning av logiske forbindelser (sesjoner) mellom to samarbeidende prosesser (eks. klient og tjener).

[1] TCP/IP-stakken er ikke direkte overførbar til OSI-modellen. I litteraturen finner vi varierende syn på hvor de forskjellige TCP/IP-protokollene skal innplasseres i OSI-modellen. Noen mener protokoller som FTP, Telnet og HTTP tilsvarer lag 7, mens andre mener det er riktigere å si at disse tilsvarer alle lagene fra 5 til 7.

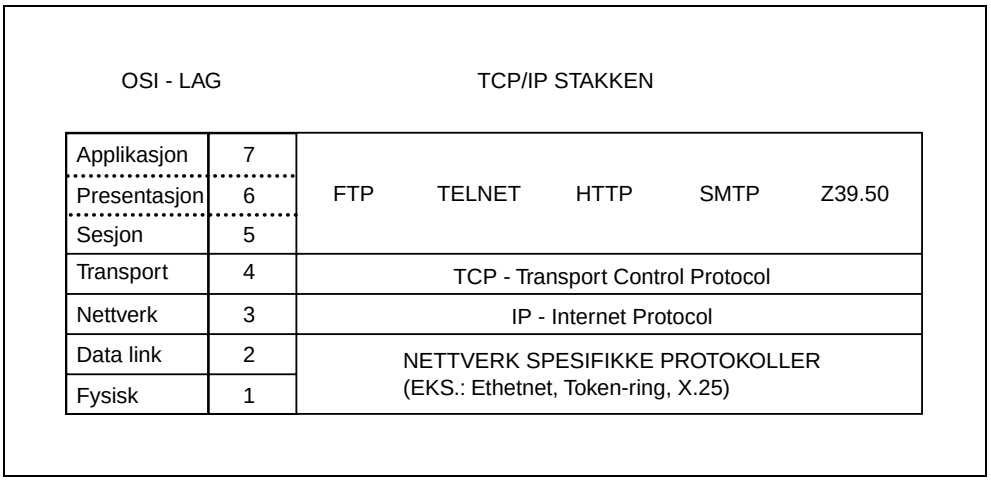


Figur 7.2: OSI Referansemodellen

- **Lag 4**. *Transportlaget* gir pålitelig dataoverføring for de høyere lag. TCP [2] og UDP [148] er protokoller på dette nivået.
- **Lag 3**. *Nettverkslaget* gir høyere lag uavhengighet fra maskinvareteknologi brukt i nettverket, og overfører pakker av data mellom maskiner i et nettverk. Eksempler er IP [3] og X.25 [94].
- **Lag 2**. *Datalink-laget* er ansvarlig for feilfri overføring av pakker mellom maskiner som er direkte koblet til hverandre.
- **Lag 1**. *Fysisk lag* består av kabler og maskinvaren i nettet. I dette laget overføres sekvenser av binærdata som elektriske signaler i kabler, lyssignaler i fiberoptiske kabler, eller elektromagnetiske signaler i radioforbindelser.

På Internett bruker vi mange forskjellige protokoller. Internett er egentlig en familie av protokoller med kommunikasjonsprotokollene IP (Internet Protocol) og TCP (Transport Control Protocol) som fundamentet. IP er en protokoll som tar seg av å sende pakker av data (datagram) mellom maskiner i det nett av nettverk som Internett egentlig er. IP sender bare pakker uten å bekymre seg om disse ankommer mottakende maskin i riktig rekkefølge og uten feil. TCP er protokollen som tar seg av å opprette en logisk forbindelse mellom programmene som kommuniserer, organiserer datagrammene fra IP protokollen, og korrigererer feil m.m. Med TCP og IP i bunn er det utviklet en rekke mer bruksorienterte protokoller - det vi vanligvis assosierer med Internett; HTTP, TELNET, FTP, Z39.50 [5], epost-protokollen SMTP [151] m.fl. Figur 7.3 viser

lagene i TCP/IP-stakken sett i forhold til OSI-modellen, selv om TCP/IP ikke helt følger inndelingen i OSI.



Figur 7.3: TCP/IP-familien av protokoller

7.3 Objekter

Objektorientering er en teknologi (eller metode) som har vært viktig i programvareutvikling de senere årene. Vi finner også objektorientering brukt på mange områder; objektorientert design og modellering, objektorienterte databaser, objektorienterte brukergrensesnitt, og ikke minst *distribuerte objekter*.

Kort beskrevet innebærer objektorientering at data og funksjonalitet samles i objekter som representerer (mer eller mindre) naturlige abstraksjonsenheter. Et objekt er en enhet som innkapsler både data og assosierte metoder (funksjoner). Metodene utgjør objektets grensesnitt med omverdenen og vi kommuniserer med objektet ved å sende meldinger. En melding har den effekten at en av objektets definerte metoder kalles. Vi kan med andre ord nyttiggjøre oss objektet uten å vite hvordan dette er implementert, så lenge vi kjenner de metodene som utgjør objektets grensesnitt med omverdenen. Objekter organiseres i klasser som er typer av objekter, og en sentral egenskap i objektorientering er arv mellom klasser; muligheten for at en klasse kan arve egenskaper fra en eller flere andre klasser. På et mer generelt nivå kjennetegnes objektorientering ved følgende aspekter:

- **Innkapsling** som gjør at vi kun aksesserer objektene via et veldefinert sett av metoder (objektets grensesnitt). Denne egenskapen, som også kalles informasjons-skjuling, gir mulighet for å skjule data samtidig som funksjonalitet og data grupperes på en logisk måte.

- **Abstraksjon** som er å gruppere assosierte objekter i klasser i henhold til deres egenskaper, for eksempel det sett av objekt-instanser som tilhører samme klasse.
- **Polymorfi** er muligheten for at to objekter kan ha like grensesnitt (like metoder), selv om metodene internt er implementert på forskjellig måte, og kall til metoden derfor håndteres forskjellig av objektene.

En skiller vanligvis mellom objektbasert og objektorientert. Med objektbasert menes f.eks. programmeringsspråk som bruker objekter og klasser, mens bruken av termen objektorientering også indikerer bruk av arv og polymorfi.

En objektbasert eller objektorientert modell kan gi flere fordeler i utvikling og vedlikehold av distribuerte systemer [155]:

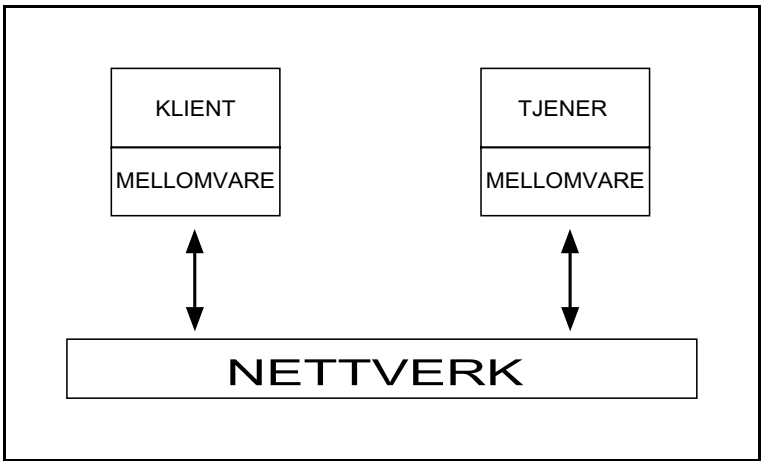
- Objekter er en naturlig enhet for distribuering, og bruken av meldinger for kommunikasjon mellom objektene passer godt inn i et distribuert system-miljø.
- Bruk av objekter er velegnet for å integrere eksisterende løsninger ved at et objekt kan innkapsle et allerede eksisterende system eller del av et system, for på den måten å gjøre det lettere tilgjengelig i et nettverk.
- Objektene kan skjule implementasjonsdetaljer, og bruk av objektene skjer kun gjennom definerte grensesnitt. Dette kan effektivisere utvikling og vedlikehold av programvare.
- Objekter kan implementeres for å håndtere forskjellige ressurser, men med likt grensesnitt.

Distribuerte objekter er blitt et viktig satsingsområde i informasjonsteknologien. Vi finner mange basis-plattformer for objektorienterte løsninger som Microsofts DCOM [125] og OMG's CORBA [141]. Også for HTTP er det visjoner om en mer objektorientert løsning (HTTP-NG) [107].

7.4 Mellomvare

Bruk av nettverk og tilgang til distribuerte ressurser er en kompleks oppgave. Når vi skal utvikle eller bruke distribuerte tjenester og ressurser, ønsker vi å gjøre dette på en mest mulig transparent måte. Vi ønsker å kunne utvikle programmer og få utført tjenester på andre maskiner uten å måtte forholde oss til kompleks nettverkskommunikasjon og forskjeller i operativsystem, maskinvare og programmeringsspråk. *Mellomvare* (middleware) gjør det mulig for enkeltdeler i et distribuert system å virke sammen uavhengig av maskinvare, operativsystem og programmeringsspråk, og hvor bruk av nettverket for å binde disse ressursene sammen er en integrert del av mellomvaren (fig. 7.4). Vi kan si at mellomvare er der for å skjule bruken av komplekse protokoller. Vanlige hovedtyper av mellomvare er:

- **Remote Data Access (RDA)** som er protokoller og verktøy for å kommunisere med databaser/datalager, f.eks. SQL og Microsofts ODBC.
- **Remote Procedure Call (RPC)** er en teknologi/protokoller for å kalle funksjoner på andre maskiner i et nettverk. Suns RPC (remote procedure call) og Java RMI (remote method innvocation), er eksempler på slik teknologi.
- **Message Oriented Middelware MOM** er basert på meldinger som sendes mellom de kommuniserende enhetene. En melding kan inneholde formaterte data eller forespørsler, og kommunikasjonen er ofte asynkron og basert på "peer-to-peer".
- **Object Request Brokers (ORB)** er mellomvare som håndterer kommunikasjonen mellom objekter som er distribuert på forskjellige maskiner i et nettverk. OMGs CORBA [141] er en standard arkitektur og spesifikasjon for ORB som gjør at forskjellige leverandører kan utvikle ORB'er som er interoperable. IIOP er en TCP/IP-protokoll for kommunikasjon mellom slike ORB som gjør at de forskjellige leverandørenes ORB'er kan kommunisere. IIOP er etter hvert blitt vanlig som basis protokoll for de fleste ORB'er.
- **Distributed Transaction Processing** er håndtering av transaksjoner i et distribuert miljø. En transaksjon er et sett eller en sekvens av operasjoner som hører sammen og som skal realiseres som en enhet. I tilfelle det oppstår en feil, f.eks. ved at en av operasjonene ikke lar seg utføre, skal heller ingen av de andre deloperasjonene utføres.



Figur 7.4: Mellomvare

7.5 World Wide Web

World Wide Web (WWW) er et globalt hypertekstsystem som ble introdusert av Tim Berners Lee. Den første beskrivelsen av World Wide Web stammer fra 1989 [28, 116], og var et forslag om et system for distribuert hypertekst for bedre å kunne dele informasjon ved CERN (det europeiske laboratoriet for partikkelfysikk).

Selve ordet World Wide Web dukket først som navnet på et program som ble utviklet for å navigere og editere hypertekstdokumenter. Dette programmet som het *WorldWideWeb* (uten bruk av mellomrom) ble senere omdøpt til *Nexus* for bedre å kunne skille mellom programmet og det abstrakte informasjonsrommet World Wide Web (som nå ble skrevet med bruk av mellomrom) [115].

I løpet av 1990-tallet har World Wide Web hatt en eksplosiv vekst, og er i dag arena for et bredt spekter av informasjon og tjenester. Veksten til World Wide Web er ikke bare en økning i mengde (antall web-steder og web-sider), men vi finner også at World Wide Web utvikler seg i bredden ved nyutvikling av web-relatert teknologi eller ved at eksisterende teknologi tilpasses eller integreres med World Wide Web.

Både World Wide Web og digitale bibliotek er informasjonssentrerte og har som målsetting å gi et globalt publikum tilgang til informasjon og tjenester. Digitale bibliotek er til en viss grad utviklet i kjølvannet av World Wide Web, og mye av det som er og har vært sentralt i utviklingen av World Wide Web, er også et viktig fundament for digitale bibliotek. Web-teknologien har derfor en sentral plass i digitale bibliotek selv om det kanskje er viktig igjen å minne om at World Wide Web ikke er et digitalt bibliotek, som diskutert i kap. 1.5.

7.5.1 Web-teknologi

Den riktige forståelsen av World Wide Web er det abstrakte rommet av informasjon (og tjenester) som er knyttet sammen ved hjelp av web-lenker. Som fundament for dette informasjonsrommet finner vi et sett av teknologiske løsninger som vi kan karakterisere som web-teknologi. Web-teknologi og web-relatert teknologi er blitt et så stort markedsegment at det antagelig er riktig å snakke om enda en type mellomvare – web-mellomvare.

Protokollen HTTP (Hypertext Transfer Protocol) og hypertekstformatet HTML (Hypertext Markup Language) utgjør selve kjernen i web-teknologien sammen med URL (Uniform Resource Locators) som adresseringsmekanisme[2]. World Wide Web's suksess er også basert på bruken av Internett (TCP/IP - protokollene) som global ryggrad, web-leseren som universell klient både for å lese web-sider og som grensesnitt for andre tjenester, og web-tjenere som universell tjener eller portal til informasjonskilder og tjenester.

[2] URI (og URL etc.) har sitt utspring i World Wide Web, men er i dag relatert til hele Internet. Det er derfor riktigere å karakterisere dette som Internett-teknologi enn web-teknologi.

World Wide Web som informasjonsrom er likevel ikke avgrenset til kun å være basert på HTTP, HTML og URL. Det er også andre protokoller, dokumentformater med støtte for hypertekst, og andre adresseringsmekanismer i bruk.

7.5.2 HTTP

HTTP er en forkortelse for "Hypertext Transfer Protocol", og den karakteriseres som en protokoll for distribuerte, samarbeidende, hypermediasystemer [56]. Det er en generisk og tilstandsløs protokoll som også kan brukes til andre oppgaver enn å formidle hypertekst, og den kan utvides for å nyttes til nye formål[3]. Vi kaller en tjener som kommuniserer over HTTP for en web-tjener selv om det kanskje hadde vært mer korrekt å kalle denne for en HTTP-tjener. En web-tjener kan være utviklet kun med publisering av filer via HTTP som bruksområde, men en web-tjener kan også være implementert som del av annen programvare som bruker HTTP for å formidle informasjon, eller det kan være implementert mellomleddsprogrammer som web-tjeneren benytter for å hente ut informasjonen den skal publisere. Hvordan informasjonen er lagret og hentes ut internt på et web-sted, er irrelevant for HTTP.

HTTP er basert på meldinger (HTTP messages) hvor en klient sender *forespørsel-meldinger* (request messages) til en tjener, og tjeneren sender *respons-meldinger* til klienten. Et vanlig tilfelle er at en web-leser sender en forespørsel som inneholder en GET-kommando og et dokumentnavn. Som svar til denne forespørselen returnerer web-tjeneren en respons-melding som inneholder det forespurte dokumentet (fig. 7.5).
Både forespørsel og respons meldinger består av:

- en startlinje, henholdsvis
  - forespørsel-linje for forespørselmeldinger
  - statuslinje for responsmeldinger
- null eller flere meldings-overskrift (message header)
- en tom linje som indikerer slutten på meldingsoverskriftene
- et eventuell meldings-innhold (message-body)

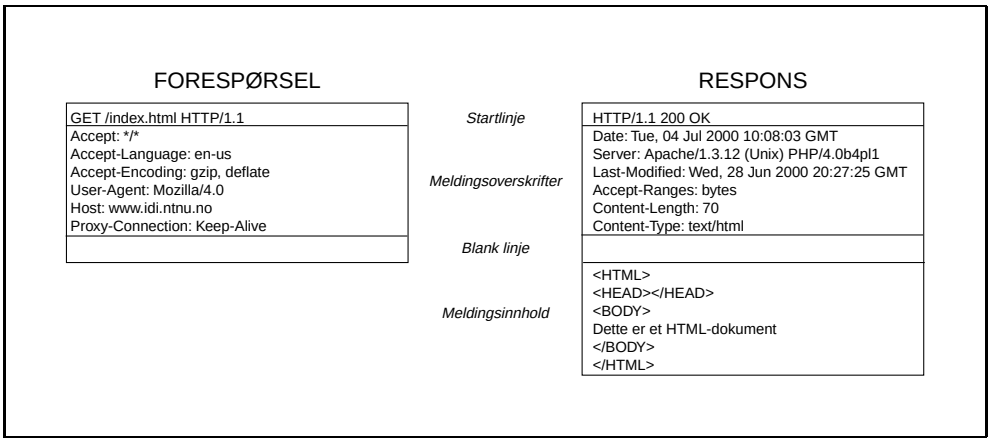
Forespørsel-linjen

Startlinjen for forespørsler består av et metodetegn etterfulgt av den URI som forespørselen gjelder for (forespørsel-URI), og til slutt protokoll versjonen, f.eks.:

```
GET /index.html HTTP/1.1
```

Her er metoden GET benyttet, og dette er en forespørsel om å få returnert den gjeldende web-siden.

[3] I innledningen til dokumentet som beskriver HTTP [56] refereres det til HTPCPC - Hypertext Cofee Pot Control Protocol [124] som et humoristisk eksempel på dette. Eksempler på mer seriøse utvidelser av HTTP protokollen er Dienst (s. 168), DASL (s. 154) og Cookies (s. 147).



Figur 7.5: HTTP meldinger

HTTP støtter et lite antall metoder (metode-tegn) som kan sendes i en forespørsel:

- GET betyr å hente (som et meldings-innhold) den informasjonen som forespørsel-URI identifiserer. Identifiserer URI en fil (f.eks. et dokument) er det dette som returneres. Identifiserer URI en data-produserende prosess (dynamisk web-side), er det de produserte data som skal returneres og ikke koden til programmet.
- POST benyttes for å forespørre tjener om å motta den informasjonen som følger i meldings-innholdet som en underordnet del av ressursen URI identifiserer. POST er ment å dekke bl.a. følgende funksjoner:
  - Annoteringer av eksisterende ressurser
  - Posting av meldinger til nyhetsgrupper, epostlister o.l.
  - Sende en blokk av data til en dataproduserende prosess (tilsvarende som for GET)
  - Legge til informasjon i en database
- PUT forespør tjener om å lagre den entitet som følger i meldingsinnholdet, under den URI som forespørsel-linjen angir. Dette kan benyttes til f.eks. å lagre web-sider via web-tjeneren i stedet for lokal filorganisering (som er den vanligste måten å gjøre dette på).
- Andre metoder er OPTIONS, HEAD, DELETE, TRACE, CONNECT. For mer informasjon om disse metodene henvises til HTTP-spesifikasjonene i [56].

Status-linjen
Startlinjen for en respons-melding inneholder protokollversjon, statuskode og en kort tekstlig beskrivelse av hva statuskoden indikerer, f.eks.:

```
HTTP/1.1 200 OK
```

Statuskoden er en resultatkode for tjenerens forsøk på å tilfredstille klientens forespørsel. Eksemplet over har statuskode 200 som forteller at enhetene som ble forespurt er sendt til klient. En annen velkjent statuskode er 404 som forteller at web-siden ikke eksisterer.

Meldingsoverskriftene
I HTTP brukes meldingsoverskriftene for å sende informasjon om meldingene og andre aspekter ved kommunikasjonen mellom klient og tjener. En meldingsoverskrift er delt i et navn og en feltverdi, adskilte med kolon. I HTTP-versjon 1.1 er det definert ca. 50 forskjellige meldingsoverskrifter. Noen av disse er generelle, andre er spesifikke enten for forespørsel eller respons. En viktig meldingsoverskrift er content-type som beskriver hvilket format meldings-innholdet har (typifisering), f.eks.:

```
content-type: text/html
```

Dette forteller web-leseren at dokumentet responsen inneholder er et HTML-dokument.

Meldingsinnholdet
Den delen av meldingen som inneholder entiteten (eller entitetene) som er assosiert med forespørselen eller responsen, kalles meldingsinnholdet (message-body). Det er f.eks. i denne delen av meldingen vi finner et dokument som sendes fra tjener til en klient.

7.5.3 HTTP og interaktivitet

Ved hjelp av metodene, meldingsoverskriftene og meldingsinnholdet kan HTTP støtte mange former for interaksjon mellom en bruker og en web-tjener. Vi har i avsnittet over beskrevet kun de metodene i HTTP som har å gjøre med overføring av informasjon fra bruker til tjener (ikke systeminformasjon som går mellom klient og tjener). Den mest brukte metoden er selvsagt GET, fordi det er dette metodetegnet som sendes til en web-tjener når vi klikker på en hypertekstlenke i et web-dokument. GET benyttes når vi forespør et dokument – det vil si at vi ønsker å få tilsendt informasjon fra tjener til klient. POST og PUT er beregnet på informasjon som skal oversendes fra klient til tjener. I realiteten er skillet mellom disse litt diffust, og metodene kan benyttes om hverandre. GET-metoden er i realiteten den vanligste å benytte for å sende f.eks. et søkeuttrykk fra en klient til en tjener, noe som er mulig pga. HTTP-URL-syntaksen hvor vi kan legge søkeuttrykk eller andre attributt-verdi-par i

HTTP-URL'ens søkedel (se fig. 5.8). Tilsvarende informasjon kan også sendes via POST, men da følger søkeuttrykket forespørselen som en del av meldingsinnholdet og er ikke i søkedelen av HTTP-URL'en. En tommelfingerregel er at GET skal benyttes når forespørselen ikke resulterer i noen oppdatering på tjenersiden [176] – er harmløs. Skal forespørselen resultere i en oppdatering eller endring på tjenersiden, bør POST benyttes. Det er selvsagt ikke alle som følger denne tommelfingerregelen. GET og POST har også den praktiske forskjellen at når GET benyttes vil informasjon som sendes til tjenere være synlig ved at den inngår i URL, og vi kan bruke URL-en som en vanlig lenke, f.eks. lage et bokmerke av denne i web-leseren eller legge den inn i et annet web-dokument. Når POST benyttes, er informasjonen ikke synlig for bruker.

PUT og POST har også overlappende funksjonalitet relatert til det å lagre dokumenter som filer på en web-tjener – såkalt "file upload" eller fil-opplasting. I dette tilfellet går det filer fra web-leser til web-tjener, og dette er relevant funksjonalitet på mange områder, f.eks. samarbeidsverktøy som BSCW[4], editorer for web-sider, og andre tjenester som er basert på at tjenersiden skal motta dokumenter fra bruker. Selv om PUT og POST er definert som forskjellige funksjoner, kan begge metodene benyttes til fil-opplasting. POST er det som i realiteten brukes når fil-opplasting skjer ved hjelp av web-leser[5]. PUT er også en del i bruk, men da som oftest i andre typer klienter enn web-leseren, f.eks. web-editorer. Bruk av PUT krever at web-tjeneren har støtte for denne metoden og er konfigurert for å ta imot dokumenter via PUT-forespørsler.

7.5.4 HTTP og sesjoner

HTTP er en tilstandsløs protokoll. Dette betyr at en web-tjener ikke automatisk husker noe om den enkelte bruker/klient fra forespørsel til forespørsel. En web-tjener svarer på en klients forespørsler uten å relatere dette til tidligere eller senere forespørsler. Dette kalles tilstandsløs eller sesjonsløs kommunikasjon. Sesjonsbasert kommunikasjon er derimot basert på at kommunikasjonen mellom tjener og klient kan foregå i en sekvens av meldinger som er relaterte til hverandre. Den tilstandsløse egenskapen er tilsynelatende i overenstemmelse med distribuert hypertekst hvor brukerne hopper fra dokument til dokument, og hvor hver webtjener må kunne håndtere et stort antall samtidige brukere, men kun har i oppgave å sende ut dokumenter.

HTTPs tilstandsløshet gjør det problematisk å lage interaktive tjenester som er basert på en kommunikasjon mellom bruker og web-tjeneren med lengre varighet, og som består av en sekvens av relaterte meldinger. Dette proble-

[4] se: http://bscw.gmd.de

[5] Årsaken til dette ligger mer i hva det er utviklet støtte for i HTML og i web-leserne, enn i forhold til hvordan HTTP-spesifikasjonene definerer disse funksjonene. HTML forms ble utviklet for å gi støtte for overføring av brukerdata til web-tjeneren. HTML forms støtter GET og POST metodene, men ikke PUT. I versjon 3 av HTML ble det også inkludert en inndata-knapp for filnavn, noe som gjør det mulig å benytte standard HTML og web-leser for slike tjenester.

met gjør seg blant annet gjeldende for web-baserte søketjenester som bruker trefflister som er delt opp i flere sider, f.eks. med et visst antall treff pr. side. Tilstandsløsheten ved HTTP gjør det også vanskelig å implementere søketjenester som husker en hel søkesesjon – det vil si alle søk som brukeren har utført siden han kontaktet webtjeneren – og for eksempel implementere funksjonalitet hvor brukerne har mulighet til å kombinere resultatsett fra forskjellige søk. Dette problemet er ikke spesifikt for søketjenester, men vil selvsagt også være til stede for alle tjenester med behov for å huske informasjon i løpet av en interaksjons-sekvens.

Det at HTTP er tilstandsløs er ikke det samme som at vi ikke kan implementere web-baserte tjenester med sesjoner, men fører til at sesjoner er noe som må implementeres for hver enkelt av de tjenestene som benytter dette. En vanlig teknikk for å gjøre dette er å sende en identifikator frem og tilbake mellom klient og tjener slik at tjener er i stand til å gjenkjenne klienten, og at tjenere bruker dette for å styre eller simulere en sesjon. Dette kan gjøres ved å sende en sesjonsidentifikator frem og tilbake mellom klient og tjener. Fra klient til tjener kan sesjonsidentifikatoren inngå i søkedelen av en URL for GET-metoden, eller som et attributt-verdi-par i meldingsinnholdet i en POST-melding. Fra tjener til klient kan en slik identifikator ligge i HTML-dokumentet, f.eks. som et skjult felt i en HTML-form.

En annen mye anvendt teknikk er å bruke en utvidelse til HTTP-protokollen som kalles "cookies" – småkaker [110]. Cookies ble først introdusert av Netscape og er nå en måte å implementere tilstand/sesjoner på som støttes av mange forskjellige web-lesere og web-tjenere selv om den ikke er del av HTTP-protokollen. En "cookie" er tilstandsinformasjon som sendes mellom en tjener og klienten, og som lagres av klienten. Cookies er basert på bruken av to nye meldingsoverskrifter: set-Cookie og Cookie. Som del av en responsmelding kan en tjener sende meldingsoverskriften set-Cookie med tilhørende verdi(er). Verdien lagres på klientmaskinen og returneres når klienten sender en ny forespørsel til web-tjener (ved å bruke Cookie-meldingsoverskriften). Den informasjonen som sendes som er basert på attributt-verdi-par, men med en del tilleggsattributter som kan benyttes for å kontrollere hvilke ressurser denne "cookie" gjelder for, hvor lenge den er gyldig, o.l.

7.5.5 Dynamiske web-sider

HTTP sammen med HTML utgjør selve kjernen i det vi kan kalle webteknologi, men det er også utviklet annen teknologi som gir nye muligheter og tilfører World Wide Web ny funksjonalitet i forhold til det disse spesifikasjonene gir. Det vi kjenner som web-sider, er i utgangspunktet basert på HTML, men HTML gir også rom for å integrere andre løsninger enn det et rent HTML-dokument kan gi. Ved hjelp av klientside-skript kan vi supplere HTML-dokumentene med egendefinert funksjonalitet, og ved hjelp av Java applets kan vi bruke web-leseren og web-sidene som en omgivelse for program-

mer som lastes ned fra nettet når vi trenger dem. Dette er løsninger som gir oss mulighet for å tilføre web-dokumentene ny og kanskje mer velegnet funksjonalitet, og som gir oss muligheten til å gjøre web-sidene mer dynamiske – web-sider som er interaktive og kan prosessere inndata og produsere utdata uten direkte kommunikasjon med en web-tjener.

Klientside-skript
HTML har støtte for såkalte klientside-skript via HTML-elementet SCRIPT [176]. Et skriptspråk inneholder instruksjoner for funksjonalitet, og tilsvarer et programmeringsspråk, men med det unntaket at den koden som skrives blir oversatt til maskinkode når (og hver gang) den kjøres. Slike skript lagres og formidles derfor som vanlige tekstfiler, og når vi skal utføre skriptet er vi avhengige av et tolkingsprogram (interpreterer) som oversetter koden til handling. Et annet karakteristisk trekk ved skriptspråk er at de er enklere å skrive, og de omtales gjerne som programmeringsspråk for ikke-programmerere.

HTML har generisk støtte for skript, noe som vil si at HTML ikke forutsetter et spesifikt skriptspråk, men kun tar seg av innkapsling av skriptet i et HTML-element. Ved hjelp av SCRIPT-elementet i HTML kan vi enten legge skriptkode direkte i HTML-dokumentet eller indirekte ved hjelp av en URL til det tekstdokument som inneholder skriptkoden. Støtte for skriptspråk er noe som kan være implementert i web-leseren fra leverandørens side, men vi kan også utvide web-leseren til å støtte andre skriptspråk ved å installere tilleggsmoduler – såkalte plugins. Grunnen til at dette kalles klientside-skript er at skriptkoden overføres til klienten (web-leseren) og utføres på denne.

Da Netscape introduserte versjon 2 av sin web-leser, inneholdt denne støtte for et skriptspråk kalt JavaScript[6]. Microsoft la inn støtte for to forskjellige skriptsspråk i sin Internet Explorer 3.0: VBScript som har syntaks basert på Visual Basic, og JScript som var kompatibelt med JavaScript. For å forhindre inkompatibilitet mellom de forskjellige web-lesernes støtte for JavaScript ble skriptspråket standardisert av den europeiske standardiseringsorganisasjonen ECMA, og fikk det nøytrale navnet ECMAScript [63, 57, 50]. Selv om det eksisterer også andre skriptspråk som enkelte web-lesere støtter eller som kan benyttes ved å installere en plugin, er det først og fremst JavaScript/ECMAScript som er det plattformuavhengige alternativet med best støtte i web-leserne.
Ved hjelp av JavaScript kan vi tilføre web-sidene funksjonalitet på flere områder:

- **Kontroll av innhold og vising av dokumentet.**
  Et dokumentet prosesseres (parses) av web-leseren før det presenteres, og

[6]Navnet Javascript har utrolig nok ingenting med Suns programmeringsspråk Java å gjøre, med unntak av noe likhet i syntaks. Javascript ble utviklet av Netscape opprinnelig under navnet Livescript, men det var en markedsføringsallianse med Sun som førte til navnet Javascript. JavaScript er et generelt skriptspråk som ikke bare er beregnet på web-lesere og HTML, blant annet brukes JavaScript av Adobe i noen av deres produkter, og Netscape benytter også JavaScript på tjenersiden.

ved hjelp av skript i HTML-dokumentet kan vi lage innhold i dokumentet eller endre utseende på dokumentet. Vi kan generere enkeltelementer som dagens dato og klokkeslett, sette bakgrunnsfarge og farge på tekstelementer, eller vi kan generere hele HTML-dokumenter fra bunnen av ved hjelp av skriptet og for eksempel vise dette innholdet i et eget vindu eller i en egen ramme (HTML-Frame).

- **Kontrollere web-leseren.**
  Skriptet kan også kontrollere web-leserens funksjonalitet. Vi kan spesifisere dialogbokser som kan dukke opp for å gi beskjeder til brukeren, eller innhente enkle inndata fra brukeren. Det er også mulig å styre opprettelsen og avslutningen av nye vinduer i web-leseren, med mulighet til å spesifisere om disse skal ha menylinje o.l. JavaScript gir også mulighet for å vise nye sider i andre vinduer, og vi får tilgang til listen over hvilke sider som er besøkt, tilsvarende det som vanligvis er implementert som menyvalg eller knapp for neste og forrige side, i web-leseren. En annen metode gir mulighet for å styre teksten som vises i web-leserens statuslinje.

- **Interaktive dokumenter.**
  JavaSkript gir også mulighet for interaktive dokumenter. Dette gjelder stort sett for HTML-Forms hvor vi kan knytte funksjoner til Formselementene. Et mye brukt eksempel for å illustrere dette er en kalkulator utelukkende implementert ved hjelp av JavaScript. Et annet område dette kan brukes på er å kontrollere om de data en bruker legger inn i et skjema er korrekte før dataene sendes til en web-tjener. Ved hjelp av JavaScript og Forms er det mulig å implementere funksjonalitet som tidligere måtte løses ved hjelp av kall til web-tjeneren.

Java applets

Java er et objektorientert programmeringsspråk som ble introdusert av Sun i 1995[7]. For å lage programmer fra programmeringskode skrevet i Java, må denne oversettes til byte-kode. Dette er et spesielt maskin-nært format som kan utføres ved hjelp av en Java-tolker. Vi kan med andre ord ikke kjøre Javaprogrammene direkte, men er avhengig av et skall som leser byte-koden og oversetter denne til maskininstruksjoner tilsvarende som for skriptspråk. Java byte-kode er portabel og plattformuavhengig. Det vil si at vi kan utvikle et program og bruke dette på alle slags maskiner så lenge disse har Java-miljøet installert.

Ved introduksjonen av Java var det lagt stor vekt på muligheten for såkalte "applets" og Java fikk mye oppmerksomhet på grunn av denne funksjonaliteten. En applet er en form for mini-program (mini-applikasjon) som er utviklet for å kjøres av web-leseren som et interaktivt web-dokument eller del av et

[7] For informasjon om Java: se http://www.javasoft.com

web-dokument. Ved hjelp av HTML-elementet APPLET eller OBJECT[8] kan applets innlemmes i web-sidene og lastes ned dynamisk til web-leserne som aksesserer denne web-siden.

Vi kan utvikle applets med samme funksjonalitet som andre programmer med enkelte begrensninger for å ivareta sikkerhet, bl.a.:

- En applet kan ikke lese og skrive filer på maskinen.
- En applet kan ikke kommunisere over nettet med andre maskiner enn den tjeneren den kom fra.
- En applet kan ikke starte et annet program på maskinen.
- En applet kan ikke lese spesielle systemvariabler på maskinen.
- Nye vinduer som en applet starter opp, ser forskjellig ut i forhold til vinduer andre programmer starter opp.

Selv om Java fikk stor oppmerksomhet på grunn av applets da programmeringsspråket ble introdusert, er det i dag andre aspekter ved Java som er årsaken til dets suksess. Den utbredte anvendelse av applets som enkelte mente ville komme, har latt vente på seg, selv om applets anvendes på mange områder.

Plugins

Plugin er en generell betegnelse for tilleggsmoduler som kan installeres (plugges inn) på en maskin, og som gir ekstra funksjonalitet til web-leseren. En plugin kan tilføre web-leseren støtte for nye typer media som video og lyd, slik at avspilling av dette er integrert med web-leseren og ikke avhengig av at det startes opp nye programmer for å håndtere dette. Hvilke mediatyper en web-leser støtter uten å måtte installere spesielle plugins, er avhengig av hva leverandøren har valgt å implementere, men i HTML 4.0 finner vi et generelt element for inkludering av alle mulige slags objekter i en web-side med OBJECT-elementet. Før dette elementet ble introdusert som del i HTML-standarden, var det kun støtte for å inkludere bilder ved hjelp av IMG og applets ved hjelp av APPLET. Både Netscape og Microsoft har i sine web-lesere implementert støtte for et HTML-element EMBED for å kunne inkludere andre mediatyper. Plugins er ikke bare koblet til støtte for nye mediaformater, men kan også benyttes for å tilføre en web-leser ny funksjonalitet som ikke er relatert til web-dokumenter.

Ved hjelp av plugins har vi mulighet for å støtte mange slags medier i en web-leser og utvide web-leserens funksjonalitet, men plugins er generelt et problematisk område fordi programmene som utvikles ofte både er avhengige av operativsystem og maskinvareplattform i tillegg til at web-lesere ikke kan nyttiggjøre seg plugins som er utviklet for andre leverandørers web-lesere.

[8] I HTML versjon 4.0 anbefales det at en bruker OBJECT-elementet i stedet for APPLET.

7.6 Informasjonsgjenfinning

Informasjonsgjenfinning har en sentral plass i digitale bibliotek. I dette avsnittet ser vi på noen eksisterende løsninger som er i daglig bruk (Z39.50, HTTP-basert informasjonsgjenfinning og tilgang til databaser over nett), i tillegg til noen eksperimentelle løsninger som er under utvikling (DASL, DLIOP og STARTS).

7.6.1 Z39.50

Z39.50 er en velutviklet og standardisert protokoll for informasjonsgjenfinning [111, 127, 132]. Selv om Z39.50 i hovedsak har vært assosiert med bibliografiske databaser, er det en generell protokoll for gjenfinning som kan brukes også for andre typer av informasjon.

Protokollen har egentlig navnet "Information Retrieval", men da første versjon av protokollen ble publisert i 1988 som ANSI/NISO-standard, fikk den nummeret Z39.50, og dette er blitt det navnet protokollen er mest kjent under. Siste utgave av standarden er versjon 3, *ANSI/NISO Z39.50-1995* [5] og den korresponderende ISO-standarden er *ISO 23950:1998* [100]. Library of Congress er "maintenance agency" for protokollen[9], og utvikling av protokollen skjer gjennom et forum som kalles "Z39.50 Implementors Group" – ZIG.

Z39.50 er utviklet som en applikasjonslag-protokoll i forhold til OSI-modellen (fig. 7.2), og er nå i hovedsak brukt over Internett og implementert som en TCP/IP-basert protokoll.

Protokollen tar utgangspunkt i de sentrale elementene av informasjonsgjenfinningsprosessen – seleksjon av informasjon basert på søkekriterier og returnering av informasjon som tilfredstiller disse kriteriene. Z39.50 er sesjonsbasert og støtter også en rekke andre relevante aspekter ved informasjonsgjenfinning og brukeres interaksjon med en søketjeneste. Kommunikasjon mellom klient og tjener foregår ved hjelp av meldinger som protokollen kaller *tekniske tjenester* (services), disse er gruppert i *fasiliteter* (facilities):

- **Initialization**
  Etablering av sesjon og forhandling om tjenestenivå. Klienten ber om at det opprettes en sesjon, foreslår parametre for kommunikasjonen og sender evt. bruker-identifiserings-informasjon. Tjener svarer med en avvisning eller en aksept og de faktiske parametrene for kommunikasjonen.

- **Search**
  Klienten sender en søkestreng til tjener, søket utføres, og det genereres et resultatsett på tjenersiden. Klienten får returnert et tall for hvor mange poster som er funnet og evt. et visst antall poster.

[9] Se: http://lcweb.loc.gov/z3950/agency/

- **Retrieval**
  Klienten ber om spesifikke poster i et gitt format, fra et navngitt resultatsett. Responsen fra tjener inneholder disse postene.

- **Result-set-delete**
  Klienten ber om at ett eller flere resultatsett på tjenersiden slettes. Respons er en status for operasjonen.

- **Access Control**
  Tjener kan avvente utførelsen av en tjeneste klienten har bedt om, og be om autentifiseringsinformasjon (f.eks. passord) før tjenesten utføres eller avbrytes.

- **Accounting/Resource Control**
  Tjenester som kan brukes for å kontrollere, verifisere eller få informasjon om ressursforbruk, f.eks. for databaser hvor det betales for bruken.

- **Sort**
  Klienten kan be om at resultatsettet sorteres. Respons fra tjener er en status for operasjonen.

- **Browse**
  Klienten kan be om indekstermer. Responsen er en sortert liste over termer som kan brukes i et søk.

- **Extended services**
  Gir tilgang til tjenester som ikke er del av protokollen, f.eks. bevaring av resultatsett etter endt sesjon og bestilling av dokumenter.

- **Explain**
  Brukes for å få informasjon om tjeneren, hvilke databaser som er tilgjengelige, hvilke attributter som kan søkes, syntaks for postene som returneres, m.m.

- **Termination**
  Avslutning av sesjon. Både klient og tjener kan be om at sesjonen avsluttes, og dette kan gjøres når som helst i løpet av kommunikasjonssekvensen.

Z39.50 er en generell protokoll som ikke definerer ett spesifikt spørrespråk eller én spesifikk syntaks for de postene som returneres fra en tjener. Det kreves likevel at klient og tjener er i stand til å forstå hverandre på disse områdene, og dette gjøres ved å bruke standardiserte formater for både søkeuttrykket og for de postene som returneres.

Attributtsett
Et søk kan uttrykkes som en samling attributter med en eller flere verdier. Det er disse attributtene som spesifiserer hvordan søket skal utføres. Hvert attributt beskriver et aspekt ved søkeuttrykket, og det sett av attributter som brukes for å representere et søk kalles *attributtsett*. Attributtsettet *Bib-1* er en del av Z39.50-standarden, og var opprinnelig utviklet for bibliografiske databaser, men er etter hvert blitt vanlig å bruke også for andre typer informasjon. Det er også etablert/spesifisert andre attributtsett som kan benyttes, og det er selvsagt mulig å utvikle nye attributtsett når protokollen skal anvendes på nye områder.

Post-syntaks
Poster som returneres fra tjener, må følge en syntaks (format) som klienten er i stand til å behandle, f.eks. når posten skal presenteres for bruker. Siden Z39.50 i stor grad har vært brukt for bibliografiske ressurser, støtter det fleste tjenere og klienter forskjellige varianter av MARC, og spesielt USMARC. Det er også utviklet andre syntakser for postene, som SUTRS (Simple Unstructured Text Record Syntax) som er basert på ren tekst, og GRS-1 (Generic Record Syntax) som er et fleksibelt format som kan benyttes på de fleste databaser. Bruk av XML er et annet alternativ.

Profiler
I bruken av Z39.50 har protokollens generelle innretning vært både en fordel og et problem. Fordelen ligger i at protokollen ikke er knyttet til ett spesifikt bruksområde eller en spesifikk informasjonstype, men kan brukes til informasjonsgjenfinning på mange områder. Problemet har vært at klienten må kjenne hvilke tjenester en tjener støtter, og hvilke attributter og post-syntakser den støtter. Dette har ført til et behov for nøye konfigurering av klienten i forhold til den tjeneren den skal kommunisere med. Protokollen har etter hvert fått "explain"-tjenesten som kan brukes for å dynamisk gjøre seg kjent med tjenerens egenskaper, men dette aspektet er foreløpig lite implementert både på klientsiden og tjenersiden.

En annen tilnærming til dette er å definere *profiler*. En profil er en samling av de attributtene, post-syntaksene og tjenestene som er nødvendige for å tilfredstille et definert bruksbehov. Ved å bruke profiler kan vi derfor redusere de store mulighetene for variasjon og det resulterende kravet til konfigurering, og i stedet operere med definerte pakkeløsninger.

Z39.50 og World Wide Web
Web-lesere støtter dessverre ikke Z39.50 direkte, men vi kan søke via Z39.50-protokollen på World Wide Web ved at en web-tjener brukes som mellomledd mot Z39.50-tjeneren, mens web-leser brukes som grafisk grensesnitt. Andre løsninger for å integrere Z39.50 med World Wide Web er å implementere en Z39.50-klient som en Java-applet, dvs. at Z39.50-klientens grafisk grensesnitt blir en del av web-leserens vindu.

7.6.2 HTTP-basert informasjonsgjenfinning

Det finnes i dag et meget stort utvalg av tjenester som tilbyr informasjonsgjenfinning på World Wide Web. Søkesystemer som AltaVista, Lycos og AllTheWeb traverserer World Wide Web og bygger opp mer eller mindre dekkende indekser. Vi finner også en markant utvikling mot at web-steder som huser et stort antall web-sider, tilbyr informasjonsgjenfinningstjenester for sine dokumenter. I tillegg finnes det en rekke tjenester som gir oss tilgang til informasjon som ikke er direkte tilgjengelig som statiske web-sider, men som genereres ved forespørsel og er basert på informasjon som f.eks. er lagret i databaser.

De fleste av disse tjenestene er utviklet kun for bruk via web, og selv om de er basert på samme basisteknologi (HTTP, HTML-Forms, URL-syntaksen o.l.), er det store variasjoner i hva de støtter og syntaks og semantikk for søkeuttrykk og resultatsett. Problemet ligger i at disse grensesnittene er basert på generell teknologi som kan anvendes svært forskjellig.

For å kunne nyttiggjøre oss eller integrere slike tjenester som komponenter i digitale bibliotek, er det behov for:

- stabilitet – slik at grensesnittene ikke endres f.eks. når web-sidene endres.
- dokumentasjon – grensesnittene må spesifiseres slik at andre kan nyttiggjøre seg disse, f.eks. ved hjelp av definisjonsspråk som WIDL [187].

7.6.3 DASL

DASL (DAV Searching and Locating) er et initiativ for å utvikle en generell protokoll for søking med basis i HTTP [13]. DASL har sitt utspring i og er basert på WEBDAV [62] som er en protokoll for distribuert samarbeid om skriving av dokumenter.

Både WEBDAV og HTTP har støtte for at en klient skal kunne foreta et søk på en server. WEBDAV utvider HTTP-protokollen ved at ressurser (dokumenter) på tjenersiden kan ha metadata (properties) knyttet til seg, og WEBDAV benytter en PROPFIND-metode for å spesifisere et søk etter ressurser basert på disse metadataene. Dette er ikke noen fullgod løsning for å spesifisere søking, for mekanismen kan være lite effektiv, gir ikke mulighet for søking basert på innholdet i dokumentene, og tar ikke i bruk de mulighetene som kan ligge i lagringssystemene.

DASL har som mål å utvikle utvidelser til HTTP-protokollen som tar opp disse begrensningene. Sentralt i dette er løsninger for:

- Hvordan uttrykke et søk; syntaks og semantikk
- Hvordan fokusere et query; identifisere hva det skal søkes i
- Hvordan bli kjent med søkemulighetene til en ressurs
- Syntaks for søkeresultat

I DASL defineres en egen HTTP-forespørselmetode – SEARCH – for å sende en søkeuttrykk til en tjener. Søkeuttrykket kodes i XML i henhold til en definert DTD, og sendes som del av forespørselmeldingens innhold (tilsvarende som for en HTTP POST-metode). Tilsvarende er resultatsett og annen informasjon fra tjener kodet i XML i henhold til en DTD og sendes til en klient i responsens meldingsinnhold.

DASL er foreløpig "på tegnebrettet" og finnes kun spesifisert som et Internet-draft, og utviklingen av protokollen er lagt til en egen IETF arbeidsgruppe. DASL er et interessant prosjekt både fordi dette prosjektet har tilslutning fra en del store aktører som Microsoft og Netscape, og fordi det er en enkel søke-protokoll basert på utvidelser i HTTP, men som likevel har fleksibilitet nok til å støtte mange forskjellige typer søk.

7.6.4 DLIOP

Stanford DLI (Digital Library Inititative) var en av 6 store digitale bibliotekprosjekter i USA [10]. Disse prosjektene dekket forskjellige aspekter ved digitale bibliotek, og fokus for Stanford DLI-prosjektet var behovet for en teknisk infrastruktur for digitale bibliotek. I Stanford DLI benyttes en annen inndeling i tjenester enn Dienst, som håndtering av metadata, opphavsrett og betalingsmekanismer. Stanford DLI tar i tillegg utgangspunkt i en annen teknologi som infrastuktur, ved at det benyttes CORBA og distribuerte objekter.

Stanford-prosjektet fokuserte på å utvikle et sett av tjeneste-protokoller for å integrere informasjonsressurser og informasjonstjenester [152] . I Stanford DLI benytter man metaforen infobus (informasjonsvei) om et sett av protokoller som binder distribuerte digitale bibliotektjenester sammen [144]. Prosjektet er avgrenset til å dekke området mellom lavnivå transport (TCP/IP) og høyere nivåers funksjonalitet som søketjenester og brukergrensesnitt (det vi har kalt digitale bibliotekapplikasjoner).

DLIOP (Digital Library Interoperability Protocol) er en av protokollene som ble utviklet (og fortsatt er under utvikling). Dette er et sett med IDL[11] definisjoner for håndtering av søk og resultater fra søk. Protokollen har senere fått navnet *The Simple Digital Library Interoperability Protocol* (SDLIP)

7.6.5 STARTS

The Stanford Protocol Proposal for Internet Retrieval and Search (STARTS) [66] er en av de andre protokollene som er utviklet under Stanford Digital Library Initiativ. Målsettingen for denne protokollen var å gi mer standardisert støtte for metasøking – tjenester hvor en bruker formulerer et søk

[10] se http://www.dli2.nsf.gov/dlione/

[11] IDL – Interface Definition Language – er et formelt språk som benyttes i CORBA for å spesifisere objektenes grensesnitt, dvs. hvilke metoder som finnes for et objekt og hvilke parametre disse metodene aksepterer.

og metasøkesystemet videresender dette til en rekke andre søkesystemer, innhenter resultatene, og gir brukeren en enhetlig presentasjon av søkeresultatene. Behovet for slike løsninger ligger i det mangfold av store og små søkesystemer som finnes på Internett, alle med forskjellige søkegrensesnitt og innhold. De primære søkesystemer kalles i denne protokollen for "kilder". Problemstilingene som tas opp i STARTS er valg av relevante kilder å søke i, utføring av søk ved de valgte kildene, og sammensetting av resultater fra forskjellige kilder.

Valg av relevante kilder å søke i
Et metasøkesystem kan ha tusenvis av potensielle kilder å videresende et søk til, og metasøkesystemet må derfor kunne velge ut de mest relevante kildene. For å gjøre dette er det behov for metadata om kildens søke-egenskaper og innholdet i kilden (informasjon om dokumentene i samlingen). STARTS spesifiserer metadataformater som kan brukes til dette.

Utføring av søk ved de valgte kildene
De søkbare kildene vi finner på Internett bruker mange forskjellige spørrespråk, og dette representerer et stort problem for metasøkesystemer som må forholde seg til alle disse søkespråkene. STARTS definerer et enkelt og generelt spørrespråk som kildene må støtte.

Sammensetting av resultater fra forskjellige kilder
En metasøketjeneste som mottar resultatsett fra mange kilder må kunne sette disse sammen slik at de kan presenteres for brukeren på en enhetlig måte. En av utfordringene på dette området er håndtering av de forskjellige måtene dokumenter rangereres på. STARTS spesifiserer hvilken statistisk informasjon kildene må returnere for trefflista og hvert enkelt dokument i, hvordan man skal håndtere sortering og multiple forekomster av samme dokument i det sammensatte resultatsettet. Ved hjelp av dette er det mulig for et metasøkesystem å integrere resultatsettene fra mange kilder på en konform måte.

Kommunikasjon
STARTS er ikke en kommunikasjonsprotokoll, men mer en spesifikasjon for hvilken informasjon som er nødvendig for metasøkesystemer, og hvilke krav som må stilles til kildesøkesystemer for at de skal kunne anvendes av et metasøkesystem. Søkeuttrykk, resultatsett og metadata representeres som attributt-verdi-par ved hjelp av SOIF-formatet (se kap. 3.5.2). STARTS benytter HTTP for å kommunisere, og søk sendes som HTTP POST-meldinger med søkeuttrykket i en SOIF-tag i HTTP meldingsinnholdet. Søkeresultater returneres som en HTTP-responsmelding . Metadata formidles som filer, identifiseres ved deres URL, og aksesseres ved hjelp av HTTP, FTP eller andre protokoller.

7.6.6 Databasetilgang

Tilgang til databaser over nett (Remote Data Access), er et område som har en del til felles med informasjonsgjenfinnings-protokoller som Z39.50 og DASL, men dette er funksjonalitet som er mer generell og på et lavere nivå.

Løsninger for kommunikasjon med databaser er et område som er preget av store kommersielle interesser, fordi bruk av databaser er et hovedelement i de fleste informasjonssystemer. Standarder for tilgang til databaser, faller i tre kategorier:

- **Standardiserte spørrespråk**
  I databaseverdenen er SQL [90] den rådende standarden for å manipulere og spørre relasjonsdatabaser. SQL ble utviklet som et relasjonsdatabasespråk, på et tidspunkt hvor sentraliserte databaser dominerte. Standarden definerer derfor ingen mekanisme eller kommunikasjonsprotokoll for å transportere SQL-uttrykket og resultat mellom klient og tjener over nett.

  Det finnes også en rekke andre databasespråk, både rene spørrespråk og språk med muligheter for manipulering av data. Dette er enten leverandørspesifikke språk, eller språk som er standardiserte. Eksempelvis har ODMG[12] utviklet OQL (Object Query Language) [30], som er et spørrespråk for objektorienterte databaser.

- **API-standarder**
  API-standarder er standardiserte prosedyrekall, som gjør at klientprogramvare kan utvikles uavhengig av databaseleverandør. Når disse brukes er det i tillegg behov for et lag av mellomvare som tar hånd om kommunikasjon over nettet og oversetter API-kallene til et format som er forståelig for tjenerdatabasen. Rent praktisk betyr dette at man i tillegg til klientprogramvaren, må ha nødvendig programvare (drivere) for å kunne kommunisere med de forskjellige databasene. Dette kan virke som en tungvint måte å tilby interoperabilitet på, men i realiteten er dette en praktisk modell for databaseleverandørene, som sikrer at klientprogramvare kan utvikles av uavhengige, mens bruk av klienten mot databaser krever lisens. Dette gjør at databaseleverandøren kan tjene penger også på de programmene som bruker databasesystemet. Microsofts ODBC (Open Database Connectivity) er et eksempel på et mye anvendt API, andre varianter er Suns JDBC for Java, Borlands IDAPI og ISO-standarden CLI [102].

- **Protokoller for datatilgang**
  Dette er standardiserte protokoller for kommunikasjon mellom databasetjener og klient. Et vesentlig skille mellom slike løsninger og API-standardene, er at databaseleverandører som integrerer støtte for standardiserte protokoller, ikke lenger har lisenskontroll for klientsiden.

[12] Object Data Management Group. Se http://www.odmg.org

Kapittel 8

Systemarkitektur

Digitale bibliotek kan være mange forskjellige typer av informasjonssystemer. Det kan være et enkeltsystem med en samling av digitale informasjonsobjekter som kanskje også er tilgjengelige på World Wide Web. Slike frittstående og avgrensede systemer som ikke er ment for integrering i andre og større enheter, vil selvsagt ha sin interne arkitektoniske løsning, men dette aspektet er av mindre interesse for oss. I den andre enden av skalaen over type digitale bibliotek finner vi digitale bibliotek som et åpent distribuert system av informasjonskilder og tjenester som skal kunne integreres og gjenbrukes i mange sammenhenger og tjene mange formål. I denne konteksten er den overordnete systemarkitekturen viktig fordi den er fundamentet for integrering av delsystemer som er utviklet og drevet av forskjellige virksomheter.

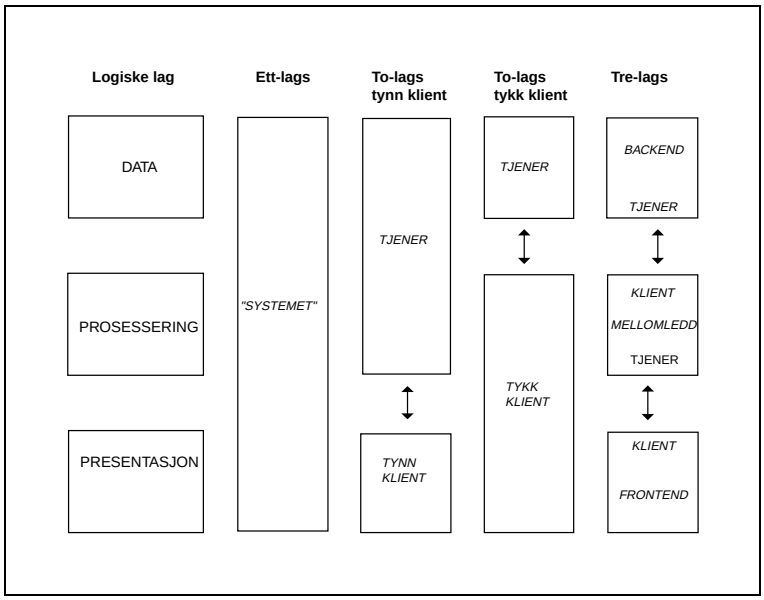
I den informasjonsteknologiske utviklingen har det vært mange store endringer som har gitt opphav til nye måter å strukturere informasjonssystemene på i forskjellige arkitekturer. For å forstå disse arkitekturene, definerer vi først følgende logiske lag i et informasjonssystem:

- data
- prosessering
- presentasjon

Fordelingen av disse logiske lagene på forskjellige systemenheter som kommuniserer over nett, er utgangspunktet for en kategorisering i noen generelle arkitekturer; ett-lags arkitektur, to-lags og tre- eller multi-lags arkitektur (fig. 8.1).

8.1 Ett-lags og to-lags arkitekturer

Da stormaskinene og dumme terminaler dominerte, var systemene tett integrerte enheter av datalagring, prosessering og presentasjon. Dette karakteriseres ofte som en *ett-lags arkitektur* fordi all lagring av data, prosessering og



Figur 8.1: Generelle arkitekturer

presentasjon, ble utført av ett og samme system. Interaksjon med systemet skjedde via "dumme" terminaler som ikke hadde annen rolle enn å være skjerm og tastatur mot brukerne (kalles også slave/master). Med den personlige datamaskinens inntog på 80-tallet kom muligheten for nye typer programvare beregnet på personlige brukere, og ikke minst ga dette opphavet til klient/tjener-rollefordelingen og *to-lags-arkitekturen*. Med to-lags menes at det overordnete systemet fra ett-lags-arkitekturen er delt på to forskjellige prosesser. Dette kan være at en tjener står for datalagring (filtjener eller databasetjener) og prosessering av informasjonen, mens klienten står for presentasjonen (brukergrensesnittet). To-lags-arkitekturen har lenge vært den rådende løsningen for distribuerte systemer, og etter hvert som pc-ene ble kraftigere og applikasjonene mer avanserte, var det naturlig å overlate deler av prosesseringen til klientene. Vi fikk det som kalles tykke klienter (fat client).

8.2 Tre-lags og multi-lags arkitekturer

To-lags-arkitekturen har sine begrensninger, og i dag er det fokus på *tre-lags arkitekturer*. Dette er en løsning som passer bedre for dagens informasjonsteknologiske landskap med stadig mer omfattende funksjonelle krav til systemene og behov for effektivitet, fleksibilitet og gjenbruk. Tre-lags-arkitekturen kjennetegnes ved at vi bruker et mellomledd mellom klienten og tjeneren fra to-lags-arkitekturen, f.eks. ved at vi splitter tjener-programmet i to. Et illustrerende eksempel på tre-lags arkitektur er bruk av web-tjenere som inngangs-

port til databaser. Web-leseren fungerer da som brukergrensesnitt (frontend), web-tjeneren representerer selve applikasjonen (mellomledd), mens en databasetjener kun er en generell ressursforvalter (backend). Web-tjeneren i en tre-lags-arkitektur er ikke begrenset til kun å bruke en database, men den kan hente inn og prosessere eller integrere, informasjon fra mange steder. Dette mellomleddet omtales av og til som applikasjons-tjener fordi den implementerer bruksfunksjonaliteten.

Bruk av Z39.50-protokollen for søking i bibliografiske databaser er ofte basert på en tre-lags arkitektur, fordi Z39.50 tjeneren fungerer som et mellomledd mellom klienten og den egentlige databasen som inneholder de bibliografiske dataene. Z39.50-tjeneren håndterer søkesesjoner, aksesserer databasen og formaterer svarene til resultatsett og utlistinger i tråd med Z39.50-protokollen. Mange Z39.50-tjenere gjøres også tilgjengelige på World Wide Web ved hjelp av en web-gateway, noe som gir ytterligere ett lag.

Mellomleddet i en tre-lags arkitektur er vanligvis ikke implementert som é enkelt applikasjon. I mange tilfeller er dette laget implementert som mange delenheter som tar seg av forskjellige oppgaver. For å illustrere dette tar vi utgangspunkt i eksemplet over med bruk av en web-tjeneren som aksesserer en eller flere databaser. En web-tjener er i utgangspunktet bare et program som sender web-sider til en klient, og skal vi bruke denne web-tjeneren som inngangsport til en database, får vi web-tjeneren til å starte et nytt program som kommuniserer med databasen og sender data tilbake til web-tjeneren. En web-tjener representerer derfor ikke bare én prosess, men flere delprosesser som utfører forskjellige oppgaver. Vi har også nevnt at web-tjeneren kan være inngangsport til mange forskjellige databaser, og vi kan også bruke en web-tjener mot mange andre former for ressurser. På denne måten er tre-lags arkitekturer ofte egentlig en fler-lags arkitekturer bestående av mange separate komponenter som hver har sin rolle og spesielle funksjon i et overordnet system.

8.3 Komponenter og grensesnitt

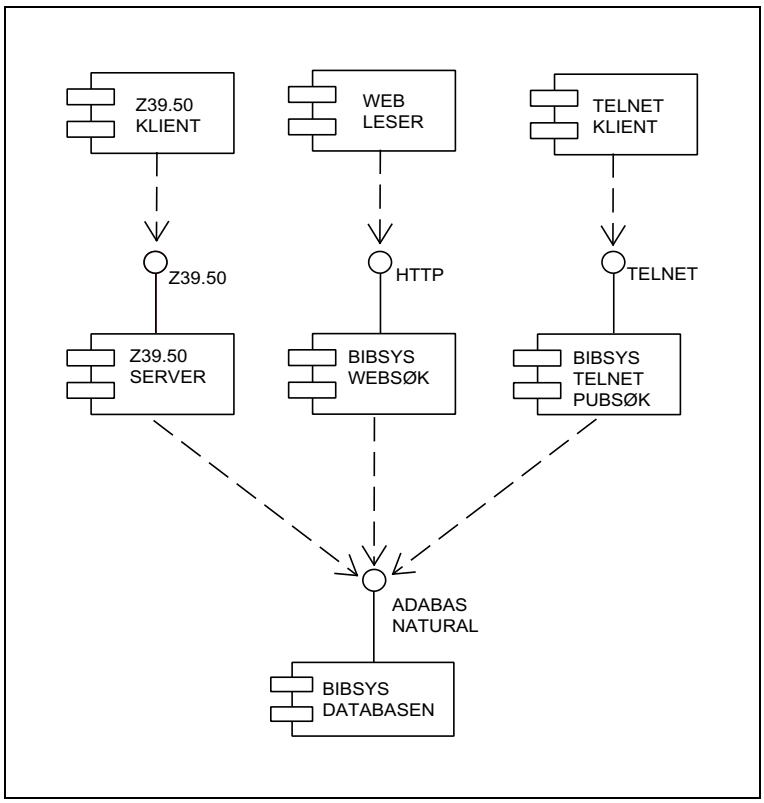
I fler-lags arkitekturen er det overordnete systemet splittet opp i mange delenheter som kommuniserer med hverandre. Slike delenheter av avgrensede og selvstendige programvare-enheter kalles ofte *komponenter*. Et *grensesnitt* er en komponents kontrakt med omgivelsene - dette er i praksis en spesifikasjon for hva komponenten kan gjøre, for eksempel hvilke funksjoner komponenten støtter og hvilke inndata og utdata som hører til funksjonene. Et grensesnitt er derfor mer eller mindre synonymt med en høy-nivå protokoll.

Mens protokoll vanligvis benyttes om standardiserte grensesnitt - gjerne mer generelle - er et grensesnitt som oftest knyttet til en bestemt tjeneste.

Vi kan ha komponenter av mange typer og størrelser. Komponenter kan være en måte å utvikle programvare på, ved at koden separeres i distinkte deler som kommuniserer med hverandre via definerte grensesnitt. Komponenter kan

være funksjonalitet vi kjøper fra andre for å integrere i det systemet vi selv utvikler (programvarebibliotek og drivere). Komponenter kan også være større selvstendige programmer og systemer vi kan kommunisere med via et definert grensesnitt. En komponent kan innkapsle både data og funksjonalitet, og en komponent kan implementere mange grensesnitt.

Komponent-løsningen gir mange fordeler. Ikke minst er bruken av komponenter og grensesnitt en velegnet abstraksjonsmekanisme for komplekse systemer. Grensesnitt-sentrert design gir mulighet for skalerbare og utvidbare arkitekturer. I fig. 8.2 viser vi deler av BIBSYS som komponenter og grensesnitt. Her er modelleringsspråket UML brukt. En boks er en komponent, og sirklene er grensesnitt. Hele linjer fra komponent til grensesnitt viser hvilke grensesnitt komponenten tilbyr (og implementerer). En klient kommuniserer med en tjener via grensesnittet - vist ved hjelp av stiplet linje. En komponent kan være tjener i én sammenheng, og klient i en annen sammenheng.



Figur 8.2: Komponenter og grensesnitt

Å definere et skille mellom distribuerte komponenter og distribuerte objekter er vanskelig. Dette skyldes antagelig at komponentbegrepet delvis har sitt utspring i maskinvare, mens objektbegrepet har sitt utspring i programmering. Brukt om kommuniserende programvareenheter i et distribuert miljø, er det i

realiteten lite som skiller disse to begrepene [143]:

> Components are standalone objects that can plug-and-play across networks, applications, languages, tools and operating systems. Distributed objects are, by definition, components because of the way they are packaged. In distributed systems, the unit of work and distribution is a component..... However, note that not all components are objects. Nor are they all distributed.....

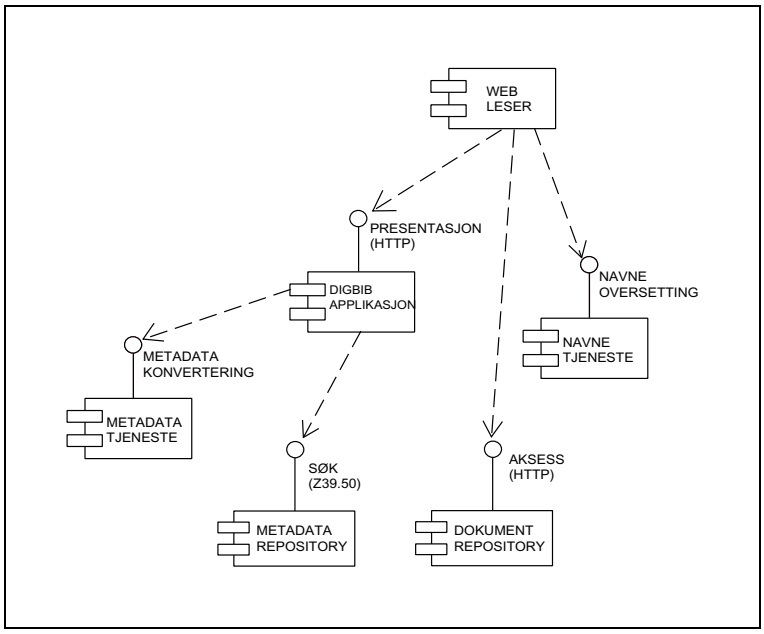
8.4 Komponentbasert arkitektur for digitale bibliotek

Det digitale bibliotek som et distribuert nettverk av informasjonskilder og tjenester, passer godt inn i en komponentbasert flerlags-arkitektur. I det digitale bibliotek har vi behov for et bredt spekter av tjenester for å publisere, forvalte og gjøre informasjon tilgjengelig, hvor en komponentbasert og grensesnittorientert arkitektur er nødvendig for å oppnå en fleksibel og åpen løsning.

Vi kan se på digitale bibliotek som en fler-lags arkitektur bestående av et globalt nettverk av komponenter. En komponent kan ha mange funksjoner, og kan best klassifiseres via de grensesnittene den kommuniserer med. Vi kan klassifisere grensesnittene i forhold til den funksjonen de har:

- **Applikasjoner**
  er grensesnitt mot større og mer fullstendige tjenester. Dette kan være komponenter som integrerer andre tjenester og tilbyr sammensatte brukerorienterte løsninger. Eksempler på dette er web-steder som samler og organiserer aksess til andre søkesystemer, eller det kan være mer enhetlige løsninger med integrert søking, adgangskontroll og aksess til dokumenter fra mange underliggende separate systemer.

- **Tjenester**
  tilbyr en avgrenset funksjonalitet. Dette kan være søking i en spesifikk bibliografisk database eller tilgang til digitale dokumenter fra en spesifikk dokumentbase. Videre kan dette være generelle fellestjenester som deles av alle eller mange av komponentene i digitale bibliotek. Det klassiske eksemplet på slike generelle tjenester er en identifikatortjeneste som oversetter fra logiske navn (URN) til lokatorer (URL). Andre generelle tjenester kan være katalogtjenester med informasjon om søkesystemer som er tilgjengelige, slik at man kan finne fram til de søketjenestene som er aktuelle, eller konverteringstjenester som konverterer mellom forskjellige metadataformater eller dokumentformater.



Figur 8.3: Generell komponent/grensesnitt-arkitektur for digitale bibliotek.

8.5 Tjenester

Modularisering i komponenter er ingen enkel oppgave. Målet er komponenter som både logisk og praktisk kan implementeres og driftes som selvstendige enheter i et nettverk, og som kan kombineres og gjenbrukes etter lego-kloss prinsippet, for å konstruere en samlet funksjonalitet tilpasset en brukergruppe eller et bruksområde. Utfordringen for digitale bibliotek basert på distribuerte komponenter, er å definere det sett av tjenester vi har behov for som selvstendige komponenter slik at vi oppnår den ønskede fleksibiliteten, uten at dette går utover stabilitet og effektivitet.

På enkelte områder er det for digitale bibliotek allerede definert slike tjenester ved at protokoller for spesifikk funksjonalitet er utviklet og i daglig bruk. På andre områder er det mindre erfaring med hvordan funksjonalitet skal modulariseres i tjenester, og det finnes ingen standardiserte spesifikasjoner for hvordan grensesnittene mot disse tjenester skal være. Eksempler på dette er håndtering av opphavsrett, betalingsmekanismer og lisens-/abonnements-tjenester.

I dette kapitlet tar vi for oss en rekke forskjellige funksjonelle aspekter ved digitale bibliotek og nevner eksempler på både standardiserte protokoller og eksperimentelle løsninger som kan videreutvikles til protkoller.

8.5.1 Søking

Søking er et viktig element for digitale bibliotek. For brukerne er dette det mest brukte verktøyet for å navigere i et stort og mangfoldig informasjonsrom. I kap. 7.6 beskriver vi enkelte protokoller for informasjonsgjenfinning, og vi nøyer oss her med å oppsummere.

- Z39.50 er den eneste av disse som er standardisert.
- HTTP støtter ikke søking direkte, men det er mulig å implementere ad-hoc løsninger for søking, som evt. kan beskrives formelt ved hjelp av WIDL [187].
- DASL er et utkast til en standard som baserer seg på å standardisere utvidelser til HTTP som støtter søking.
- DLIOP representerer bruk av distribuerte objekter og kan brukes som en overbygning på de fleste søkesystemer.
- STARTS er en protokoll som standardiserer en rekke elementer som er viktige for multi-søking.

8.5.2 Lagring og tilgang til informasjon

Informasjonsobjekter som dokumenter og metadataposter kan lagres og organiseres ved hjelp av mange forskjellige lagringssystemer, enten disse er basert på bruk av filer og kataloger, eller databasesystemer. For å gi eksterne brukere aksess til disse informasjonsobjektene, må vi gjøre disse tilgjengelige via grensesnitt som tilfredstiller krav vi har til slik aksess. Dette kan være at vi ønsker å bruke etablerte protokoller for å oppnå en mest mulig brukervennlig løsning, krav til adgangskontroll, eller andre former for kontroll over lageret med informasjonsobjekter. Velger vi å lagre og administrere informasjonsobjektene som filer i et filsystem, kan vi bruke en HTTP- eller FTP-tjener eller for den saks skyld et nettverks-filsystem. Valgmulighetene på dette området er mange. Vi kan også velge å bruke en database for å lagre metadata og dokumenter, og tilsvarende for denne løsningen er det mange valgmuligheter. Vi kan bruke en HTTP-tjener og implementere HTTP-baserte grensesnitt for aksess til databasens innhold, eller vi kan velge å bruke standardiserte protokoller for aksess til databaser over nett. Også her er valgmulighetene mange, og det finnes ingen fasit for hvilken løsning som er mest velegnet, men dette må være en avveining basert på krav og behov for både ekstern og lokal tilgang og kontroll.

I digitale bibliotek er det på den ene siden behov for løsninger som er tilpasset dette domenets behov for å gjøre informasjonsobjektene tilgjengelig for publikum eller for de brukergruppene som skal ha tilgang, men samtidig har slike tjenester en annen side ved at store samlinger av informasjon skal organiseres og administreres på enklest mulig måte. En generell trend er at metadata

lagres i databaser, mens dokumenter lagres som filer og aksesseres av publikum enten via HTTP eller FTP. Et generelt krav til lagringssystemer er likevel at de enkelte informasjonsobjekter må være identifiserbare enkeltobjekter med unik identifikator og adresse som kan benyttes i forvaltning av disse objektene i et nettverksmiljø.

World Wide Web er i stor grad basert på at informasjonen er fritt tilgjengelig og at publikum henter ut dokumenter via FTP eller HTTP. Dette er en måte å hente ut informasjon på som bare dekker deler av den informasjonen som er tilgjengelig. Informasjon kan være adgangskontrollert, enten fordi brukerne må betale for informasjonen eller fordi informasjonen av andre grunner kun skal være tilgjengelig for spesifikke brukere elle brukergrupper. Mye informasjon eksisterer også kun som fysiske eksemplarer i enkeltbiblioteker, hvor adgang til dokumenter er avhengig av lånebestillinger og fysiske forsendelser. Det å hente ut dokumenter er derfor en mangeartet prosess hvor det ikke bare vil være ett grensesnitt, men finnes flere grensesnitt. Eksempler på åpne løsninger for dokumentaksess er :

- HTTP og FTP er begge protokoller som kan brukes for å hente ut dokumenter med mulighet for adgangskontroll.
- Z39.50 og bruk av "extended services".
- ISO 10160/10161 - En ISO-standard for fjernlån, bestilling av lån fra andre bibliotek [98, 97].

8.5.3 Identifikatortjenester

Til oppretting og bruk av identifikatorer kan vi ha forskjellige tjenester. Dette er diskutert i kap. 5, og her tar vi kun med en oppsummerende liste:

- Generering av (unike) identifikatorer
- Identifikator-registre for registrering av metadata om de identifiserte enhetene.
- Administrative systemer for å registrere, endre og slette identifikatorer og informasjon om identifikatorer.
- Oversettertjenester som kan oversette fra en identifiaktor til en annen, eller fra en identifikator til lokaliseringsinformasjon.

8.5.4 Metadatatjenester

I et digitalt bibliotek med informasjonsressurser som administreres og forvaltes av mange forskjellige aktører, vil det være et stort behov for interoperabilitet også for metadata. Selv om bibliotekverdenen på overflaten virker konform og

kan skilte med MARC-formatet som en felles standard, er det også her interoperabilitetsproblemer siden det finnes mange forskjellige varianter av MARC-formatet. Tar vi hensyn til at det digitale bibliotek også skal romme andre typer virksomheter enn tradisjonelle bibliotek; museer, arkiv, utgiver, private virksomheter m. fl., er det et åpenbart behov for samvirke mellom forskjellige metadataformater.

Det finnes allerede noen løsninger som kan bidra til interoperabilitet på dette området, men dette er og vil antagelig fortsette å være en av de større utfordringene for digitale bibliotek.

I hvor stor grad det er mulig å konvertere fra ett metadataformat til et annet, kan diskuteres. I de fleste tilfeller vil dette føre til at noe informasjon går tapt, enten ved at informasjon fra et format ikke lar seg innplassere i et annet, eller ved at man mister informasjon pga. forskjellig feltinndeling o.l. Konvertering mellom MARC-formater gjøres allerede i dag, f.eks. ved hjelp av USEMARCON [109, 76] som er utviklet under EUs "Telematics Applications Programme (DGXIII-E)". En eksperimentell tjeneste for konvertering mellom fra Dublin Core til MARC er utviklet under det nordiske metadataprosjektet [165].

Vi kan også betrakte enkelte former for identifikatortjenester som metadatatjenester. NBN-tjenesten som NBR driver, kan karakteriseres som en form for metadatatjeneste, fordi den genererer en unik identifikator som kan benyttes i en metadatapost eller som metadata integrert i et dokument.

8.5.5 Andre tjenester

Annen funksjonalitet som kan defineres som egne tjenester er:

- Adgangskontroll for å kontrollere hvilke brukere som skal ha adgang til hvilke tjenester eller dokumenter. En relevant løsning for dette er bruk av proxy-tjener i kombinasjon med katalogtjenester som LDAP [189].
- Betalingsformidling for å samordne og forenkle betaling for tjenester og informasjon.
- Brukerprofil-tjenester for å kunne samle informasjon om den enkelte brukers behov for informasjon eller tjenester. Denne brukerprofilen kan f.eks. benyttes for å tilpasse brukergrensesnitt og informasjonstilbud til den enkelte bruker. Også her er det mulig å bruke en katalogtjeneste som LDAP.
- Kontroll av opphavsrett for å kontrollere plagiat, eller sørge for at rettighetshavere til informasjon mottar de vederlag de har rett på.
- Tjenester som er spesifikke for forskjellige mediatyper, f.eks. "streaming video" som er videoavspilling hvor vi ikke mottar en fil som deretter spilles av, men hvor vi mottar en videostrøm som kan spilles av etter hvert som data overføres.

- I tillegg vil det kunne defineres en rekke tjenester som er spesifikke for enkelte bruksområder.

Denne listen over funksjonalitet som kan modulariseres i tjenester, er selvsagt ikke endelig, men er mer ment som eksempler. Hvilke tjenester som skal utvikles og drives, og hvilke avgrensinger som skal gjøres for de forskjellige tjenester, vil avhenge av mange faktorer. Denne modulariseringen kan være en langsom prosess som både må baseres på praktiske erfaringer og utviklingen av standarder.

8.6 Et eksempel – Dienst

Dienst [113, 40] er en forkortelse for *a Distributed Interactive Extensible Network Server for Techreports*, og systemet har sitt utspring i et prosjekt ledet av *CNRI* kalt *The Computer Science Technical Reports Project*. Fokus for dette prosjektet var tilgang over nett til samlinger av tekniske rapporter innen fagområdet datateknikk. Dienst-systemet som ble utviklet i kjølvannet av dette prosjektet, er i dag basis for flere digitale bibliotek på Internett, bla. NCSTRL[1] og ETRDL[2].

Dienst er basert på et sett av individuelle tjenester som når de kombineres utgjør et distribuert digitalt bibliotek. Med distribuert menes at tjenestene som inngår i det digitale biblioteket kan være lokalisert hvor som helst på Internett. Funksjonaliteten i Dienst er i hovedtrekk lagring, søking og aksess til en distribuert samling av dokumenter, nærmere bestemt tekniske rapporter som omhandler datateknikk.

Dienst er egentlig tre ting:

- En konseptuell arkitektur som beskriver den logiske oppbygningen av systemet
- En protokoll som spesifiserer kommunikasjonen mellom de forskjellige tjenestekomponentene i et Dienst-basert system
- En programvare (et system) som implementerer protokollen og som er fritt tilgjengelig

Dienst-arkitekturen
Dienst er basert på en underliggende konseptuell arkitektur for digitale bibliotek, bestående av en dokumentmodell og en tjenestemodell.

[1] NCSTRL er en forkortelse for *the Networked Computer Science Tecnical Research Library* og uttales "ancestral". NCSTRL er en internasjonal samling av artikler og tekniske rapporter om datateknikk som er fritt tilgjengelige fra en rekke deltagende arkiv og institusjoner. En stor del av de deltagende institusjoner er universiteter.

[2] ETRDL er en europeisk forgreining av NCSTRL, og forkortelsen står for *the ERCIM Technical Reference Digital Library*. ERCIM er det europeiske forskningskonsortiet for informatikk og matematikk (the European Research Consortium for Informatics and Mathematics), og ETRDL er en aktivitet under DELOS Working Group.

- **Dokumentmodellen** definerer en rekke egenskaper og aspekter ved dokumenter; unike navn ved hjelp av hendler (The Handle System – se kap. 5.6.6), muligheten for multiple versjoner av samme dokument, og muligheten for logisk organisering av dokumentene i en hierarkisk struktur av subdeler (f. eks. kapitler og avsnitt). Det er denne logiske strukturen som er synlig gjennom Dienst-protokollen, slik at Dienst er uavhengig av hvordan dokumenter fysisk er lagret, enten dette er i databaser eller filbasert.

- Tjenestemodellen er det sett av individuelle tjenster som utgjør Dienst systemet. Tjenester som inngår i Dienst-protokollen og er implementert i Dienst systemet er:

  – En lagertjeneste hvor digitale dokumenter kan lagres og aksessere i henhold til dokumentmodellen.

  – En indekstjeneste som tar imot søk og returnerer en liste av dokumenter som tilfredstiller søkekriteriene.

  – En søkeformidlingstjeneste (Query Mediator) som videresender søk til egnede indekstjenester.

  – En infotjeneste som returnerer informasjon om tilstanden til en tjeneren som huser en eller flere Dienst-tjenester.

  – En samlingstjeneste som gir informasjon om hvordan et sett av tjenester inngår i en logisk samling.

  – En registertjeneste som lagrer informasjon om brukere.

  I tillegg er det implementert en rekke tjenester i Dienst-programvaren som ikke er spesifisert som en del av Dienst-protokollen, men som like fullt utgjør en del av den konseptuelle arkitekturen. Disse tjenestene skal kunne implementeres ad-hoc, f.eks. ved spesialtilpassing av et system til en spesifikk brukergruppe. Et grafisk brukergrensesnitt er en selvsagt komponent i en slik arkitektur.

Dienst-protokollen

Kommunikasjon mellom de forskjellige komponentene i Dienst skjer via Dienst-protokollen [41]. Denne protokollen er basert på HTTP uten å innføre utvidelser. Meldinger som sendes til en tjeneste (en forespørsel) uttrykkes som en URL ved hjelp av stidelene og søkedelen i URL, og bruk av HTTP GET-metoden. For overføring av dokumenter og i enkelte andre tilfeller, brukes også HTTP POST til forespørsler.

```
Eksempel:
http://bar.com/Dienst/Repository/1.2/Shred?delay=9&amperage=7.4.
```

En respons på en forespørsel som går fra en tjeneste, er formatert som en HTTP respons-melding med data i meldingsinnholdet og de nødvendige meldingsoverskriftene. Medietypen i Content-type-overskriften indikerer responstype:

- text/plain – ren tekst – brukes for responsmeldinger som inneholder ustrukturert informasjon.
- text/xml – xml-dokument – brukes for responsmeldinger som inneholder strukturert informasjon i henhold til en definert DTD.
- text/html benyttes for responser til brukergrensesnitt eller når HTML-innhold skal vises.

Dienst-protokollen benytter ellers et subsett av de samme statuskoder som er definert i HTTP.

Dienst-programvaren
Siden Dienst-protokollen ikke baserer seg på utvidelser til HTTP, men er definert som en spesialisert anvendelse av HTTP, er det mulig å implementere Dienst-tjenester ved hjelp av ordinær web-tjener-programvare. Dienst-systemet [42] som er utviklet, er basert på skriptspråket Perl, og er designet for web-tjeneren Apache. Dienst-systemet, Perl og Apache er alle fritt tilgjengelig programvare.

Programvaren inneholder de tjenestene som er definert i protokollen, men det er også implementert brukergrensesnitt før søking og "browsing".

Tillegg A

Typer av standarder

Standarder er dokumenterte avtaler som inneholder tekniske spesifikasjoner og andre presise kriterier. Disse brukes konsistent som regler, retningslinjer eller definisjoner av karakteristika, for å sikre at materiale, produkter, prosesser og tjenester er tilpasset sitt formål [85]. Det finnes en rekke virksomheter, organisasjoner og sammenslutninger som utvikler eller vedtar slike spesifikasjoner. Standarder karakteriseres ofte som *de jure*, *de facto*, eller *proprietære* [71]:

- **Industristandarder** er spesifikasjoner som eies av enkelt-virksomheter og er basert på faktiske, normdannende produkter. Disse kalles også for proprietære standarder (proprietær betyr "merkebeskyttet" eller "eid av"). Slike standarder kan være "åpne", men fordi standarden kontrolleres av en kommersiell virksomhet vil de ofte binde brukerne til en og samme produsent. Microsofts Windows-operativsystem er et eksempel på en slik standard. Spesifikasjonene for Windows utvikles av Microsoft alene, og de indre mekanismene (programkoden) er ikke tilgjengelig for resten av industrien som er begrenset til å forholde seg til det Microsoft velger å legge åpent (offentliggjøre).

- **De facto** standarder er spesifikasjoner som har oppnådd status som standard ved sin bruk eller oppslutning, og ikke gjennom formelle standardiseringsprosesser ("de facto" kan oversettes med "faktiske"). De facto standarder skiller seg fra industristandarder ved at de ikke er knyttet til enkelte kommersielle virksomheter, men ofte har utspring i utviklingsprosjekter i offentlig eller halvoffentlig regi. De skiller seg fra de jure standarder ved at de ikke er et resultat av en formell demokratisk standardiseringsprosess.

- **De jure** standarder kalles ofte formelle standarder. "De jure" kan oversettes med "etter loven" og disse standardene er utarbeidet og/eller vedtatt av standardiseringsorganisasjoner som er etablert ved lov av nasjonale myndigheter. Den internasjonale standardiseringsorganisasjon ISO *Internation Organization for Standardization* er sammenslutninger av nasjona-

le standardiseringsorganisasjoner hvor internasjonale standarder vedtaes ved avstemming blant medlemmene.

Industristandarder og de facto standarder som er utviklet, kan senere vedtaes av formelle standardiseringsorganisasjoner og dermed få status som "de jure" standard.

Det finnes mange nasjonale og internasjonale organisasjoner og sammenslutninger som utvikler og vedtar standarder. Hvorvidt standarder utviklet av slike organisasjoner skal kategoriseres som de jure eller de facto, er mer et spørsmål om de har vært igjennom en demokratisk standardiseringsprosess enn organets forankring i lovbestemte standardiseringsorganer. De facto og de jure karakteriserer to forskjellige aspekter ved standarder, og det er derfor vanskelig å trekke et presist skille mellom disse.

Mens industristandarder er et resultat av et produkt, og de facto standarder mer er en bieffekt av et prosjekt, er de jure standarder utviklet uavhengig av faktiske produkter.

Tillegg B

Organisasjoner

B.1 ANSI

American National Standards Institute[1] er det amerikanske (les USA) administrerings- og koordineringsorganet for det private standardiseringssystemet i USA. ANSI utvikler ikke selv standarder, men støtter standardisering ved å etablere konsensus om standardforslag som er utviklet av andre organisasjoner, bl.a. NISO.

B.2 CENL

Conference of European National Libraries[2] er en stiftelse som arbeider for å forsterke nasjonalbibliotekenes rolle i Europa. CENL arbeider med vedlikehold og oversikt over Z39.50-profiler, og driver også et prosjekt ved navn CoBRA+ (Computerised Bibliographic Record Actions). CoBRAs mål er blant annet å spre informasjon om pågående prosjekter og deres resultater. Spesielt interessant er arbeidet angående bruk av metadata for elektroniske ressurser. Et slikt prosjekt under CoBRA er BIBLINK[3].

B.3 CNRI

Corporation for National Research Initiatives[4] ble etablert i 1986, og er en amerikansk non-profitt organisasjon som skal fremme forskning og utvikling av den nasjonale informasjonsinfrastrukturen. CNRI støtter og deltar i en rekke samarbeidsprosjekter mellom statlige foretak, universiteter og private organisasjoner, og har som mål å designe og implementere utvalgte infrastrukturelle komponenter. Eksempler på prosjekter i regi av CNRI:

[1] se http://web.ansi.org
[2] se http://renki.helsinki.fi/gabriel/en/cenl-general.html
[3] se http://www.ukoln.ac.uk/metadata/biblink/
[4] se http://www.cnri.reston.va.us/

- *The Handle System* – som er et globalt og distribuert system for navngiving og oversetting av navn (se kap. 5.6.6).
- *The Computer Science Technical Reports Project* – et distribuert digitalt bibliotek for tekniske rapporter (se kap. 8.6).
- *Digital Object Architecture* – et rammeverk for distribuerte digitale objekter og tjenester for disse (se kap. 4.1).
- *D-Lib* – et forum for forskning og utvikling av digitale bibliotek, som også utgir det elektroniske tidsskriftet D-Lib Magazine (se http://www.dlib.org).

CoBRA arbeider også med standardisering av tegnsett gjennom prosjektet CHASE[5], og med muligheten for å etablere et nettverk av nasjonale autoritetsregistre for navn gjennom prosjektet AUTHOR[6].

B.4 ECMA

European association for standardizing information and communication systems[7] er en internasjonal, men europeisk basert industriforening grunnlagt i 1961 og dedikert til standardisering av informasjons- og kommunikasjonssystemer. ECMA har publisert en rekke standarder blant annet for datapresentasjon (tegnsett og koding), datautveksling ved hjelp av fysiske medier (kassetter og optiske disker), og datasikkerhet. En relevant standard som er publisert av ECMA er ECMAScript [50], et standardisert skriptspråk for web-lesere basert på Netscapes Javascript og Microsofts JScript.

B.5 IFLA

International Federation of Library Associations[8] ble stiftet i 1927 og fungerer som paraplyorganisasjon for bibliotek og bibliotekorganisasjoner over hele verden. IFLA samarbeider tett med UNESCO. Blant kjerneområdene for IFLAs arbeid er utvikling av retningslinjer og standarder som kan bidra til utveksling og forståelse av bibliografisk informasjon (ISBD og MARC[9]). Et viktig IFLA-bidrag til standardiseringsarbeidet er utviklingen av det generelle MARC-formatet *UNIMARC*.

En studiegruppe i IFLA leverte i 1997 en rapport om krav til bibliografiske posters virkemåte (FRBR). Rapporten får en særskilt behandling i kapittel 3.7.

[5] Se http://renki.helsinki.fi/gabriel/cobra/chase.html
[6] se http://renki.helsinki.fi/gabriel/cobra/author.html
[7] se http://www.ecma.ch/
[8] se http://www.ifla.org/
[9] ISBD = Internatianl Standard for Bibliographic Description. MARC = MAchine Readable Cataloguing.

B.6 Internett-organisasjoner

- **The Internet Society (ISOC)**[10] er en internasjonal organisasjon for global koordinering av og samarbeide om Internett. Den består av medlemmer fra over over 150 land, og ble etablert i 1992 som et resultat av behovet for en uavhengig og verdensomspennende organisasjon som kunne fremme global utbredelse, standardisering og endringer av Internett. Internet Engineering Task Force, som er Internetts standardiseringsorganisasjon, utfører sitt arbeid under overoppsyn av ISOC.

- **The Internet Engineering Task Force (IETF)**[11] er et åpent og internasjonalt fellesskap av nettverksutviklere, nettverksoperatører, leverandører og forskere som er opptatt av utvikling og drifting av Internett. Dette er den delen av Internett som utvikler og implementerer protokoller. IETF utfører sitt arbeid gjennom mange arbeidsgrupper. Disse er organisert i flere utviklingsområder som ledes av område-direktører. Områdedirektørene utgjør The Internet Engineering Steering Group (IESG), som både er ansvarlig for teknisk ledelse av IETF aktiviteter og Internetts standardiseringsprosess.

  IETF utgir en rekke publikasjoner som dokumenterer og informerer om utviklingen av Internett; *RFC* og *Internet-drafts*. RFC er en forkortelse for *request for comments* og er en arkivert serie av dokumenter relatert til Internett-standarder, både spesifikasjoner for selve standardene og andre dokumenter som er relevante – både informative og beskrivelser av eksperimentelle løsninger. *Internet-drafts* er arbeidsdokumenter for IETFs områder og arbeidsgrupper, og er første trinn mot publisering som RFC. En Internet-Draft er bare tilgjengelig i 6 måneder. Etter denne tiden må de erstattes av en ny versjon, publiseres som RFC, eller bli fjernet. Også andre grupper kan distribuere dokumenter som Internet-drafts.

- **The Internet Engineering Steering Group (IESG)**[12] er ansvarlig for teknisk ledelse av IETF-aktiviteter og Internetts standardiseringsprosess. ISEC består av områdedirektørene fra IETF-utviklingsområder, og er direkte ansvarlig for Internetts standardiseringsarbeid, og er den instans som endelig godkjenner en spesifikasjon som en Internett-standard.

  En teknisk spesifikasjon som skal utvikles til en standard av IESG, publiseres som RFC og vil ha forskjellig status etter hvert som den utvikles; foreslått standard (Proposed Standard), standardutkast (Draft Standard) og Internett-standard (Internet Standard). Spesifikasjoner som får status som Internett-standard skal være teknisk modne, og det kreves en betydelig implementering og operasjonell erfaring. I tillegg må det være

[10] se http://www.isoc.org
[11] se http://www.ietf.org
[12] se http://www.ietf.org/iesg.html

en generell oppfatning av at dette er en protokoll eller tjeneste som er fordelaktig for Internett.

En oversikt over spesifikasjoner som har status som Internett standard publiseres jevnlig som RFC. I RFC2600 "Internet Official Protocol Standards" finnes en utlisting av Internett-standarder pr. mars 2000. Protokoller som har status som standarder, er blant andre IP og TCP, FTP, TELNET. HTTP versjon 1.0 har ikke status som standard, mens HTTP versjon 1.1 foreløpig kun har status som foreslått standard.

- **Internet Assigned Numbers Authority (IANA)**[13] er ansvarlig for alle "unike parametre" på Internett, inklusive IP-adresser (Internet Protokoll adresser). IANA er også den ansvarlige instansen som endelig godkjenner URI-skjemanavn (se kap. 5.6.4) og URN-navneromsidentifikatorer (se kap. 5.6.4).

B.7 The International DOI foundation

DOI (Digital Object Identifier) er et identifikatorsystem som er opprettet for å ivareta opphavsrett for elektroniske publikasjoner. Initiativet til DOI ble tatt av den amerikanske forleggerforeningen (AAP), men er nå etablert som en selvstendig, internasjonal non-profitt organisasjon, *The International DOI foundation*[14]. Se nærmere om DOI i kapittel 5.

B.8 ISO

The International Organization for Standardization[15] er en verdensomspennende sammenslutning av nasjonale standardiseringsorganisasjoner fra rundt 130 land – én organisasjon fra hvert land.

ISO[16] utvikler et bredt spekter av standarder med målsettingen å støtte internasjonal utveksling av produkter og tjenester, og å støtte samarbeid både for intellektuelle, vitenskapelige, tekniske og sosiale aktiviteter.

Det tekniske arbeidet som ISO utfører, er desentralisert i et hierarki av tekniske komiteer, underkomiteer og arbeidsgrupper. TC 46 er den tekniske komiteen som utvikler informasjons- og dokumentasjonsstandarder, SC 9 er underkomiteen av TC46 som utvikler ISO standarder for presentasjon, identifikasjon og beskrivelse av dokumenter. Det er denne underkomiteen som har utviklet standarder som *ISO 2108:1992 Information and documentation – International standard book numbering (ISBN)* og *ISO 3297:1998 Information and documentation – International standard serial number (ISSN)* .

[13] se http://www.iana.org
[14] se http://www.doi.org/
[15] se http://www.iso.ch
[16] ISO er ikke et akronym for organisasjonens navn, men stammer fra det greske ordet "isos" som betyr "er lik".

Eksempler på andre ISO-standarder som er relevante i denne rapporten, er I*SO 8879:1986 Information processing – Text and office systems – Standard Generalized Markup Language (SGML)*, *ISO 23950:1998 Information and documentation – Information retrieval (Z39.50) – Application service definition and protocol specification* – bedre kjent som Z39.50-protokollen, *ISO 10161-1:1997 Information and documentation – Open Systems Interconnection – Interlibrary Loan Application Protocol Specification* som er en protokoll for fjernlån, og mange flere.

B.9 LC

Library of Congress i Washington har vært ledende i utviklingen av MARC-formatet - USMARC. Den nyeste utvikling på dette området er representert ved MARC21-formatet som er et samarbeid mellom LC og National Library of Canada.

LC er også vert for vedlikehold av ulike standarder som Z39.50 og ISO 639-2 (Språkkoder)

B.10 NISO

The National Information Standards Organization[17] er en amerikansk non-profitt sammenslutning som utvikler og fremmer tekniske standarder relatert til informasjonstjenester. NISO er den eneste organisasjonen som er godkjent av ANSI for å initiere, utvikle og vedlikeholde tekniske standarder for informasjonstjenester, bibliotek, utgivere o.l.

NISO har utviklet en rekke standarder, bl.a. *Z39.50 Information Retrieval (Z39.50)* og *Z39.56 Serial Item and Contribution Identifier (SICI)*. Andre aktuelle standarder som er under utvikling, er *Dublin Core Metadata Set*, *Book Item and Component Identifier (BICI)*, og *Syntax for the Digital Object Identifier*.

B.11 The Unicode Consortium

The Unicode Consortium[18] ble dannet i 1990, og er en non-profitt organisasjon for å fremme bruken av Unicode-standarden som et internasjonalt system for koding av informasjon. The Unicode Consortium samarbeider med ISO i utviklingen av et multi-byte tegnsett for alle verdens språk, og Unicode standarden [167] er samordnet med ISO-standarden ISO/IEC 10646 [104].

Medlemskap i The Unicode Consortium er åpent for alle enkeltpersoner og organisasjoner som ønsker å støtte bruk og videre utvikling av Unicode-standarden. Organisasjonen finansieres av medlemskapsavgift.

[17] se http://www.niso.org

[18] se http://www.unicode.org

B.12 The World Wide Web Consortium

The World Wide Web Consortium[19] (forkortes: W3C) ble opprettet for å lede utviklingen av World Wide Web ved å utvikle felles protokoller og sikre interoperabilitet. W3C er et internasjonalt industrikonsortium med over 330 medlemsorganisasjoner, og drives av MIT LCS (MIT Laboratory for Computer Science) i USA, INRIA i Frankrike, og Keio Universitetet i Japan. Tjenester som W3C yter, er et arkiv av informasjon om World Wide Web for utviklere og brukere, referanseimplementasjoner som demonstrerer og reklamerer for web-standarder, og diverse prototyper som illustrerer bruk av ny teknologi.

W3C publiserer en rekke forskjellige typer av tekniske rapporter. Høyeste nivå i formaliseringsarbeidet er dokumenter som får status som W3C Recommendation (W3C anbefaling). Dette er dokumenter som har konsensus i W3C, og som er godkjent av leder i W3C. Ideer og teknologi som spesifiseres i disse, er av W3C ansett for å være egnet til å tas i bruk, og er i overenstemmelse med W3Cs oppgaver. Før et dokument får denne statusen, har det vært igjennom flere trinn fra arbeidsutkast (Working Draft), kandidat til anbefaling (Candidate Recommendation), og foreslått anbefaling (Proposed Recommendattion). HTML (Hypertekst Markup Language) og XML (Extensible Markup Language) er spesifikasjoner som er publisert som W3C-anbefalinger. W3C publiserer også en rekke notater (W3C Notes) som er daterte forslag og ideer. Publisering av et notat innebærer ingen forpliktelser til oppfølging fra W3Cs side.

[19] se http://www.w3c.org

Tillegg C

Metadata - formateksempler

I dette tillegget gjengir vi eksempler på hvordan forskjellige typer metadata håndteres.

C.1 MARC

ISO 2709
For leselighetens skyld er markering for feltslutt satt lik '#', og delfeltmarkering er '$'. Vanligvis er disse bare en kode som ikke er representert av noe vanlig skrifttegn. Posten er her delt opp i linjer, men må ses på som en lang linje uten linjeskift. Posten er logisk to-delt: den første delen er innholdsfortegnelsen for den siste. Innholdsfortegnelsen kan ses som en rekke tall som 12 og 12 angir feltkode, startpunkt og lengde for hvert enkelt felt.

```
01187nam  2200337   450000100100000000800410001001500140005102000030006 5
08200150006809500100008310000200009324501110011325000140022426000420023 8
30000200028050000160030050000470044050000700048750001240031670000380055 7
71000240059571000500061974000190066985200160068885200190083085200210075 9
85200240080685200260073385200260078085200290070 4#982320647#
                      nor  #  $anf9906588#  #  $a839.822[S]#  $aS 13d#
$aTraavik, Morten#  $aKaptein Marlows testamente :$b(Mørkets hjerte) /$c
av Morten Traavik ; fritt etter romanen av Joseph Conrad#  $aVersjon 3#
 $aBergen :$bDen nationale scene ,$c1998#  $a43 bl. /$c30 cm#  $aTeaterm
anus#  $a"Manus er basert på den norske oversettelsen av Sigurd Hoel og
den svenske oversettelsen av Margaretha Odelberg" - S. 1#  $aRomanens or
iginaltittel: Heart of darkness#  $aOppført i Bergen museum, De naturhis
toriske samlinger, våren 1998#  $aConrad, Joseph$tHeart of darkness#  $a
Den Nationale scene#  $aBergen museum .$bDe naturhistoriske samlinger#
$aMørkets hjerte#  $aNBR$bNBR/PL#  $aUBIT$bGUNNERUS qB$c34475#  $aUBTØ$b
UBTØ mag$c20720#  $aNBO$bNBO$cqTS 30#  $aNBO$bNBO$cqSmåtr. 307#  $aUBB$b
UBB$cS box 434#  $aNBR$bNBR/DEP p#
```

Linjeformat.
Den samme posten som over, presentert i en mer leselig form. Stjerne (*) etterfulgt av 3-tegns tallkode angir felt, og dollar ($) etterfulgt av en bokstav angir delfelt.

```
*001982320647
*008                                      nor
*015  $a nf9906588
*020  $b h.
*082ga$d 839.822[S]
*086d $a S 13d
*100  $a Traavik, Morten
      $w Tråvik, Morten
*245  $a Kaptein Marlows testamente
      $b (Mørkets hjerte)
      $c av Morten Traavik ; fritt etter romanen av Joseph Conrad
*250  $a Versjon 3
*260  $a Bergen
      $b Den nationale scene
      $c 1998
*300  $a 43 bl.
      $c 30 cm
*500  $a Teatermanus
*500  $a "Manus er basert på den norske oversettelsen av  Sigurd Hoel og
        den svenske oversettelsen av Margaretha  Odelberg" - S. 1
*500  $a Romanens originaltittel: Heart of darkness
*500  $a Oppført i Bergen museum, De naturhistoriske samlinger, våren 1998
*700  $a Conrad, Joseph
      $t Heart of darkness
*710  $a Den Nationale scene
      $w Nationale scene
*710  $a Bergen museum
      $b De naturhistoriske samlinger
*740  $a Mørkets hjerte
```

MARC i XML
Eksemplet viser en post fra UBiTs bildedatabase i BIBSYS. Først kommer posten i XML-format. Starten på hvert felt er markert med en MARC-tag (3 siffer) prefikset med 'Post' (<Post008>) og avsluttet på tilsvarende måte (</Post008>). Delfeltene er market på samme måte (eksempel: <f> og </f>). Til slutt følger posten slik den ser ut i MARC linjeformat.

```
<BibMarc>
 <Record Type="Dataflex">
  <Post008><f>0</f>
  </Post008>
  <Post012><k>IJGR</k>
  </Post012>
  <Post088><a>UHJW, Trondheim</a>
  </Post088>
  <Post096><f>300 Positiv s/h</f>
  </Post096>
  <Post245><a>Lademoen i Trondheim, sett mot Ladehammeren</a>
  </Post245>
  <Post260><c>1892 - 1892</c>
  </Post260>
  <Post300><c>H 13.0 x B18.0</c>
  </Post300>
  <Post500><a>HORNEMAN-001</a>
  </Post500>
  <Post571><a>UBT-HO-001</a>
  </Post571>
  <Post687><a>Lademoen</a>
  </Post687>
  <Post700><a>Horneman, Johan Lebrecht</a>
           <c>Militær</c>
           <d>1868</d>
  </Post700>
  <Post740><a>Tilstand: Ny brukskopi</a>
  </Post740>
 </Record>
</BibMarc>


008 $f Fossestuen ved Nedre Leirfoss, eksteriør
012 $k IJGR
088 $a UHQ, Tiller
096 $f 300 Positiv s/h
245 $a Fossestuen ved Nedre Leirfoss, eksteriør
260 $c 1892 - 1936
300 $c H 13 x B 18
500 $a HORNEMAN-025
571 $a UBT-HO-025
687 $a Fossestuen
700 $a Horneman, Johan Lebrecht
700 $c militær
700 $d 1868
740 $a Tilstand: Ny brukskopi
```

C.2 VRA core

De følgende data beskriver en radering i en museumssamling og et digitalt bilde av raderingen. Feltnavnet angis først og starter alltid på ny linje. Innholdet av feltet skilles fra feltnavn med et likhetstegn (=).

```
Record Type = work
Type = print
Title = This is how it happened
Title.Variant = As Sucedi
Measurements.Dimensions = 24.5 x 35 cm
Material.Medium = ink
Material.Support = paper
Technique = etching
Technique = drypoint
Creator.Personal Name = Francisco Jose de Goya y Lucientes
Creator.Role = printmaker
Date.Creation = ca. 1810-1814
Location.Current Repository = Ann Arbor (MI,USA), University
of Michigan Museum of Art
Location.Creation Site = Madrid (ESP)
ID Number. Current Accession = 1977/2.15
Style/Period = Romanticism
Culture = Spanish
Subject = war
Relation.Type = Part of Disasters of war
Description = This is how it happened is No. 47 (33) from
the series "The Disasters of War", 4th edition, plates for
the series ca. 1810-14, 1820, 4th edition was published 1906.
Source = Gift of Mrs. Charles F. Weber to the University of
Michigan Museum of Art
Rights = Weber family trust

Record Type = image
Type = digital
Title = general view
Measurements.Dimensions = 72 dpi
Measurements.Format = jpeg
Technique = scanning
Creator = Fred Technician
Date.Creation = 1999
Location.Current Repository = Ann Arbor (MI,USA), University of
Michigan Museum of Art
ID Number.Current Repository = PCD5010-1611-1037-27
ID Number.Current Repository = 1977\_2.15.jpeg
Description = For more information, see \url{http://www.si.umich.edu/
Art_History/demoarea/details/1977_2.15.html}
Source = University of Michigan Museum of Art
Rights = University of Michigan Museum of Art
```

C.3 Dublin Core

HTML
Her viser vi hvordan Dublin Core-data kan angis i et vanlig HTML-dokument som en *META-tag* i dokumentets hode. Opplysningene kommer ikke fram på skjermen for brukeren. META-tagen har delfelt som angir hvilket felt det dreier seg om (NAME) og innholdet av feltet (CONTENT). For enkelte felt angis som kvalifikator også hvilke regler for utfylling som følges eller hvilket kontrollregister dataene er hentet fra (SCHEME).

```
<META NAME="DC.Date" CONTENT="1999-06-25">
<META NAME="DC.Title" CONTENT="Metadata">
<META NAME="DC.Creator.PersonalName" CONTENT="Husby, Ole">
<META NAME="DC.Subject" SCHEME="HUMORD" CONTENT="Katalogisering">
<META NAME="DC.Subject" SCHEME="UDC" CONTENT="025.32">
<META NAME="DC.Description" CONTENT="Innføring i metadata,
            med hovedvekt på Dublin Core. Inneholder litteraturhenvisninger.">
<META NAME="DC.Publisher" CONTENT="BIBSYS">
<META NAME="DC.Identifier" CONTENT="http://www.bibsys.no/BDB/metadata/">
<META NAME="DC.Type" CONTENT="Text.Article">
<META NAME="DC.Format" CONTENT="text/html">
<META NAME="DC.Language" SCHEME="Z39.53" CONTENT="NOR">
```

XML/RDF
Posten innledes av noen linjer som angir hvilke standarder som brukes. Deretter følger feltene som innledes av en kode for feltets navn (f.eks. <title>) og avsluttes med en motsvarende kode (</title>). Det er ikke noe krav at feltene skal begynne på ny linje, men for leselighetens skyld er det gjort her.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
         xmlns = "http://purl.org/dc/elements/1.0/">
 <rdf:Description about="http://purl.org/DC/index.htm">
   <title>Dublin Core Metadata Initiative - Home Page</title>
   <description>The Dublin Core Metadata Initiative Web site.</description>
   <date>1998-10-10</date>
   <format>text/html</format>
   <language>en</language>
   <contributor>The Dublin Core Metadata Initiative</contributor>
 </rdf:Description>
</rdf:RDF>
```

C.4 NKKM

Tabellen viser et komplett skjema for fotoregistrering ved Norges kunst- og kulturhistoriske museer (NKKM)[1].

FOTOTYPE:		TILV.NR.:		
REPRONR :		BILLEDLAGRING :		
	INNKOMST		REG:	. .
KLASS :				
DATERING:	FRA		TIL	
MOTIV :				
SIGN/PÅSKRIFT:				
TITTEL:				
STED:	KODE		STED	
	ADR		GNR	
NAVN		YRKE		
ALT:NAVN:				
	FØDT/ETABL.		DØD/NDL.	
	STED		ADR	
	OPPL			
FORMAT :	H		B	
MONTERING:				
BEVARINGSTILSTAND:				
ARKIVREFERANSE:				
LITT.REFERANSE:				
PRODUKTNUMMER:				
KLAUSUL:				
FAST PLASS:	BYGNING		ROM	
	VEGG/REOL			
	DATO	. .	SIGN	
MIDL. PLASS:				
RESTAURERING :				
UTF.OPPL:				

[1] se http://www.hd.uib.no/regimus/feltkode.html for nærmere beskrivelse av de enkelte feltene i dette skjemaet, samt skjemaer for registrering av annet materiale.

C.5 TEI-header

Eksemplet følger samme feltlogikk som under XML-eksemplene ovenfor, med innledende og avsluttende markeringer av felt. Denne posten inneholder også opplysninger om revisjoner av dokumentet, og om hvem som er ansvarlig for posten.

```
<teiHeader>
<fileDesc>
<titleStmt>
<title>The Story of Mankind [a machine-readable transcription]</title>
<author>Van Loon, Hendrik</author>
<respStmt>
<resp>Creation of machine-readable version: </resp>
<name>Charles Keller</name>
<resp>Creation of digital images: </resp>
<name>Electronic Text Center, University of Virginia Library</name>
<resp>Conversion to TEI.2-conformant markup: </resp>
<name>University of Virginia Library Electronic Text Center.</name>
</respStmt>
</titleStmt>
<extent>ca. 790 kilobytes </extent>
<publicationStmt>
<publisher>University of Virginia Library.</publisher>
<pubPlace>Charlottesville, Va.</pubPlace>
<idno>Modern English, VanStor</idno>
<availability>
<p>Publicly-accessible</p>
<p n=public>URL: \url{http://etext.lib.virginia.edu/modeng/modengV.browse.html}</p>
</availability>
<date>1997</date>
</publicationStmt>
<seriesStmt$><p$></p> </seriesStmt>
<notesStmt>
<note>Illustrations have been included from the print version.</note>
<note>Scanned by Charles Keller with OmniPage Professional OCR software.</note>
</notesStmt>
<sourceDesc>
<biblFull>
<titleStmt>
<title>The Story of Mankind</title>
<title level="m"$></title>
<author>Hendrik Van Loon</author>
<respStmt$><resp$></resp> <name$></name>
</respStmt>
</titleStmt>
<editionStmt>
<p$></p>
</editionStmt>
<extent$></extent>
<publicationStmt>
<publisher>Boni & Liveright, Inc.</publisher>
<pubPlace>New York</pubPlace>
```

```
<date>1921</date>
<idno>Print copy consulted: University of Virginia Library,
call number D21.V3 1921</idno>
</publicationStmt>
<seriesStmt$><p$></p$></seriesStmt>
<notesStmt>
<note>This book is from a gift of McGregor fund to the general
collection of the Alderman Library.</note>
</notesStmt>
</biblFull>
</sourceDesc>
</fileDesc>
<encodingDesc>
<projectDesc>
<p>Prepared for the University of Virginia Library
Electronic Text Center.</p>
</projectDesc>
<editorialDecl>
<p>All unambiguous end-of-line hyphens have been removed, and
the trailing part of a word has been joined to the preceding line.</p>
<p>Keywords in the header are a local Electronic Text Center
scheme to aid in establishing analytical groupings.</p>
</editorialDecl>
<refsDecl>
<p$></p>
</refsDecl>
<classDecl>
<taxonomy>
<bibl$><title>Library of Congress Subject Headings</title>
</bibl>
</taxonomy>
</classDecl>
</encodingDesc>
<profileDesc>
<creation$><date>1921</date>
</creation>
<langUsage$><language id="en">English</language>
</langUsage>
<textClass>
<keywords$><term>prose; non-fiction</term>
</keywords>
<keywords$><term>LCSH</term>
</keywords>
<keywords>
<term type="artist">Hendrik Van Loon (author)</term>
<term type="visual work">illustrations</term>
<term type="format">24-bit color; 400 dpi</term>
</keywords>
</textClass>
</profileDesc>
<revisionDesc>
<change>
<date>August 1997</date>
<respStmt>
```

```
<resp>corrector</resp>
<name>Anne Stinehart, Electronic Text Center</name>
</respStmt>
<item>Added TEI header and tags </item>
</change>
</revisionDesc>
</teiHeader>}
```

C.6 IAFA template

Eksemplet viser en post av SERVICE-typen i emneportalsystemet ROADS. Feltnavn angis alltid først på linja, og skilles fra innholdet i feltet med et kolon (:).

```
Template-Type: SERVICE
Title: Wellcome Unit for the History of Medicine
URI-v1: http://units.ox.ac.uk/cgi-bin/safeperl/wuhminfo/p?home.html
Publisher-Name-v1: Wellcome Unit for the History of Medicine
Publisher-Postal-v1: 45-47 Banbury Road, Oxford, OX2 6PE
Publisher-City-v1: Oxford
Description: The home page of the Wellcome Unit for the History of Medicine
Language-v1: English
Subject-Descriptor-v1: WZ40 History of Medicine
Subject-Descriptor-Scheme-v1: NLM
Handle: 71473886-23884
Record-Last-Modified-Date: Fri, 10 Oct 1997 19:09:16 +0000
Record-Last-Modified-Email: cataloguer\@omni.ac.uk
Record-Created-Date: Fri, 10 Oct 1997 19:09:16 +0000
Record-Created-Email: cataloguer@omni.ac.uk
Admin-Email-v1: wuhmo@wuhmo.ox.ac.uk
```

Tillegg D

FRBR i praksis

D.1 Entiteter og relasjoner i bibliografiske poster

Den modellen for bibliografisk informasjon som presenteres i rapporten *Functional requirements for bibliographic records* (FRBR) kan se enkel ut, men det kan være vanskelig å se konsekvensene når man ikke har katalogiseringserfaring, eller man kan bli skremt om man har det. Bibliografiske strukturer har det alltid vært i katalogene, f.eks. ved ensartet registrering av opplysninger, klassifikasjonskoder og henvisninger til andre relevante verk og utgaver. FRBR legger opp til en sterkere formalisering av disse strukturene. En sterkere formalisering vil gjøre det enklere å utnytte strukturen til navigering i det bibliografiske univers, men vanskeligere å lage forståelige presentasjonsmåter (brukergrensesnitt).

I de følgende avsnittene vil vi først se på en innfløkt bibliografisk struktur i lys av FRBR-modellen. Deretter vil vi se på hvordan de bibliografiske strukturene bygges opp over tid, og hvordan katalogiseringsarbeid på sikt vil bli redusert. Til slutt vil vi komme med noen forslag til framvisning på skjerm der den bibliografiske strukturen kommer mer til sin rett. Det er vårt håp at disse kan fungere som grunnlag for diskusjon.

D.1.1 Et teatermanus

Utgangspunktet er et teatermanus[1] av Morten Traavik (figur D.1). Manuset er laget fritt etter en roman av Joseph Conrad. Traaviks uttrykk er basert på to oversettelser av Conrads roman, den norske (av Sigurd Hoel) og den svenske (av Margaretha Odelberg). Innførselen for manifestasjonen av Traaviks verk er full av relasjoner til andre verk, uttrykk og personer. Noen av dem er formelt uttrykt (7XX-felt), andre er bare fritekstlig uttrykt (500-notene og 245 $c) og derfor vanskelig å utnytte i et datasystem. Det kan også være usikkerhet knyttet til 7XX-relasjonene fordi de er tekstlig uttrykt. Bruk av identifikatorer til aktuelle verk, korporasjoner og manifestasjoner ville fjernet usikkerheten. To produktentiteter i FRBR har *identifikator* som egenskap (eksemplar og manifestasjon), mens to nøyer seg med identifiserende opplysninger (mer om identifikatorer i kapittel 5).

Det er sannsynlig at det siktes til 1992-utgaven (figur D.2) av Sigurd Hoels oversettelse. Denne ble utgitt med tittelen *Mørkets hjerte*. Slår man opp på Hoels oversettelse, så ser man at den har kommet i en serie med tittelen *20. århundre* og at 1.utgaven kom i 1929 med tittelen *Det inderste mørke*. I Norske Avdelings seddelkatalog står tittelen gjengitt slik: *Ungdom. - Det inderste mørke* (to titler). Romanen er altså her manifestert sammen med

[1] En stor takk til Annema Hasund Langballe som satte oss på sporet av disse eksemplene, som forsynte oss med opplysninger fra Norsk bokfortegnelse og Norske Avdelings seddelkatalog, og som tålmodig dukket ned i Nasjonalbibliotekets djupe hvelv for å finne de norske manifestasjonene av *Heart of darkness* slik at vi fikk anledning til å sammenlikne dem.

Traavik, Morten, 1971-
Kaptein Marlows testamente : (Mørkets hjerte) / av Morten Traavik ; fritt etter romanen av Joseph Conrad. - Versjon 3. - Bergen : Den nationale scene, 1998. - 43 bl. ; 30 cm. - (839.822[S])
Teatermanus. - "Manus er basert på den norske oversettelsen av Sigurd Hoel og den svenske oversettelsen av Margaretha Odelberg" - S. 1. - Romanens originaltittel: Heart of darkness. - Oppført i Bergen museum, De naturhistoriske samlinger, våren 1998

```
*015 $a982320647$bbibsys
*020 $bh.
*0411 $hger
*08230$a839.822[S]
*098 $aa,a
*100 $aTraavik, Morten$d1971-
*24510$aKaptein Marlows testamente$b(Mørkets hjerte)$cav Morten
       Traavik ; fritt etter romanen av Joseph Conrad
*250 $aVersjon 3
*260 $aBergen$bDen nationale scene$c1998
*300 $a43 bl.$c30 cm
*500 $aTeatermanus
*500 $a"Manus er basert på den norske oversettelsen av Sigurd
       Hoel og den svenske oversettelsen av Margaretha Odelberg" - S. 1
*500 $aRomanens originaltittel: Heart of darkness
*500 $aOppført i Bergen museum, De naturhistoriske samlinger,
       våren 1998
*700 0$aConrad, Joseph$tHeart of darkness
*710 0$aDen Nationale scene$w4
*710 0$aBergen museum$bDe naturhistoriske samlinger
*7400 $aMørkets hjerte
```

Figur D.1: Mortens Traaviks verk (fra Norsk bokfortegnelse).

```
020 $a82-05-20923-5$bh.$cNkr 148.00
08230$a823[S]
1001 $aConrad, Joseph
2401 $aHeart of darkness
2451 $aMørkets hjerte$cJoseph Conrad ; oversatt av Sigurd Hoel
260 $aOslo$bGyldendal$c1992
300 $a148, [1] s.$c21 cm
440 0$a20. århundre
500 $aUtgitt første gang på norsk 1929 med tittel: Det inderste
      mørke
940 0$aTjuende århundre$z20. århundre
991 $aHoel, Sigurd
```

Figur D.2: Joseph Conrads verk i Sigurd Hoels uttrykk i 1992-utgaven (fra Norsk bokfortegnelse).

```
*100 0$aConrad, Joseph$cpsevd. for Joseph Conrad Korzeniow
     $d1857-1924$jeng.$322043300
*24510$aUngdom$cOvers. av Sigurd Hoel
*260 $aOslo$bGyldendal$c1929
*300 $a2 bl. + 195 s.
*500 $aJoseph Conrad er psevd. for Joseph Conrad Korzeniowski
*599 $axkl0420
*700 2$aConrad, Joseph$cpsevd. for Joseph Conrad Korzeniow
     $d1857-1924$jeng.$tDet inderste mørke$322043300
```

Figur D.3: Første utgave av Sigurd Hoels oversettelse fra 1929 (fra Deichmans katalog).

```
010 $a9905426n
015 $a990015009$bbibsys
020 $a82-05-25688-8$bh.$cNkr 75.00
08230$a823[S]
100 $aConrad, Joseph
2401 $aHeart of darkness
24510$aMørkets hjerte$cJoseph Conrad ; oversatt av Sigurd Hoel
250 $a3. utg.
260 $a[Oslo]$bGyldendal$c1999
300 $a148 s.
440 $aGyldendal klassiker
500 $a1. norske utg. 1929 med tittel: Det inderste mørke
7404 $aDet inderste mørke
991 $aHoel, Sigurd
```

Figur D.4: Joseph Conrads verk i Sigurd Hoels uttrykk i 1999-utgaven (fra Norsk bokfortegnelse).

et annet verk av Joseph Conrad. Det går frem av sidetallet at første tittel fyller sidene 1-50, mens *Det inderste mørke* fyller sidene 51-195. Posten ble funnet i elektronisk form i katalogen til Deichmanske bibliotek (se figur D.3).

Sigurd Hoels oversettelse ble utgitt på ny i 1999 (figur D.4), da i serien *Gyldendal klassiker*.

I figur D.5 har vi systematisert noen av entitetene og relasjonene som er gjengitt i figur D.1 og D.2 og som omfatter Traaviks teaterverk og Conrads romanverk. Vi har ikke tatt med eksemplarentitetene, men her kunne man f.eks. føye til de enkelte rolleheftene for verk 2s manifestasjoner som har forskjellige eiere, eksempelvis Nasjonalbiblioteket. Vi har inkludert noen få egenskaper, der vi mener de hører hjemme (mest for illustrasjonens skyld).

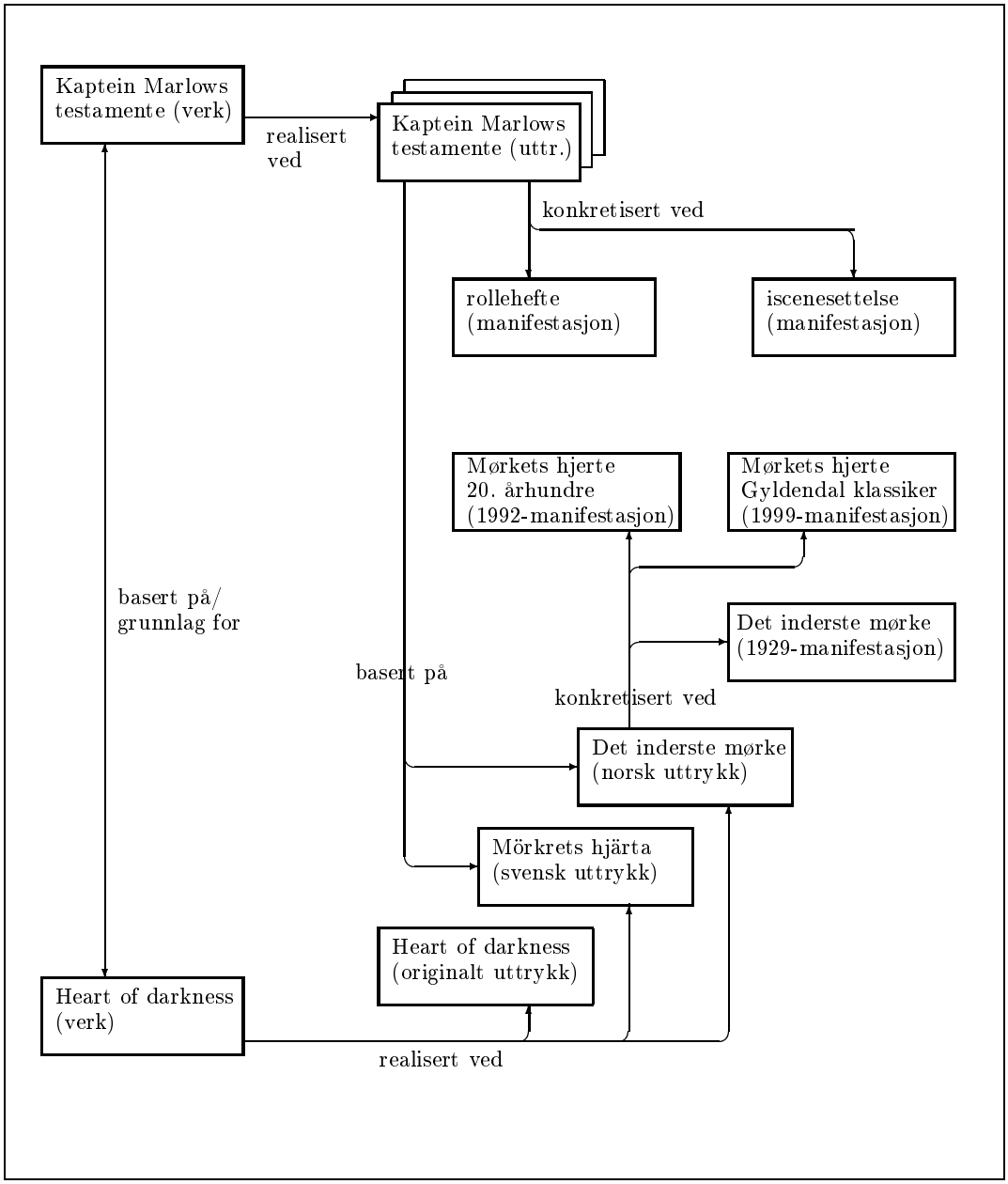
I figur D.6 har vi forsøkt å framstille dette grafisk, men har bare holdt oss til produktentitetene *verk*, *uttrykk* og *manifestasjoner*. Det er kanskje litt overdrevent å føre opp iscenesettelsen, men den står nevnt i posten fra Norsk bokfortegnelse. Det kan være et diskusjonstema om iscenesettelsen av Traaviks verk selv er et nytt verk (regissør og skuespillere vil kanskje hevde det). Vi overlater den diskusjonen til katalogavdelingene. Her regnes den som et nytt *uttrykk*. Det er forøvrig gjort flere filmatiseringer av *Heart of darkness*, men de tas ikke med her.

Det er en omstendelig affære å lage strukturen for relasjonene mellom de bibliografiske entitetene i ettertid. Det er derfor et poeng å se på hvordan denne strukturen hadde blitt

Verk 1 : Heart of darkness
skapt av Joseph Conrad
Uttrykk 1 : Det inderste mørke
realisert av Sigurd Hoel
Manifestasjon : 1929
produsert av Gyldendal Norsk Forlag
Manifestasjon : 1992
produsert av Gyldendal Norsk Forlag
del av serien 20.århundre
Egenskap : tittel : Mørkets hjerte
Manifestasjon : 1999
produsert av Gyldendal Norsk Forlag
del av serien Gyldendal klassiker
Egenskap : tittel : Mørkets hjerte
Uttrykk 2 : svensk oversettelse - ikke i NBF
realisert av Margaretha Odelberg
Egenskap : tittel : Mörkrets hjärta

Verk 2 : Kaptein Marlows testamente
skapt av Morten Traavik
har som emne 839.822[S]
basert på verk 1
Uttrykk 1 : teatermanus - versjon 3
realisert av Morten Traavik
basert på verk 1's uttrykk 1
basert på verk 1's uttrykk 2
Manifestasjon : Rollehefte
produsert av Den nationale scene
Egenskap : 43 bl., 30cm
Uttrykk 2: iscenesettelse
Manifestasjon : Oppføring i Bergen Museum
produsert av Den nationale scene
Egenskap: tidsrom : våren 1998

Figur D.5: Systematisk framstilling av noen av entitetene og relasjonene for *Heart of darkness* og *Kaptein Marlows testamente* (relasjoner er fetet). Å kalle Dewey-signaturen her et 'emne' er litt på kanten. Den er egentlig en oppstillingssignatur.



Figur D.6: Relasjonene mellom *Kaptein Marlows testamente* og *Heart of darkness*

bygd opp over tid, dvs. om man hadde brukt FRBR fra starten, og det fantes et system for universell bibliografisk kontroll med utstrakt utveksling av data internasjonalt.

D.1.2 Kronologien og strukturens utvikling

I 1902 utkom *Heart of darkness* for første gang. Siden Joseph Conrad tidligere har skapt andre verk, fins personentiteten for Joseph Conrad allerede i systemet. Den nye verkentiteten knyttes opp mot denne. Videre må man katalogisere de tre nivåene under: *uttrykk, manifestasjon* og *eksemplar*. Får man et nytt eksemplar, knyttes dette opp mot manifestasjonen (som allerede er registrert).

I 1929 kom Sigurd Hoels oversettelse av Joseph Conrads roman (figur D.3) med tittelen *Det inderste mørke*. Den ble publisert sammen med en annen historie av Joseph Conrad, *Ungdom*. Det må registreres nye uttrykk for hvert av de to verkene. Verk- og forfatterinformasjon ligger i systemet fra før eller kan hentes fra internasjonale kilder. Manifestasjonen inneholder to verk. Hvert verk er manifestert i denne boka. Personentiteten for Sigurd Hoel fins allerede. Denne må knyttes opp mot de to nye uttrykkene. Eksemplarer henges på ovennevnte manifestasjon.

I 1992 kom Sigurd Hoels oversettelse ut på ny, nå med tittelen *Mørkets hjerte*. Oversettelsen er gitt en moderne språkdrakt, og det kan da diskuteres om det er et nytt uttrykk eller en ny manifestasjon (*aa* er blitt til *å*, *indlægge* er blitt til *innlegge* osv.). Denne diskusjonen overlater vi igjen til katalogavdelingene. I denne framstillingen har vi valgt å betrakte utgaven som en ny manifestasjon. 1992-utgivelsen går inn i serien *20. århundre*. Denne nyutgivelsen koples på den forrige strukturen som en ny manifestasjon av Hoels oversettelse.

I 1998 kom Morten Traaviks teatermanus. På det tidspunktet da *Kaptein Marlows testamente* kom til katalogisering må man anta at hele strukturen for Conrads verk forelå (bortsett fra 1999-utgaven).

Foruten å beskrive det særegne ved Traaviks verk, må man generere relasjonene fra verk til verk og fra uttrykk til uttrykk. Det skal ikke mye fantasi til for å forestille seg at et datasystem kan gjøre dette til en meget enkel operasjon.

I 1999 kom en ny utgave av Sigurd Hoels oversettelse, denne gang i serien *Gyldendal klassiker*. Utgaven er et fotografisk opptrykk (noe forminsket) av 1992-utgaven, og må betraktes som en ny manifestasjon (dette kan også diskuteres). 1999-utgaven kan den knyttes opp mot det tidligere registrerte uttrykket. Utgaven finner dermed sin plass i hele strukturen. Man kan finne fram til Traaviks verk ved de relasjonene som allerede er etablert.

Poenget er altså at strukturen bygges opp gradvis. Man utnytter tidligere registrerte entiteter og relasjoner på forskjellige nivåer. Dagens katalogisering foregår på manifestasjonsnivå, verk og uttrykk registreres på ny for hver ny manifestasjon. Dette kan også føre til at man mister informasjon om ikke kontrollarbeidet i katalogiseringsprosessen er svært omfattende og tidkrevende.

D.1.3 Heart of darkness

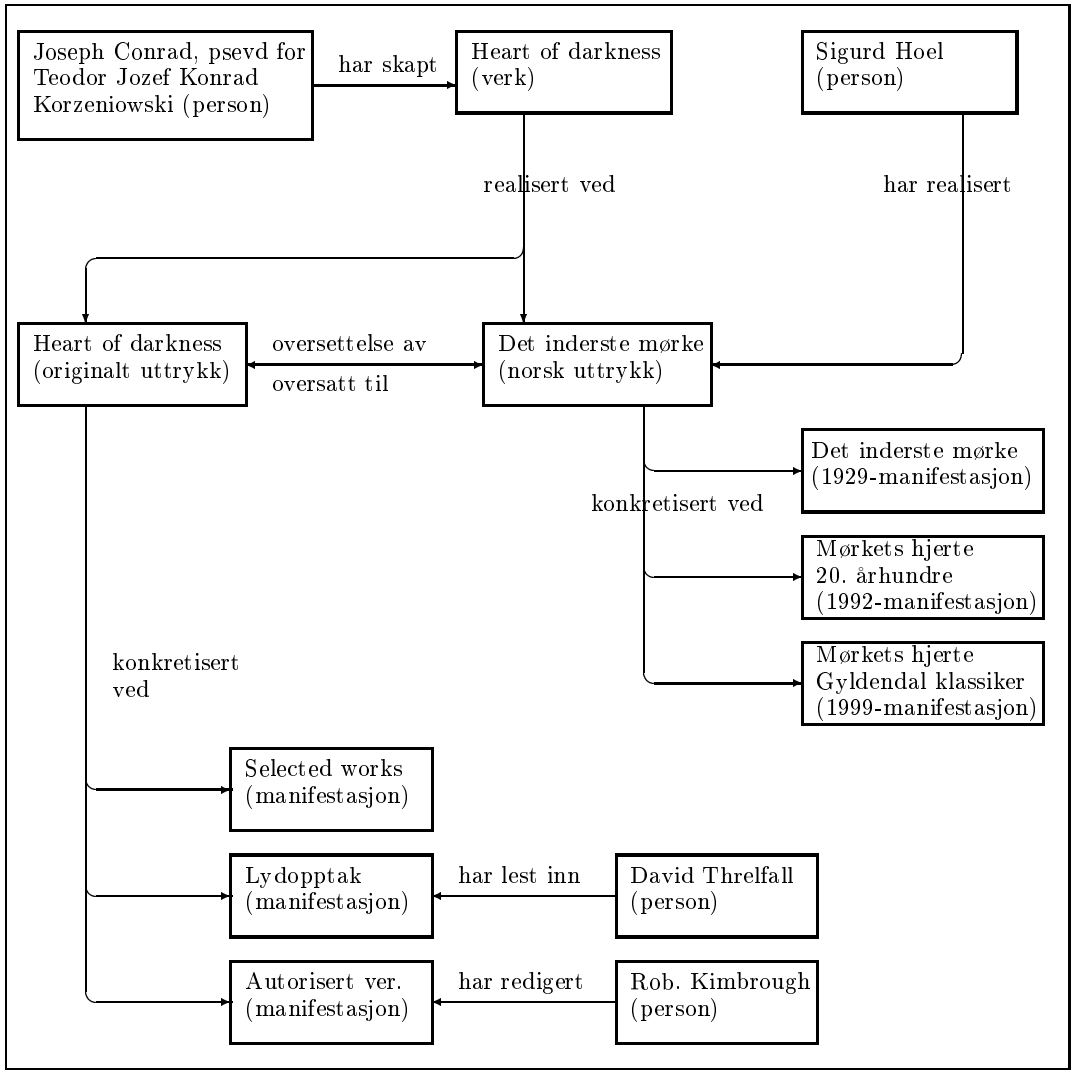
Oppslag i katalogen til Deichmanske bibliotek ledet blant annet til postene gjengitt noe forkortet med MARC-koding i figur D.7. Her framtrer *Heart of darkness* i lydbokutgave, som del av en antologi og i en kommentert utgave. Igjen kan det diskuteres om lydbokutgaven er et nytt uttrykk, en ny manifestasjon eller kanskje til og med et nytt verk. Katalogavdelingene har her et nytt diskusjonstema. I denne framstillingen karakteriseres den som en ny manifestasjon (Deichmanske bibliotek er forresten litt usikker på om Conrad er psevdonym for Korzeniow eller Korzeniowski – det riktige er *Teodor Jozef Konrad Korzeniowski*).

Vil vi nå illustrere noe av den viten vi har om Josephs Conrads verk, så kunne dette framstilles som i figur D.8, der vi også har tatt med noen av de personene som er involvert.

Relasjonen *realisert av* må kvalifiseres slik at man kan skille mellom forfatter- og oversetterrollen (bare så det er sagt).

```
*020 $a0-14-086039-8$blyd.
*090 $adb$bAVKass$c82$dCon
*100 0$aConrad, Joseph$cpsevd. for Joseph Conrad Korzeniow
     $d1857-1924$jeng.$322043300
*24510$aHeart of darkness$cJoseph Conrad ; read by
     David Threlfall$hlydopptak
*260 $aLondon$bPenguin$c1994
*300 $a2 lydkassetter (3 t.)$ci eske 19 x 14 x 3 cm
*440 0$aPenguin audiobooks$330661200


*100 0$aConrad, Joseph$cpsevd. for Joseph Conrad Korzeniow
     $d1857-1924$jeng.$322043300
*24510$aSelected works$bHeart of darkness. Lord Jim.
     The secret agent. Under western eyes.
*260 $aLondon$bLeopard$c1994
*300 $a733s.
*740 $aHeart of darkness
*740 $aLord Jim
*740 $aThe secret agent
*740 $aUnder western eyes


*020 $a0-393-95552-4
*24500$aJoseph Conrad: Heart of darkness
     $ban authoritative text backgrounds and sources criticism
     $cedited by Robert Kimbrough
*250 $a3. ed.
*260 $aNew York$bNorton & co.$c1988
*300 $a420 s.$bill., 2 kart.
*440 0$aA Norton critical edition$329931700
*500 $aHar bibliografi
*600 $aConrad, Joseph$cpsevd. for Joseph Conrad Korzeniow
     $d1857-1924$jeng.$tHeart of darkness$322043300
*700 $aKimbrough, Robert$jam.$ered.$329930800
```

Figur D.7: *Heart of darkness* som lydopptak, innlest av David Threlfall; selected works inklusive *Heart of darkness*; kildekritisk utgave redigert av Robert Kimbrough (fra Deichmans katalog).



Figur D.8: Joseph Conrad, hans verk, noen uttrykk og manifestasjoner av verket, samt relasjoner mellom disse.

Manifestasjonen *Selected works* inneholder tre andre verk av samme forfatter. De knyttes opp mot manifestasjonen på lik linje med *Heart of darkness*, men er ikke tatt med på tegningen.

D.1.4 På skjermen

Hvordan det skal se ut på skjermen når man skal ha fram søkeresultatet, vil være litt avhengig av hvordan brukerne har søkt. En egen utredning er på vei fra IFLA (Guidelines for OPAC displays), så det vil kanskje være å foregripe begivenhetene å si noe om dette.

En mulig tilnærming[2] til problemet er å ta utgangspunkt i kortkatalogens måte å vise/ordne ting på, men å utvide den noe. Vi tenker oss visning langs to akser, en horisontal og en vertikal. Vi forsøker å vise begge aksene samtidig ved hjelp av kort som er forskjøvet i forhold til hverandre, slik at toppen på alle kortene syns (horisontal akse) pluss ett kort som framheves ved at hele kortet syns og der deler av den vertikale aksen illustreres for det valgte verket. Ved å klikke på et av kortene bringes det i forgrunnen.

Horisontal- og vertikalaksen kan være bygd opp etter hva slags søk som er foretatt, eller andre ordningsvalg brukeren har gjort. Noen eksempler på horisontal/vertikal-akser kan være verk/uttrykk, uttrykk/manifestasjon, person/verk, emner/verk osv. Ved hjelp av aktive tekster (som kan klikkes på eller aktiviseres på annen måte) på hvert enkelt kort gis tilgang til nye aksepar med andre ordninger. Aktive tekster er markert med farget og kursivert skrift.

Poenget er å få fram flere dimensjoner i den bibliografiske strukturen. For det første etter det nivået som brukeren har søkt på (f.eks. personnavn) og for det andre at strukturene blir vist på en umiddelbart forståelig måte uten at brukeren blir dynget ned av informasjon. Ved å forskyve kortene i forhold til hverandre får man en dybde i skjermen som viser en hovedakse. Samtidig vises ett kort med informasjon langs en akse som gir mer detaljering. Dette gir fordeler i forhold til de 'flate' og sekvensielle presentasjonene som er mest vanlig i dagens bibliotekkataloger. Det forutsettes at brukeren intuitivt forstår at kort langs hovedaksen kan bringes i forgrunnen ved å klikke på dem.

Søk på personnavn

Ved trunkert søk på personnavn kan man få opp en liste fra autoritetsfila for navn (se figur D.9) som har navneautoriteter langs horisontalaksen og verk og egenskaper langs vertikalaksen. På hvert enkeltkort gis lister over verk personen har befatning med (kortet kan eventuelt snus dersom det er mange opplysninger). Det kan tenkes flere aktive punkter på et slikt kort. Et klikk på ordet *Romaner* kan for eksempel gi en to-dimensjonal liste som gjengitt i figur D.10. Ordningskriterium er her alfabetisk på tittel, men andre ordninger kan tenkes, for eksempel kronologisk etter første utgivelse.

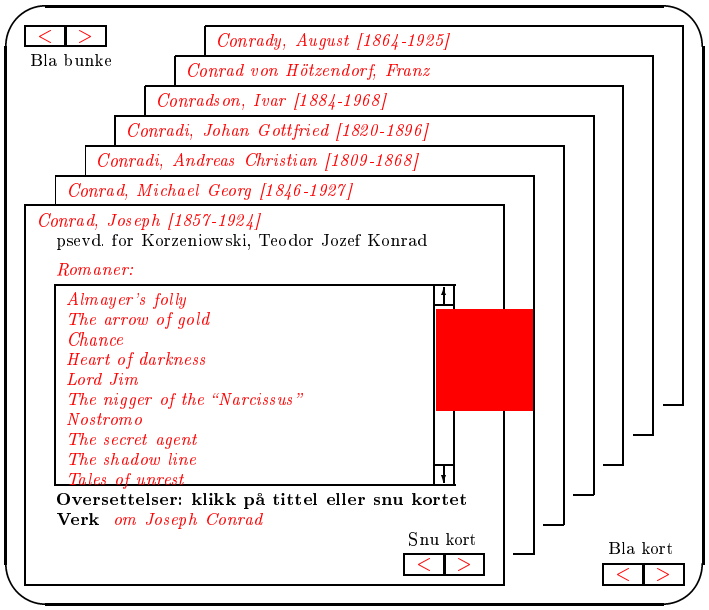
Et klikk på forfatterens navn (i figur D.9) kan gi en horisontalakse ordnet etter type forfatterskap (romaner, noveller, essays, brev) med lister over verk på den vertikale aksen (disse listene kan også bringes fram ved å snu kortet i figur D.9 tilstrekkelig mange ganger - kortet har et ubegrenset antall baksider).

Søk på tittel/ord i tittel

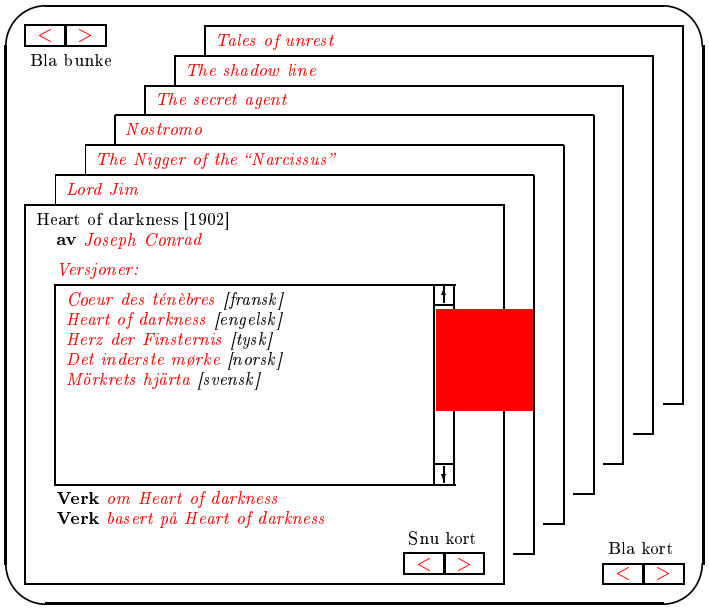
Søk på ord i tittel kan for eksempel gi verktitler langs horisontalaksen og versjoner (uttrykk) langs vertikalaksen (se figur D.11).

Fra verkkort kan man komme til uttrykk/manifestasjon-ordning, der manifestasjonene også har lenker til de enkelte eksemplarer (figur D.12).

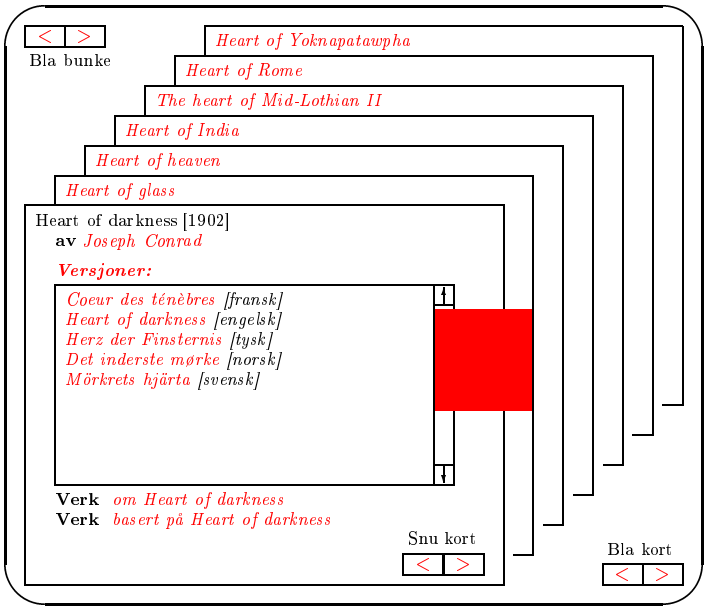
[2] En mer jordnær tilnærming - basert på mulighetene i dagens MARC-poster - er gjort av Anne Munkebyaune [133].



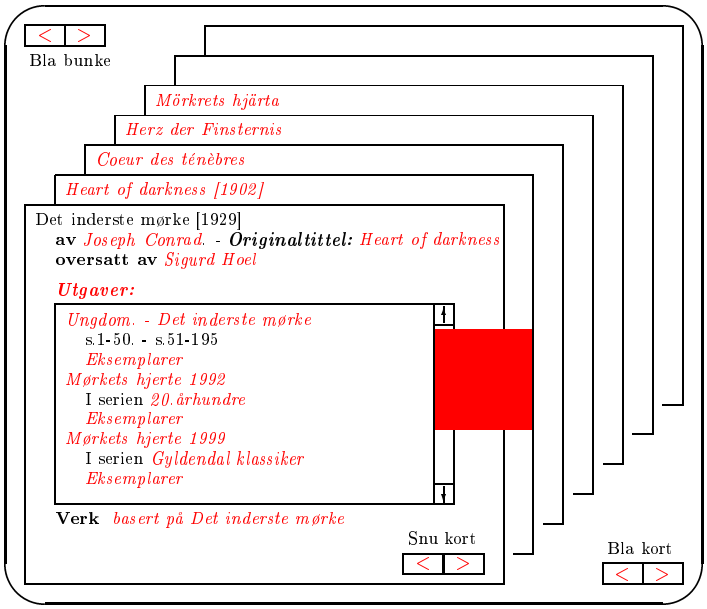
Figur D.9: Trunkert oppslag i autoritetsregister for personnavn (Conrad?,). Horisontalaksen er *forfatter*, vertikalaksen *verk*.



Figur D.10: Verkliste for *Joseph Conrad*. Horisontalaksen er Conrads *verk*, vertikalaksen *versjoner* (uttrykk).



Figur D.11: Søk på *Heart* først i tittel.



Figur D.12: Ulike versjoner (uttrykk) av *Heart of darkness*.

Søk på emne

Søk på emne er et omfattende tema som omfatter direkte og indirekte (via skjemaet) søk på klassifikasjonskoder og kontrollerte emneord. Her kan det tenkes mange variasjoner over temaet aksepar, f.eks når det gjelder hierarkiske relasjoner i klassifikasjonssystemer og oppslagsord med kvalifikatorer fra et kjederegister. Dette får utstå til en seinere anledning.

Bibliografi

[1] Nabil R. Adam og Yelena Yesha. Introduction. *International Journal on Digital Libraries*, 1(1), 1997.

[2] Defense Advanced Research Projects Agency. Transmission Control Protocol. RFC 793, IETF, September 1981.
http://www.ietf.org/rfc/rfc0793.txt [2000-07-09].

[3] Defense Advanced Research Projects Agency. Internet Protocol. RFC 791, IETF, August 1982.
http://www.ietf.org/rfc/rfc0791.txt [2000-07-09].

[4] Paul Albitz og Cricket Liu. *DNS and BIND*. O'Reilly, 3. utgave, 1998.

[5] American National Standards Institute. Information retrieval (Z39.50): Application service definition and protocol specification. American National Standard ANSI/NISO Z39.50-1995, ANSI, 1996.
http://lcweb.loc.gov/z3950/agency/1995doce.html [2000-07-13].

[6] American National Standards Institute. Serial item and contribution identifier (SICI). American National Standard ANSI/NISO Z39.56-1996 (Version 2), ANSI, August 1996.
http://sunsite.berkeley.edu/SICI/ [2000-07-09].

[7] American National Standards Institute. Coded character sets - 7-bit american national standard code for information interchange (7-bit ASCII). American National Standard ANSI X3.4-1986 (R1997), ANSI, 1997. (reaffirmation of ANSI X3.4-1986 (R1992).

[8] Arbeidsgruppen for digitale læremidler. Plan for 1999-2001.
http://www.uio.no/adl/omadl/plan1999-2001.html [2000-07-09].

[9] Arbeidsgruppen for digitale læremidler : Program digitalt bibliotek for digitale læremidler. Notat 1: Konstituering og retningslinjer for programmets arbeid.
http://info.rbt.no/arbeidsgrupper/digital/fagnot6.htm [2000-07-09].

[10] Inc ArborText. Getting started with sgml, 1995.
http://www.oasis-open.org/html/getstart.htm.

[11] William Y. Arms. Key concepts in the architecture of the digital library. *D-Lib Magazine*, July 1995.
http://www.dlib.org/dlib/July95/07arms.html [2000-07-09].

[12] William Y. Arms, Christophe Blanchi og Edward A. Overly. An architecture for information in digital libraries. *D-Lib Magazine*, 3(2), February 1997.
http://www.dlib.org/dlib/february97/cnri/02arms1.html [2000-07-09].

[13] Alan Babich, Jim Davis, Rick Henderson, Dale Lowry, Saveen Reddy og Surendra Reddy. DAV Searching and Locating. Internet draft draft-davis-dasl-protocol-00.html, IETF, April 2000.
http://www.webdav.org/dasl/protocol/draft-davis-dasl-protocol-00.html [2000-07-09].

[14] Michelle Baldano, Chen-Chuan K. Chang, Luis Gravano og Andreas Paepcke. The Stanford digital library metadata architecture. *International journal on digital libraries*, 1:108–121, 1997.

[15] T. Berners-Lee, R. Fielding, U.C. Irvine og L. Masinter. Uniform Resource Identifiers (URI) : Generic syntax. RFC 2396, IETF, August 1998.
http://www.ietf.org/rfc/rfc2396.txt [2000-07-09].

[16] T. Berners-Lee, L. Masinter og M. McCahill. Uniform Resource Locators (URL). RFC 1738, IETF, December 1994.
http://www.ietf.org/rfc/rfc1738.txt [2000-07-09].

[17] Tim Berners-Lee. Universal Resource Identifiers in WWW : A unifying syntax for the expression of names and addresses of objects on the network as used in the World-Wide Web. RFC 1630, IETF, August 1994.
http://www.ietf.org/rfc/rfc1630.txt [2000-07-09].

[18] Grady Booch, Ivar Jacobsen og James Rumbaugh. *The Unified Modeling Language : User Guide.* Object Technology Series. Addison-Wesley, 1999.

[19] Jon Bosak. XML, Java, and the future of the Web, October 1997.
http://metalab.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm [2000-07-10].

[20] C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz og Duane P. Wessels. Harvest: A Scalable, Customizable Discovery and Access System. Rapport CU-CS-732-94, University of Colorado, Boulder, August 1994.

[21] Anne Brüggemann-Klein, Rolf Klein og Britta Landgraf. BibRelEx : Exploring bibliographic databases by visualization of annotated content-based relations. *D-Lib Magazine*, November 1999.
http://www.dlib.org/dlib/november99/landgraf/11landgraf.html [2000-07-10].

[22] BSI. 1989 code for bibliographic identification (biblid) of contributions in serials and books. BSI Standard BS 7187, BSI, 1989. WITHDRAWN.

[23] Michael Buckland. *Information and Information Systems.* Praeger, 1991.

[24] Michael Buckland. What is a "document"? *Journal of the American Society for Information Science*, 48(9):804–809, September 1997.

[25] Michael Buckland. What is a "digital document"? *Document Numerique*, 2(2):221–230, 1998.
http://info.berkeley.edu/~buckland/digdoc.html [2000-07-09].

[26] Peter Buneman og Stan Zdonik. Editorial. *International Journal on Digital Libraries*, 1(1), 1997.

[27] Vannevar Bush. As We May Think. *Atlantic Monthly*, July 1945.
http://www.theatlantic.com/unbound/flashbks/computer/bushf.htm.

[28] Robert Cailliau. A Little History of the World Wide Web from 1945 to 1995, 1995.
http://www.w3.org/History.html [2000-07-09].

[29] Priscilla Caplan. You call it corn, we call it syntax-independent metadata for document-like objects. *The Public-Access Computer Systems Review*, 6(4), 1995.
http://info.lib.uh.edu/pr/v6/n4/capl6n4.html [2000-07-09].

[30] R. G. G. Cattell og Douglas K. Barry, redaktører. *The Object Data Standard : ODMG 3.0*. Morgan Kaufmann Publishers, 2000.

[31] Peter Pin-Shan Chen. The entity relationship model - towards a unified view of data. *ACM Transactions on Database Systems*, 1(1), March 1976.

[32] CIMI. CIMI projects.
http://www.cimi.org/projects/index.html [2000-07-09].

[33] CNRI. Grail home page.
http://grail.python.org/ [2000-07-09].

[34] Metadata Ad Hoc Working Group : Federal Geographic Data Committee. Content standard for digital geospatial metadata (version 2.0). Standard FGDC-STD-001-1998, FGDC, 1998.
http://www.fgdc.gov/metadata/csdgm/ [2000-07-09].

[35] Dan Conolly. Naming and adressing : URI's, URL's ..., 1999.
http://www.w3c.org/Addressing [2000-07-09].

[36] George Coulouris, Jean Dollimore og Tim Kindberg. *Distributed Systems : Concepts and Design*. Addison-Wesley, 1994.

[37] Charles A. Cutter. *Rules for a dictionary catalogue : special report on public libraries*. Government printing office, 1904.

[38] L. Daigle, D. van Gulik og R. Ianella. URN namespace definition mechanisms. RFC 2611, IETF, June 1999.
http://www.ietf.org/rfc/rfc2611.txt [2000-07-09].

[39] R. Daniel. Resolution of Uniform Resource Identifiers using the Domain Name System. RFC 2168, IETF, June 1997.
http://www.ietf.org/rfc/rfc2168.txt [2000-07-09].

[40] Jim Davis, David Fielding, Carl Lagoze og Richard Marisa. Dienst : Overview and introduction, Mars 2000.
http://www.cs.cornell.edu/cdlrg/dienst/DienstOverview.htm [2000-07-09].

[41] Jim Davis, David Fielding, Carl Lagoze og Richard Marisa. Dienst protocol specification, Mai 2000.
http://www.cs.cornell.edu/cdlrg/dienst/protocols/DienstProtocol.htm
[2000-07-09].

[42] Jim Davis, David Fielding, Carl Lagoze og Richard Marisa. Dienst software : Summary description, April 2000.
http://www.cs.cornell.edu/cdlrg/dienst/software/DienstSoftware.htm
[2000-07-09].

[43] Michael Day. ROADS cataloguing guidelines, June 1999.
http://www.ukoln.ac.uk/metadata/roads/cataloguing/cataloguing-rules.html [2000-07-09].

[44] DC Relation/Source Working Group. Review of relation qualifier usage, August 1999-08-04.
http://purl.org/DC/groups/relation-qualifierreview.htm [2000-07-09].

[45] Lorcan Dempsey og Rachel Heery. A review of metadata: a survey of current resource description formats, 1997.
http://www.ukoln.ac.uk/metadata/desire/overview/ [2000-07-09].

[46] Desmond Francis D'Souza og Alan Cameron Wills. *Objects, Components, and Frameworks with UML.* Object Technology Series. Addison-Wesley, 1999.

[47] The Dublin Core Element Set, version 1.0: Reference description. The Dublin Core Metadata Inititative, March 1995.
http://purl.org/DC/documents/rec-dces-199809.htm [2000-07-09].

[48] The Dublin Core Element Set, version 1.1: Reference description. The Dublin Core Metadata Inititative, July 1997.
http://purl.org/DC/documents/rec-dces-19990702.htm [2000-07-09].

[49] The Dublin Core Metadata Initiative. Dublin Core Qualifiers, juli 2000.
http://purl.org/dc/documents/rec/dcmi-qualifiers-20000711.htm [2000-07-09].

[50] ECMA. Ecmascript language spesification. Standard ECMA 262, ECMA, December 1999.
ftp://ftp.ecma.ch/ecma-st/Ecma-262.pdf [2000-07-09].

[51] Electronic Text Center at the University of Virginia . TEI guidelines for electronic text encoding and interchange (p3), udatert.
http://etext.lib.virginia.edu/TEI.html [2000-07-09].

[52] Ramez Elmasri og Shamkant B. Navathe. *Fundamentals of database systems.* Benjamin/Cummings publishing Company, 1994.

[53] EU-NSF Digital Library Working Group on Interoperability between Digital Libraries. EU-NSF Digital Library Working Group on Interoperability between Digital Libraries : Position paper.
http://galileo.iei.pi.cnr.it/DELOS/NSF/interop.htm [2000-07-09].

[54] Bernhard Eversberg. To link or not to link, and how. Innlegg på epostdiskusjonsliste før Toronto-konferansen om revisjon av AACR2, August 1997. Hans innlegg kan finnes i arkivet for epostlista på følgende datoer: Mon, 4 Aug 1997 12:05:44; Thu, 14 Aug 1997 10:20:17; Thu, 21 Aug 1997 13:56:24; Fri, 29 Aug 1997 14:50:52.
http://www.nlc-bnc.ca/jsc/aacrconf.log9708 [2000-07-09].

[55] Bernhard Eversberg. REUSE+ : The part/whole relationship in German and American cataloging data. Results and suggestions, June 1998.
http://www.biblio.tu-bs.de/allegro/formate/reusep.htm [2000-07-09].

[56] R. Fielding, J. Gettys, J. C. Mogul, H. Frystyk, L. Masinter, P. Leach og T. Berners-Lee. Hypertext Transfer Protocol – HTTP/1.1. RFC 2616, IETF, June 1999.
http://www.ietf.org/rfc/rfc2616.txt [2000-07-09].

[57] David Flanagan. *JavaScript : The Definitive Guide*. O'Reilly, 2nd utgave, 1997.

[58] N. Freed og N. Borenstein. Multipurpose Internet Mail Extensions : (MIME) Part One : Format of Internet Message Bodies. RFC 2045, IETF, November 1996.
http://www.ietf.org/rfc/rfc2045.txt [2000-07-09].

[59] N. Freed og N. Borenstein. Multipurpose Internet Mail Extensions : (MIME) Part Two : Media Types. RFC 2046, IETF, November 1996.
http://www.ietf.org/rfc/rfc2046.txt [2000-07-09].

[60] N. Freed, J. Klensin og J. Postel. Multipurpose Internet Mail Extensions : (MIME) Part Four : Registration Procedures. RFC 2048, IETF, November 1996.
http://www.ietf.org/rfc/rfc2048.txt [2000-07-09].

[61] Getty Information Institute and the College Art Association. Categories for the Description of Works of Art.
http://www.getty.edu/gri/standard/cdwa/ [2000-07-10].

[62] Y. Goland, E.J. Whitehead, A. Faizi, S.R. Carter og D. Jenson. HTTP extensions for distributed authoring – WebDAV. RFC 2518, IETF, February 1999.
http://www.ietf.org/rfc/rfc2518.txt [2000-07-09].

[63] Danny Goodman. *Dynamic HTML : The Definitive Reference*. O'Reilly, 1998.

[64] Michael Gorman. The future of cataloguing and cataloguers. I *Booklet 4*, side 35–41. IFLA, August 1997.

[65] Government Information Locator Service. Government Information Locator Service (GILS), August 1999.
http://www.access.gpo.gov/su_docs/gils/index.html [2000-07-10].

[66] Luis Gravano, Chen-Chuan K. Chang, Hector Garcia-Molina og Andreas Paepcke. STARTS: Stanford Protocol Proposal for Internet Retrieval and Search. Rapport SIDL-WP-1996-0043, Stanford, 1996.
http://www-diglib.stanford.edu/cgi-bin/get/SIDL-WP-1996-0043 [2000-07-09].

[67] Kaj Grønbæk og Randall H. Trigg. Toward a Dexter-based model for open hypermedia unifying embedded references and link objects . I *Proceedings of the the seventh ACM conference on HYPERTEXT '96*, side 149–160, 1996.

[68] Juha Hakala. Using national bibliography numbers as uniform resource names. Internet draft draft-hakala-nbn-00, IETF, February 2000.
http://search.ietf.org/internet-drafts/draft-hakala-nbn-00.txt [2000-07-09].

[69] Frank Halasz og Mayer Schwartz. The Dexter Hypertext Reference Model. I *Proceedings of the Hypertext Standardization Workshop*, nr 500-178 i Nist Special Publications, side 95–133. National, Institute of Standards and Technology, U.S. Department of Commerce, January 1990.

[70] Frank Halasz og Mayer Schwartz. The Dexter Hypertext Reference Model. *Communications of the ACM*, 37(2):30–39, February 1994.

[71] Gisle Hannemyr. *Åpne systemer : teknologi, strategi og praksis*. Universitetsforlaget, 1992.

[72] Lynda Hardman, Dick C.A. Bulterman og Guido Van Rossum. The Amsterdam Hypermedia Model : Adding Time and Context to the Dexter Model. *Communications of the ACM*, 37(2):50–62, February 1994.

[73] Darren R. Hardy, Michael F. Schwartz og Duane Wessels. Harvest user's manual. Rapport CU-CS-743-94, University of Colorado, Boulder, Januar 1994.

[74] Rachel Heery. Review of metadata formats. *Program*, 30(4):345–373, October 1996.

[75] Steve Hitchcock, Les Carr, Wendy Hall, Stephen Harris, S.Probets, D.Evans og D.Brailsford. Linking electronic journals : Lessons from the open journal project. *D-Lib Magazine*, December 1998.
http://www.dlib.org/dlib/december98/12hitchcock.html [2000-07-10].

[76] Brian Phillip Holt. Project UseMARCON, 1998.
http://libis.lt/events/ifla/holt1.html [2000-07-13].

[77] IANA. Protocol Numbers and Assignment Services : Media Types Directory.
http://www.isi.edu/in-notes/iana/assignments/media-types/media-types
[2000-07-09].

[78] IANA. Protocol Numbers and Assignment Services : Uniform Resource Locator (URL) Schemes.
http://www.isi.edu/in-notes/iana/assignments/url-schemes [2000-07-09].

[79] IANA. Protocol Numbers and Assignment Services : Uniform Resource Names (URN) Namespaces.
http://www.isi.edu/in-notes/iana/assignments/urn-namespaces [2000-07-09].

[80] IANA. Protocol Numbers and Assignment Services : Uniform Resources Identifiers (URI) Namespaces.
http://www.isi.edu/in-notes/iana/assignments/uri-namespaces [2000-07-09].

[81] IFLA. Universal Bibliographic Control and International MARC Core Programme. UNIMARC: An Introduction, 1999.
http://www.ifla.org/VI/3/p1996-1/unimarc.htm [2000-07-09].

[82] IFLA study group on the functional requirements for bibliographic records. Functional requirements for bibliographic records : final report. Rapport, IFLA, 1998.
http://www.ifla.org/VII/s13/frbr/frbr.pdf [2000-07-09].

[83] IFPI. International Standard Recording Code (ISRC).
http://www.ifpi.org/online/isrc_intro.html [2000-07-09].

[84] International Conference on Cataloguing Principles . *Statement of principles: adopted at the International Conference on Cataloguing Principles.* International Federation of Library Associations (Committee on Cataloguing), 1961.

[85] The International Organization For Standardization. Introduction to ISO.
http://www.iso.ch/infoe/intro.htm [2000-07-09].

[86] The International Organization For Standardization. International Standard Recording Code (ISRC). International Standard ISO 3901, ISO, 1986.

[87] The International Organization For Standardization. Standard Generalized Markup Language (SGML). International Standard ISO 8879:1986, ISO, 1986.

[88] The International Organization For Standardization. 7-bit coded character set for information interchange. International Standard ISO/IEC 646:1991, ISO, 1991.

[89] The International Organization For Standardization. International Standard Book Number (ISBN). International Standard ISO 2108, ISO, 1992.

[90] The International Organization For Standardization. SQL. International Standard ISO/IEC 9075:1992, ISO, 1992.

[91] The International Organization For Standardization. International Standard Music Number (ISMN). International Standard ISO 10957, ISO, 1993.

[92] The International Organization For Standardization. Basic Reference Model : The Basic Model . International Standard ISO/IEC 7498-1:1994, ISO, 1994.

[93] The International Organization For Standardization. International Standard Technical Report Number (ISRN). International Standard ISO 10444, ISO, 1994.

[94] The International Organization For Standardization. X.25 Packet Layer Protocol for Data Terminal Equipment. International Standard ISO/IEC 8208:1995, ISO, 1995.

[95] The International Organization For Standardization. Document Style Semantics and Specification Language (DSSSL). International Standard ISO/IEC 10179, ISO, 1996.

[96] The International Organization For Standardization. Hypermedia/Time-based Structuring Language (HyTime). International Standard ISO/IEC 10744, ISO, 1997.

[97] The International Organization For Standardization. Interlibrary Loan Application Protocol Specification . International Standard ISO 10161:1997, ISO, 1997.

[98] The International Organization For Standardization. Interlibrary Loan Application Service Definition. International Standard ISO 10160:1997, ISO, 1997.

[99] The International Organization For Standardization. 8-bit single-byte coded graphic character sets – part 1: Latin alphabet no. 1. International Standard ISO/IEC 8859-1:1998, ISO, 1998.

[100] The International Organization For Standardization. Information retrieval (Z39.50) : Application service definition and protocol specification. International Standard ISO 23950:1998, ISO, 1998.

[101] The International Organization For Standardization. International Standard Serial Number (ISSN). International Standard ISO 3297, ISO, 1998.

[102] The International Organization For Standardization. Call-Level Interface (SQL/CLI). International Standard ISO/IEC 9075-3:1999, ISO, 1999.

[103] The International Organization For Standardization. Document Description and Processing Languages : HyperText Markup Language (HTML). Draft International Standard ISO/IEC 15445:1999(E), ISO, 1999.

[104] The International Organization For Standardization. Universal Multiple-Octet Coded Character Set (UCS) : Part 1 : Architecure and Basic Multilingual Plane. International Standard ISO/IEC 10646-1:2000, ISO, 2000.

[105] ISMN-kontoret i Norge. ISMN – Internasjonalt Standard MusikkNummer. http://www.nb.no/html/veil21.html [2000-07-09].

[106] ISSN Norge. ISSN-internasjonalt standardnummer for periodika. http://www.nb.no/html/veil14.html [2000-07-09].

[107] Bill Janssen, Henrik Frystyk Nielsen og Mike Spreitzer. HTTP-ng Architectural Model. Internet Draft draft-frystyk-httpng-arch-00.txt, W3C, August 1998. http://www.w3.org/Protocols/HTTP-NG/1998/08/draft-frystyk-httpng-arch-00.txt [2000-07-09].

[108] Robert Kahn og Robert Wilensky. A framework for distributed digital object services, May 1995. http://www.CNRI.Reston.VA.US/home/cstr/arch/k-w.html [2000-07-09].

[109] Koninklijke Bibliotheek - National Library of the Netherlands. User controlled generic MARC converter. http://www.konbib.nl/kb/resources/frameset_kb.html?/kb/sbo/bibinfra/usema-en.html [2000-07-13].

[110] D. Kristol og L. Montulli. HTTP state management mechanism. RFC 2109, IETF, February 1997. http://www.ietf.org/rfc/rfc2109.txt [2000-07-09].

[111] John A. Kunze og R. P. C. Rodgers. Z39.50 in a nutshell : An introduction to Z39.50, 1995. http://www.informatik.tu-darmstadt.de/VS/Infos/Protocol/Z39.50/z39.50-nutshell.html [2000-07-10].

[112] Carl Lagoze og Sandra Payette. An infrastructure for open architecture digital libraries. Cornell Computer Science Technical Report TR98-1690, Cornell University, Department of Computer Science, June 1998. http://ncstrl.cs.cornell.edu:80/Dienst/UI/1.0/Display/ncstrl.cornell/TR98-1690 [2000-07-09].

[113] Carl Lagoze, Erin Shaw, James R. Davis og Dean B. Krafft. Dienst: Implementation reference manual. Cornell Computer Science Technical Report TR95-1514, Computer Science Department of Cornell University, 1995. http://cs-tr.cs.cornell.edu:80/Dienst/UI/1.0/Display/ncstrl.cornell/TR95-1514 [2000-07-17].

[114] Brian Lavoie og Henrik Frystyk Nielsen. Web characterization terminology and definitions sheet. W3C Working Draft 24-May-1999, W3C, May 1999. http://www.w3.org/1999/05/WCA-terms/ [2000-03-15].

[115] Tim Berners Lee. WorldWideWeb. http://www.w3.org/People/Berners-Lee/WorldWideWeb.html [2000-07-09].

[116] Tim Berners Lee. Information Management: A Proposal , Mars 1989. http://www.w3.org/History/1989/proposal.html [2000-07-09].

[117] Library of Congress. MARC standards, 2000. http://lcweb.loc.gov/marc/ [2000-07-09].

[118] Library of Congress. Network development and MARC standards office. MARC DTD : background and development, 1998.
http://www.loc.gov/marc/marcdtd/marcdtdback.html [2000-07-09].

[119] Håkon Wium Lie og Bert Bos. Cascading Style Sheets, level 1. W3C Recommendation REC-CSS1-19990111, W3C, January 1999.
http://www.w3.org/TR/REC-CSS1 [2000-07-09].

[120] C. Lynch, C. Preston og R. Daniel. Using existing bibliographic identifiers as Uniform Resource Names. RFC 2288, IETF, February 1998.
http://www.ietf.org/rfc/rfc2288.txt [2000-07-09].

[121] Clifford Lynch. Searching the internet. *Scientific American*, Mars 1997.
http://www.sciam.com/0397issue/0397lynch.html.

[122] David Martin. A standard identifier for book items and contributions. Final draft, Book Industry Communication and the British National Bibliography Fund, September 1999.
http://www.bic.org.uk/bici.html [2000-07-09].

[123] James Martin. *Strategic data-planning methodologies*. Prentice-Hall, 1982.

[124] L. Masinter. Hyper Text Coffee Pot Control Protocol (HTCPCP/1.0). RFC 2324, IETF, April 1998.
http://www.ietf.org/rfc/rfc2324.txt [2000-07-09].

[125] Microsoft. Dcom.
http://www.microsoft.com/com/tech/DCOM.asp [2000-07-19].

[126] Francisco Millarch. The digital information : an analysis of information overload, and document preservation in cyberspace. Rapport, University of Westminister, September 1998.
http://www.millarch.org/francisco/papers/Dissertation.htm [2000-07-09].

[127] Paul Miller. Z39.50 for all. *Ariadne*, 21, September 1999.
http://www.ariadne.ac.uk/issue21/z3950/.

[128] Jessica Milstead og Susan Feldman. Metadata : cataloging by any other name. *Online*, January 1999.
http://www.onlineinc.com/onlinemag/metadata/ [2000-07-09].

[129] R. Moats. URN syntax. RFC 2141, IETF, May 1997.
http://www.ietf.org/rfc/rfc2141.txt [2000-07-09].

[130] P. Mockapetris. Domain Names : Concepts and Facilities. RFC 1034, IETF, November 1987.
http://www.ietf.org/rfc/rfc1034.txt [2000-07-09].

[131] P. Mockapetris. Domain Names : Implementation and specification. RFC 1035, IETF, November 1987.
http://www.ietf.org/rfc/rfc1035.txt [2000-07-09].

[132] William Moen. The ANSI/NISO Z39.50 protocol: Information retrieval in the information infrastructure.
http://www.cni.org/pub/NISO/docs/Z39.50-brochure/.

[133] Anne Munkebyaune. Bibliografiske data og gjenfinningssystemer : en undersøkelse av hvordan noen norske bibliografiske databaser har utnyttet de bibliografiske dataenes struktur. Hovedfagsoppgave, Høgskolen i Oslo, 1999. http://www.nb.no/~anne/AnnesDiplom.pdf [2000-07-09].

[134] Robin Murray. The digital library jigsaw: fitting the pieces together. I *Online information 99 : 23rd International Online Information Meeting*. Learned Information Europe Ltd, December 1999.

[135] S. Nelson og C. Parks. The model primary content type for multipurpose internet mail extensions. RFC 2077, IETF, January 1997. http://www.ietf.org/rfc/rfc2077.txt [2000-07-09].

[136] Ted Nelson. A file structure for the complex, the changing and the indeterminate. I *proceedings of the ACM 20th national conference*. ACM, 1965.

[137] G. Nicol, G. Adams og M. Duerst. Internationalization of the hypertext markup language. RFC 2070, IETF, January 1997. http://www.ietf.org/rfc/rfc2070.txt [2000-07-09].

[138] NISO. Syntax for the Digital Object Identifier. Draft standard Z39.84.XXXX, NISO, July 1999. http://www.niso.org/Z3984.html [2000-07-09].

[139] NKKM og Østbye, Jon Birger. Feltkatalog for NKKMs EDB-prosjekter. Rapport, Humanistisk datasenter, Universitetet i Bergen, 1992. http://www.hd.uib.no/regimus/feltkode.html [2000-07-09].

[140] Nordic Metadata Project. Nordic countries URN-generator. http://www.lub.lu.se/cgi-bin/nmurn.pl [2000-07-09].

[141] OMG. The common object request broker : Architecture and specification, July 1995. ftp://ftp.omg.org/pub/docs/formal/99-10-08.pdf [2000-07-09].

[142] OMG. OMG Unified Modeling Language : Specification, March 2000. ftp://ftp.omg.org/pub/docs/formal/00-03-01.pdf [2000-07-10].

[143] Robert Orfali, Dan Harkey og Jeri Edwards. *The Essential Distributed Objects Survival Guide*. Wiley, 1996.

[144] Andreas Paepcke, Steve B. Cousins, Hector Garcia-Molina, Scott W. Hassan, Steven P. Ketchpel, Martin Röscheisen og Terry Winograd. Using distributed objects for digital library interoperability. *IEEE Computer*, May 1996. URL http://www.computer.org/computer/dli/r50061/r50061.htm [2000-07-09].

[145] Norman Paskin. DOI : Current status and outlook. *D-Lib Magazine*, 5(5), May 1999. http://www.dlib.org/dlib/may99/05paskin.html [2000-07-09].

[146] Norman Paskin. Toward unique identifiers. *Proceedings of the IEEE*, 87(7):1208–1227, July 1999. http://teaser.ieee.org/pubs/mags/9908/87proc07-paskin.html [2000-07-09].

[147] Norman Paskin og Godfrey Rust. The Digital Object Identifier initiative: metadata implications. DOI discussion paper, The DOI initiative, February 1999. http://www.doi.org/P2VER3.PDF [2000-07-09].

[148] J. Postel. User Datagram Protocol. RFC 768, IETF, August 1980. http://www.ietf.org/rfc/rfc0768.txt [2000-07-09].

[149] J. Postel og J. Reynolds. Telnet Protocol Spesification. RFC 854, IETF, May 1983. http://www.ietf.org/rfc/rfc0854.txt [2000-07-09].

[150] J. Postel og J. Reynolds. File Transfer Protocol. RFC 959, IETF, October 1985. http://www.ietf.org/rfc/rfc0959.txt [2000-07-09].

[151] Jonathan B. Postel. Simple Mail Transfer Protocol. RFC 821, IETF, August 1982. http://www.ietf.org/rfc/rfc0821.txt [2000-07-09].

[152] Martin Roscheisen, Michelle Baldonado, Kevin Chang, Luis Gravano, Steven Ketchpel og Andreas Paepcke. The Stanford InfoBus and Its Service Layers: Augmenting the Internet with Higher-Level Information Management Protocols. Rapport SIDL-WP-1997-0065, Stanford, 1997. http://www-diglib.stanford.edu/cgi-bin/get/SIDL-WP-1997-0065 [2000-07-09] Finnes også i Digital libraries in computer science: the MeDoc approach, s.213-30, Springer-Verlag, 1998.

[153] Bill Rosenblatt. The Digital Object Identifier : Solving the dilemma of copyright protection online. *The Journal of Electronic Publishing*, 3(2), December 1997. http://www.press.umich.edu/jep/03-02/doi.html [2000-07-09].

[154] Keith Shafer, Stuart Weibel, Erik Jul og Jon Fausey. Introduction to Persistent Uniform Resource Locators. http://purl.oclc.org/OCLC/PURL/INET96 [2000-07-09].

[155] Errol Simon. *Distributed Information Systems : From Client/Server to Distributed Multimedia*. McGraw-Hill Publishing Company, 1996.

[156] Arne Sølvberg. Introduction to concept modeling for information systems, 2000. Del av kompendium i faget DIF 8915 Semantisk datamodellering, våren 2000, IDI, NTNU.

[157] K. Sollins. Architectural principles of Uniform Resource Name resolution. RFC 2276, IETF, January 1998. http://www.ietf.org/rfc/rfc2276.txt [2000-07-09].

[158] K. Sollins og L. Masinter. Functional requirements for Uniform Resource Names. RFC 1737, IETF, December 1994. http://www.ietf.org/rfc/rfc1737.txt [2000-07-09].

[159] Herbert Van de Sompel og Patrick Hochstenbach. Reference linking in a hybrid library environment : Part 2: SFX, a generic linking solution. *D-Lib Magazine*, April 1999. http://www.dlib.org/dlib/april99/van_de_sompel/04van_de_sompel-pt2.html [2000-07-10].

[160] Inger Cathrine Spangen. *Katalogiseringsregler : Anglo-American cataloguing rules, second edition / oversatt og bearbeidet for norske forhold ved Inger Cathrine Spangen.* Norsk bibliotekforening, 1998. ISBN 82-90790-14-7.

[161] Sam X. Sun og Larry Lannom. Handle system overview. Internet-draft draft-sun-handle-system-04.txt, IETF, Feb. 2000.
http://www.ietf.org/internet-drafts/draft-sun-handle-system-04.txt [2000-07-03]
http://www.handle.net/overview-current.html [2000-07-09].

[162] Sam X. Sun, Sean Reilly og Larry Lannom. Handle system namespace and service definition. Internet-draft draft-sun-handle-system-def-02.txt, IETF, Feb. 2000.
http://www.ietf.org/internet-drafts/draft-sun-handle-system-def-02.txt
http://www.handle.net/namespace-current.html [2000-07-09].

[163] The International Organization For Standardization TC 46/SC 9. International Standard Audiovisual Number (ISAN) : Frequently Asked Questions.
http://www.nlc-bnc.ca/iso/tc46sc9/isan.htm [2000-07-09].

[164] The International Organization For Standardization TC 46/SC 9. International Standard Musical Work Code (ISWC).
http://www.nlc-bnc.ca/iso/tc46sc9/iswc.htm.

[165] The Nordic Metadata Project. d2m : Dublin Core to MARC converter, 1998.
http://www.bibsys.no/meta/d2m/ [2000-07-13].

[166] Barbara B. Tillett. A taxonomy of bibliographic relationships. *Library Resources & Technical Services,*, 35(2):150–158, April 1991.

[167] The Unicode Consortium. *The Unicode Standard, Version 3.0.* Addison Wesley, 2000.

[168] Visual Resources Association Data Standards Committe. VRA core categories, version 3.0, June 2000.
http://www.gsd.harvard.edu/~staffaw3/vra/vracore3.htm [2000-10-06].

[169] Visual Resources Association.
http://www.oberlin.edu/~art/vra/vra.html [2000-07-10.

[170] W3C. An Index of WWW Addressing Schemes.
http://www.w3.org/Addressing/schemes.html [2000-07-09].

[171] W3C. Uniform Resource Names (URN) .
http://www.ietf.org/html.charters/urn-charter.html [2000-07-09].

[172] W3C. XML Query.
http://www.w3.org/XML/Query.html [2000-07-10].

[173] W3C. XML Schema.
http://www.w3.org/XML/Schema.html [2000-07-10].

[174] W3C. Extensible Markup Language (XML) 1.0. W3C Recommendation REC-xml-19980210, W3C, February 1998.
http://www.w3.org/TR/1998/REC-xml-19980210 [2000-07-10].

[175] W3C. Synchronized Multimedia Integration Language (SMIL) 1.0 Specification . W3C Recommendation REC-smil-19980615, W3C, June 1998.
http://www.w3.org/TR/1998/REC-smil-19980615 [2000-07-09].

[176] W3C. HTML 4.01 Specification. W3C Recommendation REC-html401-19991224, W3C, December 1999.
http://www.w3.org/TR/1999/REC-html401-19991224/ [2000-07-09].

[177] W3C. Mathematical Markup Language (MathML) 1.01 Specification. W3C Recommendation REC-MathML-19990707, W3C, January 1999.
http://www.w3.org/1999/07/REC-MathML-19990707 [2000-07-09].

[178] W3C. Namespaces in XML. W3C Recommendation REC-xml-names-19990114, W3C, January 1999.
http://www.w3.org/TR/1999/REC-xml-names-19990114 [2000-07-09].

[179] W3C. Resource Description Framework (RDF) : Model and Syntax Specification. W3C Recommendation REC-rdf-syntax-19990222, W3C, February 1999.
http://www.w3.org/TR/1999/REC-rdf-syntax-19990222 [2000-07-15].

[180] W3C. XML Path Language (XPath) Version 1.0. W3C Recommendation REC-xpath-19991116, W3C, November 1999.
http://www.w3.org/TR/1999/REC-xslt-19991116 [2000-07-10].

[181] W3C. XSL Transformations (XSLT) Version 1.0. W3C Recommendation REC-xslt-19991116, W3C, November 1999.
http://www.w3.org/TR/1999/REC-xslt-19991116 [2000-07-09].

[182] W3C. Extensible Stylesheet Language (XSL) Version 1.0. W3C Working Draft WD-xsl-20000327, W3C, 2000.
http://www.w3.org/TR/2000/WD-xsl-20000327/ [2000-07-09].

[183] W3C. Scalable Vector Graphics (SVG) 1.0 Specification. W3C Working Draft WD-SVG-20000303, W3C, 2000.
http://www.w3.org/TR/2000/03/WD-SVG-20000303 [2000-07-09].

[184] W3C. XHTML 1.0: The Extensible HyperText Markup Language : A Reformulation of HTML 4 in XML 1.0. W3C Recommendation REC-xhtml1-20000126, W3C, January 2000.
http://www.w3.org/TR/2000/REC-xhtml1-20000126 [2000-07-09].

[185] W3C. XML Linking Language (XLink) Version 1.0. W3C Candidate Recommendation CR-xlink-20000703, W3C, July 2000.
http://www.w3.org/TR/2000/CR-xlink-20000703/ [2000-07-10].

[186] W3C. XML Pointer Language (XPointer) Version 1.0. W3C Candidate Recommendation CR-xptr-20000607, W3C, June 2000.
http://www.w3.org/TR/2000/CR-xptr-20000607 [2000-07-10].

[187] Mark G. Wales. WIDL : Interface Definition for the Web. *Internet Computing*, side 55–59, January/February 1999.

[188] Martha M. Yee. Guidelines for OPAC displays, November 1998.
http://www.ifla.org/VII/s13/guide/opac-d.pdf [1999-12-07].

[189] W. Yeong, T. Howes og S. Kille. Lightweight Directory Access Protocol. RFC 1777, IETF, March 1995.
http://www.ietf.org/rfc/rfc1777.txt [2000-07-09].