

Presentasjon av avanserte lenkestrukturer

Xlink i Webbaserte dokumente

Tor Åge Ballo

Master i informatikk
Oppgaven levert: Juni 2006
Hovedveileder: Trond Aalberg, IDI

Sammendrag

Avanserte lenkestrukturer tilbyr lenkemuligheter som er mer kompleks enn ved lenking i HTML. Lenkemodellen XLink introduserer funksjonalitet som blant annet muliggjør beskrivelse av relasjoner mellom flere enn to ressurser, rik metadata til lenkene, og tredjeparts lenker. Avhandlingen vil ta for seg ulike løsninger for å implementere avanserte lenkestrukturer, slik at de støttes av dagens nettlesere. Det blir sett på hvilke funksjonalitet fra XLink det er mulig å integrere på Web med eksisterende teknologi og hvordan slike lenkener bør utformes i grensesnitt. Det vil bli testet gjennom en applikasjon som har til hensikt å utforsker mulighetene ved bruk av XLink på Web. Applikasjonen vil bli implementert i en prototyp av BIBSYS som benytter FRBR modellen.

Forord

Denne avhandlingen er skrevet som en del av studieprogrammet informatikk master ved Norges teknisk-naturvitenskapelige universitet i Trondheim. Oppgaven er det siste leddet i studieprogrammet og markerer en avslutningen på masterstudiet.

Jeg ønsker å takke min veileder Trond Aalberg for all hjelp, og alle innspill i forhold til avhandlingen. Hans positive holdning har gjort det til en glede å gå siste året på master ved NTNU. Ønsker også å takke Fabio Arciniegas for e-post med dokumentasjon av XLink2HTML, min medstudent Morten Tvenning for innspill og de kulinariske duftene av gamle oster, hvitløksmarinerte oliven og annen mat som har forpestet arbeidsplassen, Mathis S Mathisen for hjelp til gjennomlesning, og Marit Neerland for støtte og tålmodighet.

Denne masteroppgave er skrevet i tidsperioden august 2005 – juni 2006. Oppgavens tema er avanserte lenkestrukturer og hvordan disse kan presenteres på Web.

God lesning!

NTNU Trondheim, 1. juni 2006

Innholdsfortegnelse

SAMMENDRAG	I
FORORD	III
INNHOLDSFORTEGNELSE	V
FIGUR- OG TABELLISTE	IX
1 INNLEDNING	1
1.1 BAKGRUNN FOR AVHANDLINGEN.....	2
1.2 PROBLEMSTILLING.....	2
1.3 AVGRENSNINGER.....	3
1.4 OPPGAVENS STRUKTUR.....	3
1.5 FRAMGANGSMÅTE.....	4
2 LENKEMODELLER	5
2.1 HTTP.....	5
2.2 LENKING I HTML.....	6
2.2.1 Anchor.....	6
2.2.2 LINK.....	7
2.2.3 Image og object.....	8
2.3 XLINK.....	8
2.3.1 Designmål for XLink.....	9
2.3.2 Simple links.....	10
2.3.3 Extended links.....	10
2.3.4 XLink navnerom.....	12
2.3.5 Elementer og attributter definert i XLink spesifikasjon.....	12
2.4 XPOINTER.....	16
2.5 MULIGHETER VED BRUK AV EXTENDED XLINK OG XPOINTER.....	17
2.5.1 Flerveislenker.....	17
2.5.2 Generiskelenker.....	17
2.5.3 Tredjeparts lenker.....	18
2.5.4 Lenkegrupper (lenkefiltrering).....	18
2.5.5 Dynamiske lenker.....	19
2.5.6 Beskrivende lenker.....	19
2.6 XHTML.....	19
2.6.1 HLink.....	20
2.6.2 Bruk av HLink.....	20
3 INTEGRERING AV AVANSERTE LENKESTRUKTURER	23
3.1 STILARK.....	23
3.1.1 Cascading Style Sheets.....	24
3.1.2 Extensible stylesheet language (XSL).....	25
3.1.3 DSSSL.....	27
3.1.4 XML User Interface Language.....	27
3.2 PROGRAMMERINGSVERKTØY.....	28
3.2.1 JavaScript, JScript og ECMAScript.....	28
3.2.2 JavaApplet.....	29
3.2.3 Flash.....	30
3.2.4 Sammenligning.....	31
3.3 ARKITEKTUR.....	32
3.3.1 Tjenerløsning.....	33

3.3.2 Proxy løsninger.....	33
3.3.3 Klientløsning.....	34
3.3.4 Plug-in løsning.....	34
3.3.5 Tjener Vs klient.....	35
4 STATE OF THE ART.....	37
4.1 NETTLESERE.....	37
4.1.1 Amaya.....	37
4.1.2 Mozilla.....	38
4.1.3 Doczilla.....	38
4.2 APPLIKASJONER FOR AVANSERTE LENKER.....	40
4.2.1 Xlip.....	40
4.2.2 XLinkit og XTooX.....	40
4.2.3 X2X.....	41
4.3 XLINK-TRANSFORMERERE.....	42
4.3.1 Goate.....	42
4.3.2 XLink2HTML.....	44
4.4 OPPSUMMERING.....	47
5 XLINK; DESIGN OG FUNKSJONALITET PÅ WEB.....	49
5.1 BRUK AV XLINK PÅ WEB.....	49
5.1.1 Ressursbaserte dokumenter.....	50
5.1.2 Tekstbaserte dokumenter.....	51
5.2 XLINK FUNKSJONALITET I NETTLESER.....	52
5.2.1 XLink-elementer på Web.....	52
5.2.2 XLink attributter på Web.....	53
5.3 DESIGN AV LENKER.....	55
5.3.1 Utforming av lenker ved adgang.....	55
5.3.2 Ankomst.....	57
5.4 UTFORMING AV XLINK I GRENSESNIITT.....	57
5.4.1 Struktur.....	58
5.4.2 Semantikk.....	59
5.5 XLINK I HTML.....	61
5.5.1 Trestruktur i HTML.....	61
5.5.2 XLink tooltip i HTML.....	63
5.5.3 Andre mulige XLink implementasjoner i HTML.....	64
6 TEST CASE; XLINK I RESSURSBASERTE DOKUMENTSYSTEM.....	67
6.1 BIBSYS FRBR-PROSJEKTET.....	67
6.1.1 FRBR.....	67
6.1.2 BIBSYS FRBRXML.....	69
6.2 XLINK I BIBSYS FRBR PROTOTYP.....	70
6.2.1 Datastruktur.....	71
6.2.2 Transformering til XLink.....	72
6.2.3 Fra XLink til HTML.....	74
6.2.4 Utforming i grensesnitt.....	74
6.2.5 Drøfting.....	77
7 ANALYSE.....	79
7.1 XLINK DATASTRUKTUR.....	79
7.1.1 Tredjeparts lenke.....	79
7.1.2 Lenker integrert i dokument.....	80
7.1.3 Organisering av XLink.....	82
7.2 UTFORMING OG BRUK AV XLINK.....	83
7.2.1 XLink i tekstbaserte dokument.....	83
7.2.2 XLink i ressursbaserte dokument.....	85

7.2.3 <i>Utforming av XLink ved tredjeparts lenking</i>	86
7.3 IMPLEMENTASJON AV XLINK FOR RESSURSBASERTE DOKUMENT	86
7.4 XLINK-STØTTE PÅ WEB (DESIGN)	88
7.4.1 <i>XLink simple VS extended</i>	90
8 KONKLUSJON	93
8.1 OPPSUMMERING AV ARBEID	93
8.2 KONKLUSJON	93
8.2.1 <i>XLink i tekstbaserte og ressursbaserte dokument</i>	93
8.2.2 <i>Resultat</i>	94
8.3 VIDERE ARBEID	95
REFERANSER:	97

Figur- og tabelliste

Figur 1 Traversering av simple XLink.....	10
Figur 2 Extented XLink med flere traverseringer	11
Figur 3 Traversering av extended XLink.....	11
Figur 4 XLink-elementhierarki	13
Figur 5 XSLT transformasjon	26
Figur 6 Applikasjon for transformering på tjener nivå	33
Figur 7 Applikasjon for transformering på Proxy nivå.....	34
Figur 8 Applikasjon for transformering på klient nivå	34
Figur 9 Grensesnitt for tredje parts lenker	38
Figur 10 Lenke fra ankeret "Doczilla 2.7" til fire forskjellige endepunkt	39
Figur 11 Endepunkt for en bi-direksjonell lenke i Doczilla	39
Figur 12 Meldingsdialog for lenker med flere endepunkt i Doczilla.....	39
Figur 13 Arkitektur for XLinkit.....	41
Figur 14 Mozilla screenshot av Goate	44
Figur 15 XLink presentert ved hjelp av XLink2HTML	46
Figur 16 Liste over antall studenter som er med i lenken, og til venstre alle kurs	46
Figur 17 Lenking i et ressursbasert HTML-dokument. Eksempel hentet fra Amazon	51
Figur 18 Figur 3 tekstutdrag fra http://www.thestreet.com	58
Figur 19 En-til-mange relasjon med extended XLink	61
Figur 20 XLink transformert til en uordnet HTML liste	61
Figur 21 Demonstrasjonssekvens av XLink i HTML.....	62
Figur 22 Demonstrasjonssekvens av XLink i HTML med CSS formatering.....	62
Figur 23 Utvidet XLink med metadata	63
Figur 24 XLink med tooltip	64
Figur 25 Entitetstyper og hovedrelasjonene i FRBR	68
Figur 26 Relationship elementer i en record.....	69
Figur 27 BIBSYS FRBR prototyp record i en nettleser	70
Figur 28 Overordnet struktur av XLink implementasjon	71
Figur 29 Gruppering av relasjoner "har del" i entiteten "Abbey road"	72
Figur 30 Mapping av attributter fra FRBRXml til XLink	73
Figur 31 XLink i BIBSYS	73
Figur 32 Pseudo kode for lenkeorganisering i HTMLs uordnet lister	74
Figur 33 Ankerets form.....	75
Figur 34 Horisontal lenkeplassering	76
Figur 35 XLink i grensesnitt	77
Figur 36 Utforming av lenker i grensesnitt	83
Tabell 1 XLink-elementer og attributter.....	12
Tabell 2 DHTML verktøy	32
Tabell 3 XLink-elementer i HTML	53
Tabell 4 XLink attributters bruksområde i HTML	54

1 Innledning

“A simple concept, the link has been one of the primary forces driving the success of the Web.” -World Wide Consortium [45]

Web¹ med sitt utspring fra Arpanet [13] fra 1967 er blitt en dominerende og verdensomspennende informasjonsressurs. Web kan ses som et nett av noder sammenbundet med lenker. Mye av suksessen til Web kan tilskrives dokumentformatet HTML² og de enkle lenkemulighetene som dette formatet tilbyr, samt HTTP³ protokollen som har gjort det mulig å gjøre adresserbare dokument enkelt tilgjengelig. Lenkene er ”limet” som knytter ressurser sammen på nett, og er med på å gjøre Web navigerbart.

Lenkemodellen til HTML begrenser struktur og beskrivelse av lenker på Web. XML⁴ [46] introduserer bruk av XLink for lenking mellom ressurser. Målet med XLink er å utvide mulighetene til lenking i hypermedia [17]. XLinks avanserte struktur og funksjonaliteten gir nye utfordringer for utforming av lenker i brukergrensesnitt, fordi det i motsetning til lenker i HTML ikke finnes noen felles konvensjon for hvordan XLink bør presenteres. Det er derfor nødvendig å få kartlagt bruksmulighetene til XLink, og undersøke mulighetene for å benytte de i samspill med eksisterende Web-teknologi. Utformingen og bruksnyttene for lenkene er kritisk for standardens overlevelsessevne.

Nettlesere i dag støtter primært lenking ved bruk av HTML. Markup-språk som XML har i senere tid blitt benyttet i større grad for å forfatte ressurser på Web. Databasesystemer som lagrer informasjon i XML, og deretter presenterer innholdet i form av HTML, er blitt mer vanlig ettersom flere benytter Web som portal for tjenestebaserte virksomheter, som for eksempel netthandel. Det som er felles for slike dokumentsystemer er at det ofte presenteres mange relasjoner mellom ressursene. Noe som ofte fører til store utfordringer for utforming av lenker i grensesnittet. Det er derfor et behov for å kunne presentere avanserte lenker i slike systemer.

¹ World Wide Web

² HyperText Markup Language

³ HyperText Transfer Protocol

⁴ eXtensible Markup Language

1.1 Bakgrunn for avhandlingen

I løpet av de årene som er gått siden Theodor Holm Nelson [47] innførte begrepet hypertekst er mange lenkingsmodeller prøvd ut, HTML er veldig enkel i forhold til mange av disse. Etersom Web har utviklet seg har er flere blitt klar over begrensningene lenkemodelle:

- HTML kan bare beskrive enveis relasjoner.
- Integreerte anker begrenser muligheten til å lenke til eller fra materiale der en bare har leserettigheter.
- Det er vanskelig å finne inngående lenker til dokument i HTML.
- HTML støtter ikke rike lenker. Rike lenker er lenker som inneholder metadata.
- Kan ikke separere struktur fra innhold. Dette er noe som gjør at det ikke er mulig å opprette databaser med lenker mellom dokument. [41]

Det er derfor ønskelig å se på behovet for avanserte lenker der HTML blir for begrenset. XLink er en mulig løsning, men er fortsatt en spesifisering det er lite støtte for og det finnes lite kunnskap om muligheter og problemer med denne.

1.2 Problemstilling

Arbeidet med XLink har foregått siden "XML part 2:linking" ble publisert av W3C i 1997 [48]. Dette er en lenkingsmodell som er ment å være mer sofistikert enn lenking i HTML og samtidig enklere enn lenking i SGML⁵ [50]. Den første XLink spesifiseringen fra W3C gruppen kom i 2001. XLinks mål er å forbedre brukernes navigeringsmuligheter, gjennom å gi muligheter for mer avansert funksjonalitet knyttet til lenking i en stadig voksende mengde informasjon på Web.

Denne avhandlingen tar med utgangspunkt i begrensningen ved lenking i HTML for seg hvordan avansert lenkestruktur kan brukes til å gi ny funksjonalitet i lenking på Web. Det blir sett på mulighetene for integrering av XLink i HTML-dokumenter ved bruk av teknologi som allerede er støttet av en stor andel nettleserplattformer. Det er ønskelig å gå

⁵ Standard Generalized Markup Language

nærmere i sømmene på bruk og muligheter med XLink, behov for slik lenking og utfordringene til utforming avanserte lenker fører med seg.

Denne avhandlingen vil:

- Kartlegge og redegjøre for bruksområdene til XLink i Web-baserte hypermediasystem.
- Se på hvordan datastrukturen i XLink kan tas i bruk i Web-baserte dokumenter.
- Analysere hvordan dagens Web-relaterte teknologi kan bidra til å realisere løsninger basert på XLink.
- Undersøke design-utfordringer som XLink medfører.

1.3 Avgrensninger

Problemstillingen for avhandlingen har utviklet seg fra å omhandle implementering av XLink i hypermediasystemer generelt til fokus på integrering på Web gjennom HTML-dokument. Avgrensningen har kommet naturlig, da det i seinere tid viste seg at hypermediasystemer generelt ville blitt et for stort omfang for avhandlingen. Fokus er rettet mot ressursbaserte Web-dokument. Dette er dokumenter som beskriver metadata om en ressurs som kan etterspørres, for eksempel dokumentene i Web portalen til en netthandler.

1.4 Oppgavens struktur

Denne avhandlingen har som mål å forklare hvordan den avanserte lenkemodulen XLink kan benyttes i dagens nettlesere. I kapittel 2-4 blir det gjort rede for nødvendig teori for å få forståelse for strukturene, bruk av anvendbar teknologi og hva som allerede er gjort på feltet, mens kapittel 5-8 inneholder det praktiske og analyseringen av XLink i forhold til bruk på Web.

- Kapittel 2 redegjør for lenkemodeller som benyttes på Web. Det er lagt vekt på XLink og XPointer og deres bruksområde, men også lenkemodell for HTML, og XHTML⁶

⁶ eXtensible HyperText Markup Language

- Kapittel 3 beskriver hvordan XLink kan tas i bruk på Web. Kapittelet tar for seg web-teknologier og hvordan disse kan brukes i forhold til å benytte XLink på Web. De arkitektoniske mulighetene for håndtering av XLink vil også bli vurdert.
- Kapittel 4 tar for seg tidligere arbeid gjort med XLink.
- Kapittel 5 deler Web-dokumenter i to hovedtyper og redegjør for behovet og design implikasjonene for avanserte lenkestrukturer i disse dokumenttypene. Kapittelet beskriver så XLink funksjonalitet og utforming på Web.
- Kapittel 6 demonstrerer hvordan XLink kan tas i bruk i et ressursbasert dokumentsystem, med Web portal. Det blir sett på hva som er mulig, og hvilke komplikasjoner en transformasjon fra en tidligere lenkemodell til XLink vil fører med seg.
- Kapittel 7 analyserer erfaringene som er opparbeidet gjennom arbeidet med XLink. Kapittelet vurderer også XLinks-datastruktur, utforming av XLink og applikasjonen som er implementert
- Kapittel 8 konkluderer og oppsummerer resultatene av denne avhandlingen. Det blir også foreslått videreutvikling og videreføring av arbeid.

1.5 Framgangsmåte

I arbeidet med avhandlingen er det først sett på XLink strukturen og muligheter med denne. Det er så sett på teori, spesifikasjoner og eksisterende systemer som omhandler/benyttter XLink. Videre er det blitt definert hvilke behov det er for bruk av XLink funksjonalitet på Web i dag, for deretter å implementere en løsning på dette. Løsningen integreres og testes i en prototyp av BIBSYS. Arbeidet med XLink blir så evaluert, og framtidig arbeid blir foreslått.

Teorien for kapittelet "Lenkemodeller" er hentet fra W3C spesifikasjonene for de ulike lenkemodellene. I kapittel "Integrering av avanserte lenkestrukturer" og "State of the art" er teori hentet fra artikler, og i noen tilfeller Webressurser. Videre har boka "XPath, XLink, XPointer and XML" [41], og artiklene "Comparing Link Marker Visualization Techniques" [34] og "*The Look of the Link*" [35] vært essensiell for avhandlingen.

2 Lenkemodeller

“Hyperlinks were by far the preferred method of traversal, accounting for 52% of all document requests.” -Resultat fra “Characterizing browsing strategies in the world-wide web” [11]

Selv om WWW har vokst siden 1989 har disse tekniske spesifikasjonene forandret seg lite. XML introduserer en betydelig forandring på måten elektroniske dokument kan struktureres. XML er et mer omfattende og utvidet dokumentformat for publisering av dokument i hypermediasystemer. Dette er et fleksibelt tekstformat utviklet fra SGML (ISO 8879), som i utgangspunktet er designet for storskala elektroniske publiseringer[46]. I etterkant av XLink har også en standard for lenking i XHTML-dokumenter blitt definert, denne kalles HLink og bygger på den fleksible og utvidbare tankegangen fra XLink.

2.1 HTTP

HTTP er i dag den mest brukte protokollen for utveksling av informasjon på Web. Det er denne protokollen som gjør det mulig å kommunisere eksternt i et nettverk. Dette er en protokoll for forespørsel og respons mellom klienter og tjenere. En nettleser er en HTTP-klient som ved bruk av en nettverksprotokoll TCP (Transmission control protocol) skaper en forbindelse over en pakkeleveringstjeneste kalt Internet protocol (IP) til en spesiell port på en annen vert. En HTTP-tjener ”lytter” på denne porten venter på at klienten skal sende en forespørsel som for eksempel ”get/http/adresse”, som vil føre til at tjenerer returnerer det dokumentet med forespurt adresse. Forskjellen mellom HTTP og de fleste andre TCP baserte protokoller er at en forbindelse termineres etter at en forespørsel er fullført. Dette gjør HTTP velegnet for Internet fordi Webressurser lenker til andre tjenere, ved at forbindelsen avsluttes kan nye forespørsler sendes. Lokasjoner til HTTP ressurser oppgis ved bruk av URI (uniform resource identifier). Dette er en identifikator for bruk i HTTP bestående av et navn eller en adresse som representerer en plassering til en ressurs. Det er denne adressen som benyttes når en skal opprette lenker mellom eksterne dokumenter på Web.

2.2 Lenking i HTML

HTML er et markup-språk for utvikling av Websider som kan bli visst i en nettleser, og benyttes til å beskrive både utforming og semantikk til et dokument på Web. HTML-dokumenter består ofte av innhold og lenker til andre ressurser som kan forespørres gjennom HTTP.

Lenker i HTML[45] har to ender, disse er kilde og målressurs. Med en ressurs menes en lenkbar del av informasjon, som for eksempel et bilde, en film, et lydklipp, et HTML-dokument og lignende. Lenker skrives i HTML ved å benytte A (anchor), img (image), object eller Link element. A-lenker er de som benyttes i dokumentstrukturen for å opprette en referanse fra kildedokumentet til måldokumentet. A-lenkene kan inneholde attributtet name som gjør det mulig å referere til en eksakt posisjon innen et dokument. Elementet img brukes for å integrere et bilde i HTML. Multimedia integreres i HTML-dokument med bruk av object. Link benyttes for å referere til stilark som for eksempel CSS eller for å lage menylenker. Dette elementet kan bare benyttes i head delen av dokumentet.

2.2.1 Anchor

Lenker i HTML har en bestemt oppførsel og funksjon. Ved å klikke på en lenke lastes et måldokument inn i nettleser. A-lenker har fått en standard konvensjon for utforming, nemlig de blå understrekte lenkene. Lenkene skifter farge etter at de har blitt traversert slik at vi kan kjenne igjen de lenkene vi allerede har traversert. Denne utformingen stammer fra hypermediasystemet Mosaic og har senere blitt adoptert av de fleste store nettleserne [35].

For at en lenke i HTML skal være brukbar så må en URI defineres, dette gjøres gjennom attributtet href. Det er definert flere attributter for lenking i HTML. I den nyeste standarden; HTML 4.01[45], kan blant annet disse attributtene benyttes i A elementet;

- name; identifiserer lenkens navn slik at lenken kan være destinasjon til en annen lenke.
- type; informerer om måldokumentets type. En brukeragent kan for eksempel benytte en tilbakefallsmetode om den ikke støtter ressursens format.

- rel; dette attributtet brukes for å beskrive relasjonen fra lenkens kilde til målressurs.
- rev; benyttes for å beskrive lenkens tilbakegående ressurs, dette kan for eksempel være forfatterens hjemmeside.
- target; definerer hvor lenken blir åpnet. Denne kan ha verdien, _blank som åpner ressursen i nytt vindu, _self laster lenken i samme vindu, _parent tillater ressursen å lastes inn i rammen som lenken befinner seg i, _top laster lenken i "kroppen" til HTML-dokumentet.
- title; inneholder semantisk informasjon ment for brukerne, kan for eksempel benyttes i et tooltip⁷.

Lenker i HTML kan ikke være "nøstet", noe som betyr at de bare kan ha ett mål definert per lenke. Dette vil si at slike lenker bare kan definere en lenke fra en ressurs til en annen og er derfor enveis.

Name

I HTML kan attributtet Name brukes for å lokalisere en bestemt plassering innen et HTML -dokument. Dette attributtet kan benyttes i A-elementer i HTML, og gjør det mulig å referere direkte til elementets plassering. En slik plassering kan for eksempel være starten av et avsnitt, starten av en bestemt setning eller til en lenke. Dette gjør det mulig å definere mer eksakte lenker. Attributtet benyttes på slutten av URIen som i en lenke. En lenke til en eksakt plassering skrives som ``.

2.2.2 LINK

Dette er et lenkeelement som hører til i den delen av HTML-dokumentet som er definert som HEAD. Det er brukt for å indikere relasjonen mellom dokumentet og andre ressurser. Elementet benytter de samme attributtene som A-lenker, men er ment for å presentere forfatter, relaterte dokumenter, ordbøker, eldre versjoner av dokumentet og lignende. Det var opprinnelig tenkt at denne lenken skulle være til støtte for nettleser, slik at

⁷ Tooltip er en tekstboks med forklarende tekst som blir aktivisert når musen traverserer over et bestemt område.

dokumentets plassering i et hierarki kunne identifiseres og navigeres i, som for eksempel relasjon til ”neste”, ”forrige”, ”foreldre” dokument.

2.2.3 Image og object

Image benyttes for å integrere bilder og figurer i et HTML-dokument, og presentere de med teksten, i motsetningen til å ha A-element som lenker til et bilde som hentes inn i og vises i nytt vindu. Elementet img refererer til plasseringen av bildet for så å laste det opp når det blir etterspurt. Dette skjer som oftest automatisk når nettsiden lastes inn i nettleseren, men kan også spesifiseres til å lastes inn på kommando fra bruker. A-lenker kan kombineres med img referanser, på denne måten kan klikkbare bilder opptre som anker i HTML-dokument.

Elementet object spesifiserer data og parametere for å plassere objekt i HTML dokument. Slike objekter er for eksempel bilder, video, lyd, JavaApplets og lignende.

2.3 XLink

“The model defined in this specification shares with HTML the use of URI technology, but goes beyond HTML in offering features, previously available only in dedicated hypermedia systems, that make hyperlinking more scalable and flexible” – World wide web consortium [54]

XLink benytter på samme måte som A-lenker i HTML, URI for å adressere til eksterne ressurser. I denne lenkemodulen kan alle elementer bli definert som lenkeelementer. Dette gjøres ved at elementene inneholder attributter som inneholder informasjon som gjør de traverserbar. XML spesifikasjonen inneholder ingen innbygd metode for lenking, men forslag til struktur for lenkene gjennom et spesifisert vokabular. Det er altså ikke noen forhåndsdefinert måte å utforme et XLink-element på, som i HTML der lenkeelement blir utformet med blå tekst og understreking. XLink benytter XML markupspråk for å definere alle brikkene som er nødvendig for å strukturere lenker. Dette gjør at lenkene må være ”well formed”. At et dokument er ”well formed” betyr at dokumentet må være organisert slik at alle element som åpnes også lukkes, og at elementene er

organiser i rekkefølge slik at de ikke overlapper. Et eksempel på dette er at `<link><loc>tekst</loc></link>` er ”well formed”, mens `<link><loc>tekst</link></loc>` ikke er ”well formed”. Fordelen ved å benytte XML markup-språk er at det gjør lenkemodellen skalerbar og utvidbar.

XLink deler lenking i to uavhengige konsepter, simple og extended lenker. Lenker som er simple er et konsept som har en klar analogi med lenker sånn vi kjenner dem fra HTML. Disse lenkene har en enkel struktur og gir lik funksjonalitet som A-lenker i HTML. Det er derimot ingen åpenbar analogi for å forklare extended links, fordi extended XLink er basert på lenking i mer avanserte hypermediasystemer.

2.3.1 Designmål for XLink

World Wide Web Consortium definerer i ”XML Linking Language (XLink) Design Principles” [54] flere mål for design og bruk av XLink. Noen av målene med design av XLink er:

- XLink skal være lett å benytte på Web.
 - Dette relaterer til at enkle lenkestrukturer som allerede er i bruk fremdeles skal virke i XML. I tillegg sier dette målet noe om at XLink skal kunne benyttes på Web uten å være for komplisert.
- XLink skal skrives i XML.
 - Målet er ikke å lage ett helt nytt språk men å skape et vokabular for lenking i XML, siden XML markup-språk benyttes unngår forfatterne og utviklerne å lære et nytt språk for å skrive lenker i XML.
- XLink skal være lesbar for mennesker.
- XLink skal være enkel å implementere.
- XLink skal representere den abstrakte strukturen og de betydningsfulle delene ved lenking.
 - Designerne av XLink ønsket at lenkemekanismen skulle kunne mer enn bare å lenke mellom dokumenter i XML. XLink skal kunne beskrive

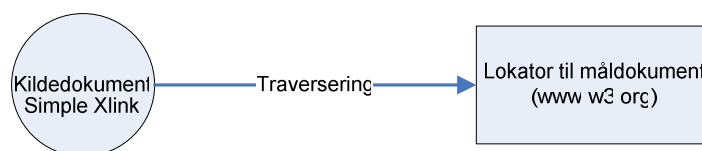
semantikken til lenker og beskrive deres relasjoner til objekt og ressurser som de kobler sammen.

- XLink skal kunne ligge utenfor dokumentet der ressursen ligger.
 - Dette punktet relaterer til utvidbarheten ved XLink, og mulighetene ved å kunne representere mer enn bare enveis lenker integrert i et dokument.

2.3.2 Simple links

Lenker som er simple er integrert i dokumentet, kan bare koble sammen lenker i en retning (enveis) og har bare en målressurs. Kilderessursen til simple XLink er implisitt den plasseringen lenken befinner seg på i det dokumentet den er skrevet i. Lokatorelementet inneholder data om plasseringen til destinasjon objektet på samme måte som href attributtet i HTML -lenker. xlink:type attributtet definerer om lenken er enten simple eller extended. En simple Xlink kan skrives slik:

```
<simpleLink xlink:type="simple" href="HTTP://www.w3.org/TR/1998/WD-XLink-19980303#simple">
  Lenkens anker tekst
</simpleLink>
```



Figur 1 Traversering av simple XLink.

2.3.3 Extended links

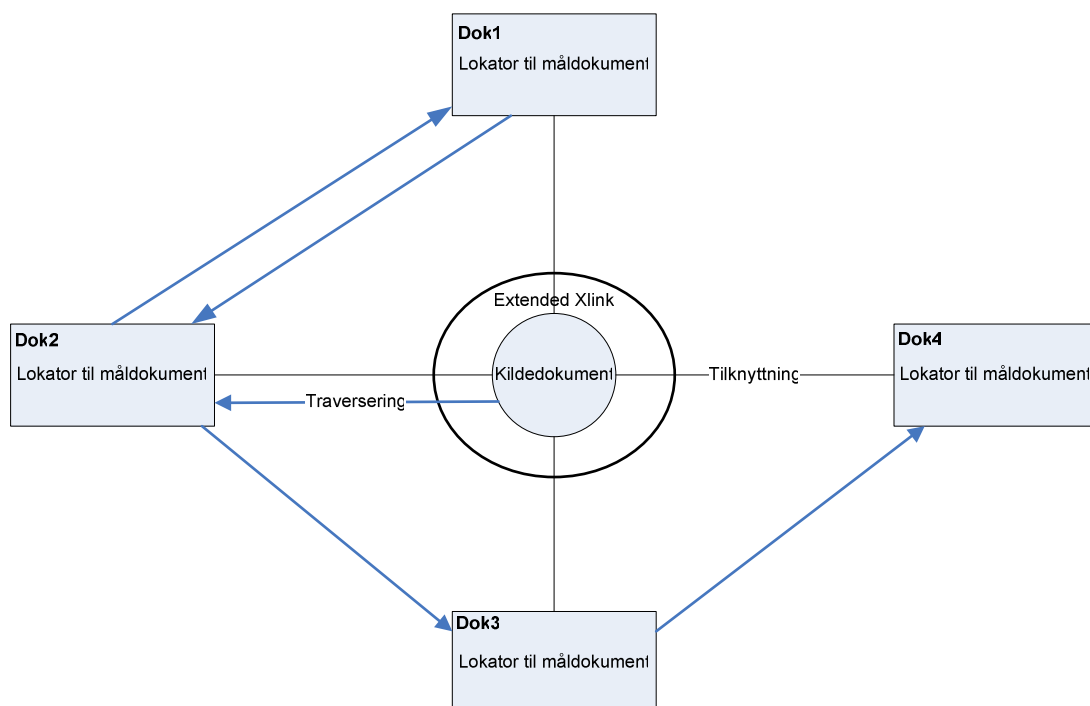
Lenker som er extended tilbyr i motsetning til simple lenker full XLink funksjonalitet. Xlink av typen extended kan koble sammen flere ressurser, defineres uavhengig av kildedokumentet, og skrive lenker med rik semantikk. En extended XLink består av elementer for beskrivelse av mål- og kilde- ressurser og separate elementer for å beskrive traverseringen. I eksemplet under ser vi en XLink der flere ressurser er innblandet.

Elementene "ressurs" er relasjonene som er med i XLinken, mens "go" definerer traverseringen mellom ressursene.

```
<extendedLink xlink:type="extended" XMLNs:XLink="HTTP://www.w3.org/1999/XLink">
  <ressurs xlink:type="resource" xlink:title="Kildedokument" xlink:label="Kildedok"/>
  <ressurs xlink:type="locator" xlink:title="document 1" xlink:label="Dok1"/>
  <ressurs xlink:type="locator" xlink:title="document 1" xlink:label="Dok2"/>
  <ressurs xlink:type="locator" xlink:title="document 1" xlink:label="Dok3"/>
  <ressurs xlink:type="locator" xlink:title="document 1" xlink:label="Dok4"/>
  <go xlink:type="arc" xlink:from="Kildedok" xlink:to="Dok2"/>
  <go xlink:type="arc" xlink:from="Dok2" xlink:to="Dok1"/>
  <go xlink:type="arc" xlink:from="Dok1" xlink:to="Dok2"/>
  <go xlink:type="arc" xlink:from="Dok2" xlink:to="Dok3"/>
  <go xlink:type="arc" xlink:from="Dok3" xlink:to="Dok4"/>
</extendedLink>
```

Figur 2 Extended XLink med flere traverseringer

Figur "Traversering av extended XLink" demonstrerer hvordan XLinken som er skrevet over vil traverseres.



Figur 3 Traversering av extended XLink.

2.3.4 XLink navnerom

Siden XLink skrives i XML er det fritt for å definere egne navn på elementer og attributter. Alle elementer i XML kan derfor være lenkeelementer. Dette kan føre til konflikt, der flere dokumenter benytter samme attributtnavn i ulik betydning. Det er derfor nødvendig å definere en universell navnestandard. Dette gjøres ved at dokumenter identifiserer et navnerom gjennom en URI der alle ”lovlige” navn er definert. Navnerommet til XLink identifiseres ved HTTP://www.w3.org/1999/XLink. Dette er ikke en adresse til en nettside men et logisk navn til et fellesrom som mange programmer vil tolke likt. Elementer og attributter knyttes til dette navnerommet ved å benytte prefiks som indikerer hvilket navnerommet de tilhører.

xlink:type=	simple	extended	resource	locator	arc	title
xlink:title	O	O	O	O	O	
xlink:href	O		O	R		
xlink:label			O	O		
xlink:role	O	O	O	O		
xlink:arcrole	O				O	
xlink:show	O				O	
xlink:actuate	O				O	
xlink:from					O	
xlink:to					O	

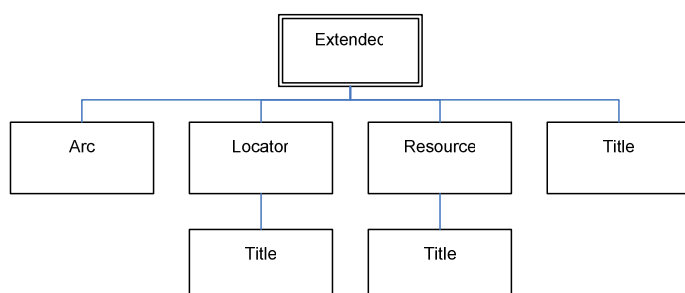
Tabell 1 XLink-elementer og attributter (“O” markerer hvilke attributter som valgbart kan benyttes i elementene, “R” er de attributtene som er nødvendig)

2.3.5 Elementer og attributter definert i XLink spesifikasjon

XLink har flere globale elementer og attributter identifisert av XLink-navnerommet. Disse definerer lenkemodulen og hvilke muligheter bruk av denne spesifikasjonen vil føre med seg. Samspillet mellom elementene og attributtene i XLink er det som gir denne lenkemodulen mer avansert funksjonalitet enn i HTML-lenking.

Elementer i XLink

En XLink har seks forskjellige elementer disse er simple, extended, resource, locator, title og arc. Disse typene brukes for å identifisere elementene og hvilken rolle de har. Element av typen simple består av et element og kan ikke inneholde underelementer. Element av typen extended kan inneholde alle elementene i spesifikasjonen utenom simple. Ved å benytte flere elementer til å beskrive en lenke, separerer XLink beskrivelse av ressurser fra informasjon om traversering mellom dem. Elementet resource beskriver kilderessursens mens locator innehar informasjon om målet til traverseringen. Traversering av XLink blir definert av arc elementet. Dette elementet spesifiserer relasjonen mellom resource og locator elementene. Ved å separere traverseringsinformasjon fra ressursene, muliggjøres traversering fra flere lokale ressurser til flere destinasjoner. Kilde og mål blir identifisert av attributtene xlink:from og xlink:to. På denne måten kan multidireksjonale lenker implementeres på en strukturert måte. Elementet title er lite omtalt i XLink spesifikasjonen, men er tenkt til å kunne tilføre ytterligere beskrivelse til elementet. Dette kan brukes som løvnode for elementene, extended, locator og resource.



Figur 4 XLink-elementhierarki

Attributter for traversering

Attributtene som beskriver lokasjon, relasjon og traversering er ved extended XLink er xlink:href, xlink:label, xlink:to, xlink:from, xlink:type og xlink:arcrole. Disse attributtene har forskjellige roller ved å beskrive en traversering i XLink, noe som muliggjør avanserte relasjoner og traverseringer. En extended XLink kan eksistere uavhengig av kildedokumentet fordi det er mulig å definere ”fra” og ”til” attributtene ved en traversering.

xlink:href

Dette attributtet er en destinasjonslokator, og inneholder URI til lenkens destinasjon. Attributtet har også muligheten til å benytte seg av XPointer, noe som muliggjør lenking til en spesiell plass i et dokument. I likhet til HTMLs href attributt, inneholder dette attributtet en adresse til en ressurs.

xlink:label

Dette attributtet identifiserer elementets plass i en traversering. Attributtet benyttes av arc elementet for å identifisere locator og resource elementenes posisjon i en lenke.

xlink:to og xlink:from

Disse to attributtene defineres i arc elementet, for å spesifisere kilde og målressurs ved en traversering. Verdiene som benyttes i disse attributtene er må samsvare med verdiene brukt i xlink:label. Gjennom disse attributtene knytter xlink:to og xlink:from sammen locator og resource elementene.

Det kan tenkes at resource elementene gis xlink:label verdi "foreldre" og locator elementene verdi "barn". Disse knyttes sammen gjennom å definere xlink:from="foreldre", og xlink:to="barn" i arc elementet. Attributtene xlink:from og xlink:to knytter alle ressursene som har label verdi lik disse tekststrengene. Dette gjør at vi i en lenke kan gi ressursen "mor" og "far" verdi lik "foreldre" og barna verdi "barn". Alle elementene med verdi "foreldre" er da relatert til de med verdi "barn". Er det et element med verdi "foreldre" og flere elementer med verdien "barn", får vi en en-til-mange relasjon.

xlink:type

Attributtet xlink:type beskriver hvilken type elementet er. Dette attributtet kan ha verdien simple, extended, resource, locator, arc eller ingen verdi.

Attributter for oppførsel ved traversering

XLink attributtene xlink:actuate og xlink:show er de eneste attributtene som definerer oppførselen til XLinken. Men disse er valgbare og tolkningen av XLink bestemmes av applikasjonen som prosesserer lenken. I XLink er det applikasjonsavhengig hvordan lenkene utformes og vises. Det er flere tenkte eksempler på hvordan dette kan gjøres.

Alle ressursene kan for eksempel åpnes samtidig, lenken kan aktiviseres gjennom museklikk, det kan brukes logikk for å løse for å velge lenker (adaptive system) og lignende. Det er applikasjonene som prosesserer lenken som bestemmer utforming og oppførsel, men XLink har tre attributter for å gjøre valg om hvordan lenker skal traverseres. Denne informasjonen kan tas i bruk av en applikasjon.

xlink:show

Dette attributtet forklarer hvordan lenken skal prosesseres, og sier hvordan objektet som blir forespurt skal vises. Attributtet `xlink:show` kan ha verdiene "replace"; det nye objektet erstatter den ressursen som er i nettleseren, "new"; objektet åpnes i nytt vindu, og "embeded" laster inn den nye resursen som en del av kildedokumentet. Attributtet kan for eksempel benyttes om en ønsker å lenke til en ordforklaring, gis `xlink:show` verdien "embeded" vil forklaringen integreres der lenken er i kildedokumentet.

xlink:actuate

Dette attributtet sier noe om hvordan lenken traverseres, og kan ta to verdier. Ved "auto" traverseres lenken automatisk når den blir prosessert av en applikasjon. Inneholder attributtet verdien "user" blir lenken traversert når den aktiviseres av en ekstern mekanisme, som for eksempel et museklikk.

2.3.5.4 Semantiske attributter

Disse attributtene beskriver XLinken, hva den er, hvordan rolle den har og hvilken gruppe lenker den tilhører. Attributtene `xlink:role` benyttes for å beskrive lenken på en maskinprosesserbar måte, mens `xlink:title` beskriver lenken for brukeren.

xlink:role

Attributtet `xlink:role` forteller applikasjonen eller programvaren hvilken kontekst lenken er i, og om lenkens helhet. Dette attributtet legger til informasjon som forteller hvilken gruppe lenken tilhører, som for eksempel ordforklaring, bakgrunnsinformasjon, teknisk informasjon og lignende. Attributtet defineres som en URI som peker til en destinasjon der lenkegruppen blir definert. Er for eksempel ressursen en locator som refererer til www.w3c.org/tr/XLink, som er den tekniske avhandlingen til XLink, så defineres `xlink:role`

attributtet som `www.w3c.org/tr`. Dette gir oss informasjon om at denne XLinken tilhører en gruppe tekniske rapporter av W3C.

xlink:arcrole

Dette attributtet definerer rollen til en traversering. Attributtet er meget lik `xlink:role` attributtet, men defineres for arc-elementer, og gjør det mulig å definere lenkegruppen til traverseringen.

xlink:title

Tittel attributtet til XLink inneholder semantisk informasjon ment for brukeren. Dette kan være ankeret til lenken eller en forklarende tekst som presenterer lenken. Dette kan for eksempel benyttes som den klikkbare teksten i et dokument eller som et tooltip.

2.4 XPointer

XPointer [55] er en mekanisme for å definere et fragment av en ressurs. Ved hjelp av XPointer kan en referere til en bestemt destinasjon innen et dokument. Denne fragmentidentifikatoren for XML dokumenter tillater adressering inn i den hierarkiske dokumentstrukturen basert på verdier som elementtype, attributtverdier, karakterinnhold og relativ posisjon. XPointer er applikasjon av XPath⁸ [15], som er et uttrykksspråk i XSL [49]. XPath er utviklet for å kunne benyttes som et grunnlag for utvikling av andre standarder som XPointer. Ved å utvide XPath med XPointer muliggjøres [55] flere funksjoner enn ved bruk av bare XPath. Noen av disse er:

- Adressering til punkt og deler av tekst.
- Lokalisering av informasjon ved hjelp av ”String matching”.
- Benytte adresseringsuttrykk i URI referanser som fragment identifikatorer.

⁸ XML Path language

2.5 Muligheter ved bruk av extended XLink og XPointer

Ved å benytte extended XLink og XPointer kan en få lenker med meget avansert funksjonalitet. XPointer kan benyttes slik at lenker kan defineres til eksakte posisjoner i et dokument, eller bare et fragment av dokumentet. Dette samspillet gjør at lenker kan spesifiseres for dokumenter som en ikke har skrivetilgang til. Benyttes XPointer i samspill med Xlink får vi tre forskjellige alternativer for å skriving av lenker:

1. simple XLink kan bare integreres innen dokumentet. Ved å bruke XPointer kan destinasjon til lenken identifiseres mer nøyaktig. Kan bare skrives av dokumentets forfatter.
2. extended XLink uten XPointer kan bare implementeres i kildedokumentet. Vi er derfor også her avhengig av å ha skriverettigheter til dokumentet. Dette er en integrering av XLink på mellomnivå.
3. extended XLink med XPointer kan i tillegg til å integreres i kildedokumentet eksistere uavhengig av andre dokument. Ved å kunne definere plassering av ankeret i kildedokumentet med XPointer, kan lenken både definere kilde og mål, og kan derfor eksistere utenfor kildedokumentet. Dette er tredjeparts lenking.

2.5.1 Flerveislenker

Flerveislenker er lenker som binder sammen multiple ressurser. Siden extended Xlink separerer traversering med arc elementet kan en i tillegg til en-til-en relasjoner definere traverseringer mellom en-til-mange, mange-til-en og mange-til-mange ressurser. Dette gir brukerne muligheten til å velge mellom forskjellige kilder av informasjon. Gjennom å gruppere lenker i et binært hierarki i XLink, vil lenkene være enklere å organisere i et grensesnitt. Flerveislenker i HTML kan ofte bli en påkjenning for grensesnittet.

2.5.2 Generiskelenker

Generiskelenker er lenker som kan benyttes til for eksempel ordforklaring. En kan for eksempel opprette en lenke fra hver instans av ordet "generisklenke" i et dokument, til en målressurs med definisjonen av dette uttrykket. Generiske lenker kan definere lenker ut i fra et bestemte mønstre til lenkeankeret ved å benytte XPointer og dens "stringmatch" metode. Disse kriteriene kan være elementer av en gitt type, elementer som

har attributter med en spesiell verdi eller tekst som er lik en gitt tekststreng. Kildeankeret er altså definert av et bestemt mønster. På denne måten kan lenkene eksistere uavhengig av kildedokumentet og legges inn på forespørsel. Dette gjør at vi kan gjenbruke lenker, og få mange kilder per destinasjon, altså en mange-til-en lenke. På denne måten kan det lenkes til dokumenter der man i utgangspunktet ikke har skriverettigheter.

2.5.3 Tredjeparts lenker

En av de mest revolusjonerende mulighetene med XLink er det å kunne definere lenker mellom dokument uten å ha skriverettigheter til noen av dem. Dette er lenker som går fra en utenforstående resurs til en lokal ressurs. Et XML-dokument som inneholder mange tredjeparts lenker kalles en lenkebase. En lenkebase oppretter lenker mellom dokumenter som ikke er lenkebasen selv. [17]

For å kunne skrive tredjeparts lenker er man avhengig av XPointer eller en annen fragmentidentifikator til å referere til traverseringsressursen. En lenkedatabase kan muligjgjøres i extended XLink ved å identifisere XPointer-informasjon som referer til kildedokumentet i XLinks resource elementet, deretter defineres ressursen en ønsker å lenke til ved å benytte et locator element, for så å opprette en arc som definerer traverseringen mellom kildedokumentet og målet. Denne informasjonen lagres på en tjener og hentes etter forespørsel. En slik database kan for eksempel inneholde ordforklaringer som lenker hvert tilfelle av et ord opp mot en forklaring (generisk).

2.5.4 Lenkegrupper (lenkefiltrering)

Lenker av lik karakter kan samles i grupper. Det som definerer hvilken gruppe lenken er i, er verdien til attributtet xlink:role eller i traverseringsfasen gjennom attributtet xlink:arcrole. Disse attributtene hjelper til å organisere lenkene på en oversiktlig måte. Fordelen med slike lenkegrupper er at en applikasjon kan behandle XLinken slik at grupper av lenker kan skrus på/av etter ønske. Har man definert en gruppe lenker som ”forklarende lenker” kan disse i tilfeller der man leser en tekst med mye fremmedord være greie å aktivere. Ved å kunne skru av lenker vil en kunne øke lesbarheten til dokumentene. Problemet med å lage en generell applikasjon for dette er at det ikke defineres noen verdier for

gruppenavn i XLink-navnerommet. Det finnes ingen forslag/definisjon for global bruk av gruppenavn i XLink. [42]

2.5.5 Dynamiske lenker

Dynamiske lenker er lenker som forandrer struktur eller oppførsel over tid eller etter forskjellige brukere. Dette kan være forandring i destinasjon eller oppførsel som om lenken åpnes i nytt vindu og lignende. Det kan tenkes i bruk hvor for eksempel lenkens destinasjon er en tjeneste. Lenken kan forandres etter hvilket land man bor i, slik at for eksempel språket passer brukeren. Det kan også benyttes forskjellige lenkegrupper etter hvilke brukere som benytter informasjonskilden. Lenkene kan automatisk filtreres etter informasjon som er lagret om brukerne i for eksempel HTTP cookie. [24]

2.5.6 Beskrivende lenker

Ved at en XLink skrives i XML, kan en slik lenke inneholde en rekke elementer som beskriver resursen. En XLink kan inneholde en rekke barnenoder, disse defineres som tittel element. Slike elementer kan knytte metadata til ressursene. Disse elementene kan inneholde informasjon om for eksempel forfatter, type innhold, størrelse på måldokument og lignende. Denne informasjonen er ment for å kunne gi en utvidet beskrivelse av ressursene som inngår i en relasjon. Informasjonen gir brukerne en bedre forståelse av måldokumentet, og skal være med på å forårsake færre feiltraverseringer. Informasjon om lenkene er derfor med på å skape hypermediasystem som er mer konsis og bedre å navigere i.

2.6 XHTML

For å bygge bro mellom HTML og XML har W3C kommet med XHTML [51]. XHTML er et markup-språk som bygger på HTML og XML. Markup-språket oppstod da en følte at en mer strikt syntaktisk markup-språk var nødvendig, siden Web-innhold nå trenger å bli levert til mange forskjellige enheter, som mobiltelefon, PDA, etc., som ikke kan støtte kompleksiteten som kommer i tillegg når HTML brukes. I forhold til HTML som har ferdigdefinerte tagger gir XHTML muligheten til opprette egne tagger og attributter. XHTML må også være "well formed" som i XML. Dokumenttypen er basert på XML og

er designet for å kunne fungere sammen med XML baserte brukeragenter. Meningen med språket er at det skal kunne brukes til å beskrive innhold som er XML tilpasset, og kan fungere på HTML 4 brukeragenter.

XHTML 2.0 [52] som enda er under utvikling er ikke tilbakekompatibel med den forrige utgave av standarden. Denne utgaven er mer likt XML enn den forrige utgaven. Den er mer syntaktisk og mindre semantisk og dermed mer fleksibelt og utvidbart. Standarden er kompatibel med XML standardene XLink og XForm.

2.6.1 HLink

"...specify which attributes of elements represent Hyperlinks, and how those hyperlinks should be traversed, and extends XLink use to a wider class of languages than those restricted to the syntactic style allowed by XLink." –World Wide Web Consortium[44]

XLink-standarden for lenking kan benyttes i XHTML, men grunnet at XLink bare beskriver strukturen til lenkene og i liten grad sier noe om oppførselen til lenkene, har arbeidsgruppen for W3C definert en lenkemodul for XHTML, kalt HLink. Mye av motivasjonen for å opprette HLink var at arbeidsgruppen for HTML ønsket at lenkene skulle benytte href og ikke andre attributter for å identifisere en lenkes destinasjon [5]. HLink utvider XLink, derfor er mye av markup-språket hentet fra XLink-spesifikasjonen. Forskjellen er at HLink kan ha flere elementer og attributter for oppførsel. HLink er ikke en motsetning til XLink, men en utvidelse av XLink. Målet med HLink er at det ønskes en generell mekanisme for å identifisere lenker, samtidig som det ønskes full kontroll når syntaksen til språket designes.

2.6.2 Bruk av HLink

HLink elementer defineres bare i roten av dokumentet, dette er enten i definisjonselementet til XHTML dokumentet eller i "head" elementet. Her defineres lenkeelementene som skal være tilgjengelig i resten av dokumentet. I disse elementene identifiseres elementets navn, oppførsel og bruk gjennom attributter definert i HLink navnerommet. På denne måten skreddersys lenkeelementene og kan tas i bruk i resten av dokumentet.

2.6.2.1 Attributter i HLink

HLink introduserer en rekke nye attributter i tillegg til at den benytter flere elementer kjent fra XLink spesifikasjonen. Det er to attributter som må inngå i hver HLink, disse attributtene er namespace som spesifiserer navnerommet som elementet hører til i og element som identifiserer elementets navn. Resten av attributtene kan bli brukt på to måter, enten spesifiseres en bestemt verdi, eller så kan en definere et attributt som kan identifisere denne verdien. Dette gjøres ved å gi attributtet verdien "@" i tillegg til et attributt navn.

Et eksempel på dette:

```
<HLink namespace="http://www.w3.org/1999/xhtml"
  element="home"
  locator="www.miksha.net"/>
```

Brukes nå elementet <home>Miksha</home> i dokumentet vil alltid dette elementet lenke til adressen "www.miksha.net".

```
<HLink namespace="http://www.w3.org/1999/xhtml">
  element="home"
  locator="@href"/>
```

Benyttes dette elementet vil en være avhengig av å gi attributtet href en verdi for å få en lenke. Dette vi da se noe slikt ut i et XHTML-dokument; <home href="www.miksha.net">Miksha</home>.

Andre attributter som kan benyttes i HLink er:

- locator, dette attributtet inneholder URI referansen for lenken.
- effect, spesifiserer effekten av å aktivisere en lenke, dette elementet er likt elementet xlink:show i XLink. I tillegg til verdiene dette kan ha i XLink kan attributtet inneholde verdiene submit (locator benyttes for å forelegge et skjema, enda ikke ferdig utviklet), og map (kildedokumentet lenker til en ressurs som er en lenke, denne beskriver hvordan presentasjonen av lenken i kildedokumentet skal bli tolket av en serie lenker til andre ressurser).
- replacment, dette attributtet spesifiserer en omgivelse som den nye ressursen åpnes i. Denne er avhengig av at effect eksisterer.
- actuate, dette attributtet bestemmer hvordan en lenke skal aktiveres og er lik xlink:actuate i XLink, og target i HTML. Verdiene onLoad (lenken aktiviseres

automatisk) `onRequest` (lenken aktiviseres ved brukerens primære interaksjon med lenken) og `onRequestSecondary` (lenkene blir aktivisert av brukerens sekundær interaksjon med lenken, på denne måten kan en lenke ha to verdier for aktivisering) kan tilordnes dette attributtet. Det er spesielt `onRequestSecondary` som er interessant. Denne kan være nødvendig om en for eksempel har en lenke som automatisk aktiviserer et bilde, det kan da være ønskelig å benytte `onRequestSecondary` for å beskrive funksjonalitet om lenken blir klikket på.

- `role`, gir informasjon til brukeragenten om hvilken rolle lenken har. Denne kan ha samme verdier som `rev` og `rel` i HTML lenking.
- `mediatype`, benyttes for å spesifisere mediatypen til ressursen som det lenkes til. En brukeragent kan bruke dette til å bestemme om den kan/ønsker å traversere denne lenken.
- `onSuccess`, indikerer nødvendig prosessering etter at en lenke suksessfullt har blitt traversert. Denne kan ha to verdier `processChildren` (prosesseringen fortsetter med å prosessere barnenodene til elementet) og `ignoreChildren` (prosesseringen fortsetter ved å ignorere barnenodene til elementet).
- `onFailure`, dette er likt `onSuccess` bare at dette elementet forteller hvordan lenkeelementet skal prosesseres videre når det oppstår en feil i traverseringen. Dette kan være nettverksproblemer, ugyldig lenke, fordi ikke brukeragenten forstår medieformatet og lignende. I tillegg til `processChildren` og `ignoreChildren` kan dette attributtet også ha verdiene `warn` (prosesseringen fortsetter men det blir gjort klart for brukeren at det har oppstått en feil) og `fail` (prosesseringen stoppes og det sendes en feilmelding til brukeren).

I tillegg kan `XLink` attributtene `arcrole`, `from`, `to` og `label` brukes. Attributtene `reverseRole`, `shape` og `coords` er oppført i utbedringen av `XHTML 2.0` uten å enda ha fått et bestemt bruksområde. Gjennom denne måten å definere lenker og attributtene tilgjengelig i `HLink` kan en definere elementer som er helt lik `A` element lenker i `HTML`, men også avanserte `XLink`er kan defineres.

3 Integrering av avanserte lenkestrukturer

De fleste nettlesere i dag støtter ikke eller har liten støtte for extended XLink. Dette er fordi nettleserne ikke inneholder programvare for tolking og håndtering av lenkemodellen. For å kunne ta i bruk avanserte lenkestrukturer på Web, må lenkene bearbeides. Lenkene må utformes slik at de har den funksjonaliteten som er tenkt for dem. Det er derfor viktig å se hvordan disse kan presenteres ved å benytte allerede eksisterende teknologier. På grunn av at ikke de fleste nettlesere ikke støtter XLink, må det også vurderes hvilke mulige arkitektonisk nivå lenkene skal behandles på.

Kapittelet tar for seg forskjellige stilark som kan benyttes for å transformere og knytte stil til lenkene. Teknologier for interaktivt innhold på Web blir vurdert i forhold til dets evne til å muliggjøre XLink funksjonalitet. Det gjøres så rede for de arkitektoniske mulighetene for behandling av XLink.

3.1 Stilark

XML har ingen implisitt visualiseringssemantikk. Dette gjør at elementene knyttes opp mot "stylesheets" (stilark) for å kunne utforme grensesnitt. XML er godt tilpasset GUI utvikling. Dette er fordi dens deklaratve tilnærming gjør at man slipper å tenke på mye prosedural kode. Deklarativ programmering er effektivt når det er høyfrekvent bruk av elementer og mønster, og kan kombineres med skriptspråk for å tillate prosedural kode. XMLs hierarkiske struktur gjør språket velegnet til å utformes i grensesnitt, og for å holde på logiske tilstander i en applikasjon. Ved å separere presentasjon og innhold, kan grafisk utforming enkelt forandres og tilpasses det brukergrensesnittet som benyttes. En annen fordel med dette er at forandringer i stilarket vil påvirke alt innholdet som formateres av stilarket. I forhold til lenker gjør dette at vi får en konsistent utforming av grensesnittet, fordi vi får en standardisert presentasjon av lenkene i Webresursen. Det er flere alternative stilark som kan benyttes ved en XLink-implementasjon, disse kan også kombineres. De fleste ressurser på Web bruker stilark for å definere utforming.

Noen av fordelene ved å benytte stilark:

- Utveksling av informasjon over HTTP vil gå fortere, og dermed gi kortere responstid for brukerne. Dette fordi dokumentets kode blir mindre når stil informasjon trekkes ut, samme stilinformasjon benyttes også ofte på mange sider innen tjeneren, noe som gjør at stilarket er gjenbrukbart.
- Redigering og utforming vil kunne gjøres uten å måtte gjøre inngrep i dokumentet som inneholder informasjonen.
- Stilarket trenger bare å bli lastet ned en gang for hver separerte tjener.
- Fleksibelt, informasjon kan tilpasses andre enheter (mobiltelefon, pda) uten å måtte forandre på informasjonsdokumentet.
- Samme side kan vises på flere måter, for eksempel for skjermvisning, utskriftversjon, taleversjon og blindespråk.
- Ved å separere stil fra informasjonsbærer vil indeksering av dokumenter bli mer effektivt. Dette gjør at det er enklere for søkemotorer å finne innhold.

En annen fordel med stilark er at de kan konvertere dokumenter til nye formater. De fleste nettlesere har begrenset støtte for XML. Det er derfor ofte ønskelig å konvertere dokumentet til HTML eller et annet format før det blir presentert for bruker. XSL, CSS og DSSSL kan benyttes for å gjøre konverteringen til HTML. Det kan også konverteres til andre språk som for eksempel RTF, PDF, Tex og lignende [23]. En slik konvertering kan være smart fordi en ikke trenger å bekymre seg om støtten til brukerens nettleser.

3.1.1 Cascading Style Sheets

CSS [8] er et stilarkspråk som tillater forfattere å knytte stil til strukturelle dokument. Dette er det mest brukte stilarket for dokumenter på Web. Før CSS var all informasjon for presentasjon inkludert i markup-språket. Dette gjorde at det måtte spesifiseres presentasjonsinformasjon for hvert element, noe som førte til mye duplisering av informasjon og dermed unødvendig store dokumenter. CSS lar forfatteren eller brukeren flytte mye av stilinformasjonen til et stilark, noe som gjør informasjonsdokumentene enklere. I CSS skal flere lag med stilark kunne legges oppå hverandre, på denne måten kan for eksempel bakgrunn, tekst, lenker, knapper og lignende defineres i forskjellige dokumenter. Stil som kan spesifiseres i CSS-dokumentet er farger, størrelser, teksttyper

til plasseringer og mer strukturell design. Stilarkene knytter utforming til dokumentene gjennom å referere til elementtaggene.

CSS informasjon kan tilføres dokumentene på flere måter. Forfatteren kan gi dokumentene stil gjennom å referere til et eksternt CSS dokument, integrere blokker av stilinformasjon i stilarket, eller å definere stil for hvert element i HTML-dokumentet ved å bruke stilattributter i markup-elementene. Det er også mulig for brukeren å definere stil til eksterne dokumenter. Dette kan gjøres ved at brukeren lagrer en CSS-fil lokalt som nettleseren benytter istedenfor den stilinformasjonen som defineres for dokumentet. Brukeragenten kan ha ferdig definert stilinformasjon som er "default" for presentasjon av enkelte elementer.

Det største problemet med å benytte CSS er "bugs" eller mangel på støtte av nettleser. IE⁹ er den blant dagens nettlesere som har dårligst støtte for CSS, bare ca. 70 prosent av CSS egenskapene støttes av IE 6.0 [43]. Det som er unikt med CSS er at språket kan benyttes til å formatere både HTML og XML-dokumenter. CSS og XSL-FO bruker de samme underliggende formateringsmodellene, designere har derfor tilgang til de samme formateringsmuligheter i begge språk.

3.1.2 Extensible stylesheet language (XSL)

XSL er en familie med språk som beskriver hvordan XML-dokumenter skal bli transformert eller formatert. XSL består av tre språk, disse er "XSL Transformations (XSLT), XSL-FO¹⁰ og XPath¹¹.

XML Path Language

XPath er et fjerdegenerasjons deklarativt språk for å lokalisere noder i XML-dokument. Med XPath kan spesifikke deler og fragmenter av et XML-dokument adresseres. Navnet XPath kommer av at den bruker "path" notasjon for å navigere gjennom den hierarkiske strukturen til et XML-dokument. XPath gir også muligheten til å manipulere strenger, nummer og boolske verdier. XPath modellerer et XML-dokument som et tre av noder.

⁹ Microsoft windows, Internet Explorer 6.0, www.microsoft.com/windows/ie/ie6/default.msp

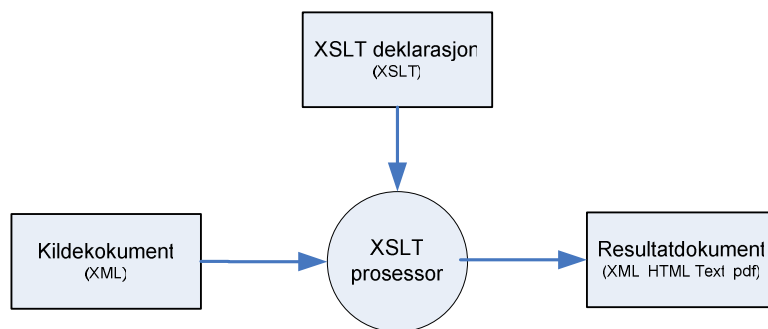
¹⁰ XSL Formatting Objects

¹¹ XML Path Language

Det finnes forskjellige typer noder, disse er elementnoder, attributtnoder og tekstnoder [15]. XPath er blitt adoptert av utviklere som et spørrespråk innenfor XML. XQuery og XPointer er begge bygd på XPath uttrykk.

XSL Transformations

XSL Transformation (XSLT) som er et språk for å transformere eller omstrukturere et XML-dokument. Denne transformeringen gjøres av en dedikert XSLT prosessor [27]. En slik prosessor tar imot et XML-dokument og transformerer dette etter regler spesifisert i en XSLT deklarasjon. Resultatet av transformasjonen kan være et nytt XML-dokument, eller et annet format som HTML, tekstdokument eller andre format som prosessoren er i stand til å produsere. Den mest vanlige bruken av XSLT er for å transformere dokumentene til HTML eller PDF-dokument.



Figur 5 XSLT transformasjon

XSLT er et deklarativt språk, bestående av en rekke template-regler¹² som spesifiserer hva som skal legges til resultattreet når XSLT prosessoren går gjennom kildedokumentet. En template består av algoritmer som finner noder i kildedokumentet ved hjelp av å assosiere dets mønster med det som er beskrevet gjennom XPath i XSLT-dokumentet. En template fungerer som en metode. Denne utfører en bestemt handling basert på de reglene beskrevet i den. Gjennom å filtrere og sortere kildedokumentet etter template-regler, kan kildedokumentet omstruktureres, og gi et resultatdokument totalt forskjellig fra kildedokumentet [49].

¹² Template regler inneholder XSLT kode, og kan kalles fra andre stilarer.

XSL Formatting Objects

I motsetning til HTML så har ikke XML-dokumenter noe implisitt format for presentasjon. XSL-FO er et XML vokabular for å spesifisere formateringssemantikk for XML-dokument. XSL-FO bygger på tidligere arbeid gjort innen CSS og DSSSL. Stilarket introduserer en modell for layout som utvider mulighetene for stilark generelt. XSL-FO blir uttrykt i XML noe som gjør det fleksibelt og utvidbart. Vokabularet består av ett sett med FO (Formatting objects) elementtyper som representerer de typografiske abstraksjonene som er tilgjengelig for designeren. Formateringsobjektene er delt i klasser som representerer en spesifikk del stilinformasjon.

XSL ble utviklet for å gi designere kontroll over stil elementene når dokument blir delt opp i sider i likhet med ramme basert struktur som når man ”browser” på veven. I forhold til CSS er XSL-FO mektigere når det kommer til å definere ”scrolling” og sideoppdelinger. Ved at XSL-FO blir uttrykt som er ”well formed” XML er det enklere å integrere det i XSLT eller direkte i XML-dokumentene om dette er ønskelig.

3.1.3 DSSSL

Dokument Style Semantics and Specification Language (DSSSL) er et internasjonalt stilark språk for SGML-dokumenter. Dette stilarket benyttes for å presentere og transformere informasjon som er skrevet i SGML. Med DSSSL kan SGML konverteres til formater som HTML, RTF og LaTeX. Dette språket benyttes spesielt for å formatere dokument før utskrift og er brukt av DocBook (markup-språk for teknisk dokumentasjon) [20]. Ved overgangen fra SGML til XML har utviklere gått over til å benytte XSL i stedet for DSSSL.

3.1.4 XML User Interface Language

XUL er en applikasjon av XML som brukes til å beskrive innhold i nettleservindu og dialoger. Dette er et brukergrensesnittspråk som er laget for å støtte Mozilla applikasjoner[20]. XUL gjenbraker eksisterende teknologier, som CSS, JavaScript, DTD og RDF. XUL benyttes blant annet i Netscape 7.0 for å konstruere verktøylinjer, knapper, menyer og dialoger. Språket har blitt standardisert av Mozilla og kjøres på

samme "layout engine" som behandler HTML-innhold i en nettleser. Dette er et markuspråk for XML med en spesifikk mening definert for hver elementtype. Brukergrensesnittbeskrivelsene er i stor grad lik de i HTML 4. Å forfatte grensesnittet til et vindu i XUL er veldig likt det å skrive en HTML beskrivelse. Forskjellene er at syntaksen som benyttes er XML, og noen unike elementer i XUL. En XUL-fil kan bestå av XML-elementer og HTML-elementer og unike XUL-elementer [29]. Lenking i XUL gjøres ved å benytte HTMLs A-element.

3.2 Programmeringsverktøy

For å kunne presentere extended XLink på i en Web ressurs, vil det være hensiktsmessig med en transformering til et format som støttes på Web. Dette kan gjøres ved bruk av et stilark som XSLT. På grunn av strukturen og den rike semantikken til XLink vil det kreve en bredere presentasjonsteknikk enn det som kan tilbys gjennom HTML. For å få dette til må en benytte programmeringsverktøy som muliggjør aktivt innhold i Web-dokumenter. Dette kan spesielt være nødvendig i forbindelse ved presentasjon av multidireksjonale lenker, metadata, lenker fra lenkebasen, dynamiske lenker og annen funksjonalitet som blir gjort tilgjengelig gjennom avanserte lenkemoduler.

3.2.1 JavaScript, JScript og ECMAScript

JavaScript er et objektbasert skriptspråk basert på gjenbruk. Språket benyttes mye på Web for å muliggjøre dynamisk HTML (DHTML). Syntaksen til språket ligner noe på C, C++ og Java, men er mer begrenset. Integret i nettleser kobler JavaScript seg opp til applikasjoner gjennom DOM¹³ grensesnitt, spesielt når det gjelder oppkobling til applikasjoner på tjenersiden og på klientsiden til Web-applikasjoner. En av de store fordelene ved å benytte nettbasert JavaScript, er at en kan skrive funksjoner som er integrert i HTML-dokumentet. Denne kan kommunisere med DOM for å utføre handlinger som ellers ikke hadde vært mulig med statisk HTML.

JavaScript kan ta imot verdier fra brukeren, åpne nye vinduer, forandre bilder og lignende. Dette gjør HTML mer dynamisk og åpner for større mulighet for

¹³ Document Object Modell

kommunikasjon mellom bruker og dokument. Et problem med JavaScript er at DOM grensesnittet varierer over de forskjellige plattformene, dette gjør at forskjellige nettlesere presenterer forskjellige objekter og metoder til skriptet. Det er ingen annen logisk løsning på dette enn å forfatte forskjellige skript til de forskjellige nettlesere. JavaScript er utviklet og implementert av Netscape og er registrert varemerke under Sun Microsystems. Grunnet suksessen til JavaScript utviklet Microsoft JScript som er kompatibel med JavaScript. ECMAScript er navnet på dette språket standardisert i ECMA-262, dette er både JavaScript og JScript kompatibel.

JavaScript er spesielt interessant å benytte når en skal implementere extended XLink funksjonalitet på Web. Istedenfor å benytte HTML, CSS og events kan dynamikk skapes i HTML med JavaScript og dens metoder og objekter. Ved å benytte JavaScript kan man integrere XLink funksjonalitet i et vanlig HTML-dokument. Dette kan være hensiktsmessig da de fleste nettlesere i dag støtter JavaScript, JScript og ECMAScript.

E4X [16] er en utvidelse av ECMAScript og støtter bruk av XML data uten å måtte bruke XSLT eller DOM. Dette for å gjøre bruk av XML-data enklere å lære, og mer intuitivt for ECMAScript programmerere. Målet med E4X er at de som kan programmere i EMAScript skal kunne bruke XML data med lite eller ingen ekstra kunnskap [16]. Det er hittil bare Mozilla Firefox og nettlesere som er basert på Gecko som kan brukes for å kjøre scripts lagd i E4X.

3.2.2 JavaApplet

En JavaApplet er et program skrevet i Java. Disse programmene kalles Applets. En Applet kan kjøre i en nettleser ved hjelp av Java Virtual Machine (JVM). Applets benyttes i likhet med JavaScript for å muliggjøre interaktiv funksjonalitet i en Webressurs. Koden fra en Applet blir lastet ned som separerte filer til nettleseren parallelt til HTML-dokumentet. Resultatet av dette kan virke som om Appleten er integrert i HTML-dokumentet, men Appleten har en gitt posisjon i nettleseren. Siden Javas bytekode er plattformuavhengig kan applets kjøre på mange forskjellige plattformer.

En Applet for å behandle XLink vil kunne produseres og legges inn der det er ønskelig å benytte avanserte lenkestrukturer. Problemet ved å benytte en JavaApplet for å behandle og framstille XLink er at Appleten alltid vil ha en fast posisjon i nettleseren, og vil derfor ikke framstå som en integrert del av HTML-dokumentet. Dette kan føre til designkonflikter, fordi ikke bare plasseringen men også størrelsen, og utformelsen av en Applet vil være statisk. Designeren må derfor ha tilgang på koden til Appleten for å kunne forandre på dens utseende. Dette krever at brukeren må ha god teknisk innsikt, noe som vil minske brukervennligheten på en slik implementasjon. Dette programmet må også lastes inn for hver instans av lenker i Web-dokumentet, noe som fører til uønsket kompleksitet.

Kompatibilitet har vært et problem når JavaApplets skal benyttes på Web. Internet Explorer brukte å transformere Java plug-in slik at den benyttet Microsofts egen JVM som i mange tilfeller ikke var kompatibel til å benyttes på Suns JVM. Noen nettlesere, blant annet Firefox, har problemer med å takle høyden til en applet. Dette resulterer i at appleten ikke fyller hele vinduet i nettleseren.

3.2.3 Flash

Macromedia Flash var opprinnelig et rent animasjonsprogram men har i seinere versjoner blitt utvidet slik at det også inneholder skriptspråket, ActionScript. ActionScript har lik syntaks med JavaScript, men et helt annet programmeringsrammeverk og klassebibliotek. Skriptspråket gjør Flash interaktivt og dynamisk. Dette muliggjør kommunikasjon med brukeren, og det er muligheter for å lage dynamiske nettsider. Flash kan integreres i HTML, men mange velger å lage nettsider ved å bare benytte Flash. Flash kompiles til filformatet ".swf" som er interaktive filmsnutter som er avhengig av Macromedia Flash Player for å håndteres. Dette er en et tilleggsprogram til nettleser som er tilgjengelig gratis.

I likhet til JavaApplets har Flash en bestemt plassering når den integreres i HTML. Dette gjør at plasseringen til Flash-filmen vil være statisk i HTML-dokumentet. Siden Flash også kan eksistere uavhengig av HTML, er en ikke avhengig av å integrere sidene i

HTML for å vise de i en nettleser. En lenke skrives i Flash ved å benytte kommandoen `”getURL (“www.url.com”, “_blank”); ”`. Flash er dynamisk og kan derfor benytte dynamiske trestrukturer kan en skrive en-til-mange lenker i språket. Siden Flash kommer med et programmeringsspråk som muliggjør dynamikk, er også mulig å gjøre funksjonalitet som for eksempel tooltip, eller dynamiske lenker. Et annet problem som introduseres ved bruk av Flash er at ingen stilark kan transformere informasjon til dette formatet. Det er derfor ikke mulig å automatisk transformere en XLink til Flash. Lenker i Flash kan til stor grad være avanserte, men de må skrives og brukes eksplisitt i Flash.

3.2.4 Sammenligning

XLink introduserer nye designutfordringer for ressurser på Web. Dette fordi dens avanserte struktur og rike semantikk vil kreve at mer informasjon må presenteres i grensesnittet. Ved å utvide HTML-dokument med programmeringsverktøy kan en gjøre webressursen interaktiv, og dermed gi større muligheter for å organisere XLinken i et grensesnitt. Programmeringsverktøyene JavaScript, JavaApplets og Flash har muligheter til å lage fleksible grensesnitt og støttes i bruk av de fleste av dagens nettlesere. Språkene muliggjør trestrukturer på Web gjennom å dynamisere HTML-lister. Dette er spesielt interessant når lenker med mange endepunkt skal integreres i HTML.

Den store forskjellen på disse språkene er måten de kan integreres i Web-dokumenter. Mens JavaScript kan integreres direkte i koden til HTML-dokumentet, så må JavaApplets og Flash integreres på en bestemt plass i nettleseren eller presenteres separat som en egen uavhengig applikasjon. Flash er avhengig av at innhold er integrert i selve filmen, noe som gjør den uegnet til å styre grensesnitt for HTML, XHTML eller XML. JavaApplets og JavaScript kan brukes for å gi HTML nødvendig dynamikk, og derfor styre en avansert lenkemodul som XLink. Det er viktig å forstå at JavaScript ikke lager enkeltstående applikasjoner, slik som en JavaApplet, men er et skriptspråk lagd som et supplement til HTML. JavaScript har også et mye enklere og mindre vokabular enn det som er mulig å presentere ved bruk av JavaApplets, men kan tilby tilfredsstillende funksjonalitet for å gjøre HTML dynamisk.

Tabell 2 "DHTML verktøy" identifiserer bruksområdet til JavaScript, JavaApplet og Flash. Denne er kategorisert etter følgende begrep; "dynamisk", som forklarer språkets dynamikk under kjøring, "interaktiv", som er komponentens emne til å kommunisere med bruker, "autonom", som er komponentens mulighet til å opptre som en selvstendig ressurs, "integring i markup – språk", som er muligheten til å skrive kode direkte i hypertekst – dokumentet, "komponent kommunikasjon", som er språkets mulighet å motta og sende data fra/til en webside og "direkte kompilering", som omhandler språkets mulighet til å kompileres "just in time" før kjøring.

	JavaScript	JavaApplet	Flash
Dynamisk	X	X	X
Interaktiv	X	X	X
Autonom		X	X
Integrering i markup-språk	X		
Komponent kommunikasjon	X	X	
Direkte kompilering	X		

Tabell 2 DHTML verktøy

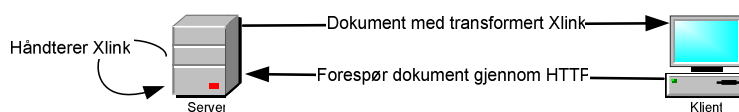
3.3 Arkitektur

Nettleserne som benyttes i dag har ikke utvidet støtte for avanserte lenkemekanismer som XLink. Dette gjør at en applikasjon må håndtere XLinken før den blir visst til brukeren. En applikasjon som tolker XLink må derfor integreres på et arkitektonisk nivå i dokumentflyten. Denne kan integreres på fire forskjellige måter;

- Tjenerløsning; transformeringen av XLink foregår på tjeneren. XLinken transformeres før dokumentet blir sendt til nettleseren.
- Proxyløsning; applikasjon som transformerer XLinken blir lagt på proxyen, vi får dermed en "just in time" transformering før dokumentet mottas av nettleser.
- Integrert i nettleseren; en applikasjon integrert med nettleseren håndterer dokument med en XLink.
- Plug-in løsning; løsningen krever at en installerer en applikasjon som en utvidelse av nettleseren. Dette gjør at klienten kan håndtere avanserte lenker.

3.3.1 Tjenerløsning

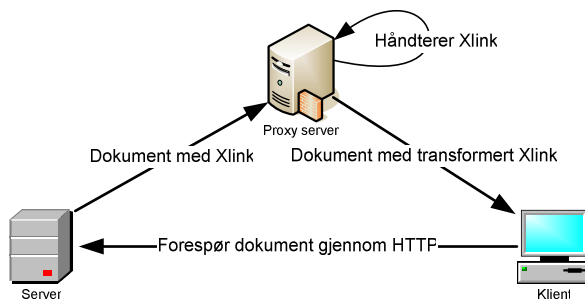
Plasseres applikasjonen for transformering av XLink på tjener vil lenkene bli transformert og sammenflettet i dokumentet før de sendes til nettleser. Ved en dokumentforespørsel fra klient vil tjeneren sende lenkene i et format som støttes av klienten. På denne måten vil ikke nettleseren være avhengig av å kunne tolke den strukturen som benyttes for å beskrive dokument. Fordelen med dette er at ingen ekstra applikasjon må installeres eller tas i bruk for å kunne benytte seg av avansert lenke funksjonalitet.



Figur 6 Applikasjon for transformering på tjener nivå

3.3.2 Proxy løsninger

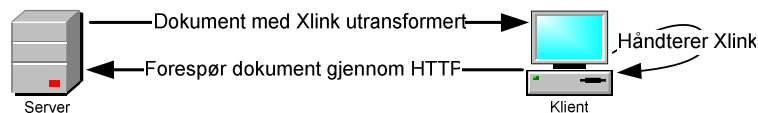
Ved å modifisere HTML-dokumentene i proxyen kan en integrere avansert XLink funksjonalitet i dokumenter rett før de motas av nettleser [3]. Normalt benyttes HTTP-proxyen til å kontrollere tilgang til Web og "caching", men det kan også brukes til å legge til informasjon i dokumenter når de går gjennom proxyen. Plasseres applikasjonen som håndterer de avanserte lenkene i dette nivået, slipper brukeren å installere egen programvare for visning av lenkene. Tjeneren trenger heller ikke å kunne håndtere XLinken, noe som gjør at dokumentet som blir forespurt kan inneholde XLinker, og bli visst på Web plattformer uten utvidelse av nettleser. XLinkProxy[12] og Goate[4] benytter denne løsning for å integrere avansert hypermediefunksjonalitet på veven.



Figur 7 Applikasjon for transformering på Proxy nivå

3.3.3 Klientløsning

Med en klientløsning, menes det at nettleseren inneholder en innebygd løsning for å kjenne igjen og utforme XLink i grensesnittet. Da vil en applikasjon for håndtering av lenkestrukturen følge med nettlesere i det de kommer fra leverandøren. Forfattere av XLink kan abstrahere bort presentasjonen av lenken, fordi nettleserne har bestemte metoder for utforming av lenkene, på samme måte som nettleser har en forhåndsdefinert metode for visning av A-lenker i HTML. Det er fremdeles en applikasjon som håndterer lenkene, men denne er nå innebygd i nettleseren.



Figur 8 Applikasjon for transformering på klient nivå

3.3.4 Plug-in løsning

Dette er en løsning som er basert på en installasjon av en tredjeparts komponent. Applikasjonen installeres på et klient nivå, som en tilleggskomponent til nettleseren. Ved å utvide nettleseren med en applikasjon som kan tolke XLink, kan klienten tolke XLink når dokumentet mottas fra tjeneren. I dag er det lite støtte av XLink blant de mest brukte nettlesere, det må derfor lages en applikasjon som kan håndtere funksjonaliteten som tilbys gjennom avanserte lenker. Dette krever at brukeren selv må installere den programvaren som er nødvendig. En slik løsning vil muliggjøre forskjellige applikasjoner

å definere hvordan XLink skal presenteres. Vi får da et mangfold av applikasjoner som produserer forskjellige resultat. Brukerne vil da få større mulighet til å velge hvilken presentasjon som er ønsket.

3.3.5 Tjener Vs klient

Det presenteres fire alternativer for hvilke nivå en slik transformasjon kan foregå på. Disse nivåene kan i hovedsak deles i to, tjenerside og klientside. Tjener og proxy-løsningene er på tjernivå mens klient og plug-in løsningene er på et klientnivå. Gjøres dette på tjenerside er dokumentene transformert til et format som støttes av nettleseren før den når klienten. En slik implementasjon er derfor plattformuavhengig. En klientløsning avhenger av at brukeren har kompatibel programvare til å håndtere og tolke XLinken.

Ved å legge applikasjon til klient kan en risikere at hver applikasjon for lenkehåndteringen kan tolke og utforme lenkene forskjellig. Det vil dermed bli store variasjoner i lenkenes funksjonalitet og utforming. På klientsiden overlates mer av kontrollen til brukerne, de kan selv bestemme hvilke applikasjon de ønsker og benytte og derfor også ha større kontroll over utforming. En ulempe med dette er at ikke alle brukere vil ha denne applikasjonen installert, dette fører til ekstraarbeid og vil dermed senke brukervennligheten. For å få testet en applikasjon på klientsiden må den testes mot mange plattformer og i samspill med andre plug-ins. Dette kan bli et omfattende arbeid. Det vil derfor være nødvendig med en standardisert måte å håndtere XLink før denne løsningen kan benyttes. Den vil da kunne integreres i nyere versjoner av nettleserne. En implementasjon av XLink på dette nivået vil gi forfatterne en større frihet til å skrive lenker i XLink, uten å måtte ta hensyn til applikasjon liggende på tjener eller proxy for transformering.

Er applikasjonen på tjenersiden vil det ta lengre tid å få returnert forespørselen av dokumentet, og returdokumentet vil da mest sannsynlig være større enn om dokumentet returneres uttransformert. Dette kommer av at en er avhengig av å kjøre to prosesser på tjener, og vil derfor få en lengre "latency", dette kan være minimalt og umerkbart, men ved store systemer som har mange forespørsler avhenger det av hvor godt tjeneren

skalere. Fordelene med å ha en applikasjon som håndterer XLink på tjenernivå er at ingen ekstra applikasjoner eller utvidelser av nettleseren er nødvendig. Dette vil gjøre at så lenge nettleseren støtter det formatet som blir levert fra tjener eller proxy, vil de kunne dra nytte av de avanserte lenkene. Ved å implementere lenkene på et makronivå vil de skalere bedre, og være anvendbar for et større antall brukere. Lenkene vil også ha en lik utforming for hver plattform.

4 State of the art

Siden XLink spesifikasjonen ble innført i 2001 er det gjort lite arbeid med XLink. På oversiktssidene til XML lenke implementasjoner [14] er det registrert syv applikasjoner som benytter XLink, bare tre av disse har delvis implementert avansert XLink. Dette kapittelet tar for seg tidligere arbeid gjort med XLink. Det ser på anvendelse av XLink og XPointer ved alle de arkitektoniske nivåene. Første delen av kapittelet vil ta for seg implementasjoner på klientnivå, presentert under nettlesere. I andre del vil det bli sett på applikasjoner på proxy og tjernivå. Disse er delt opp i applikasjoner som benytter tolker og benytter XLink, og applikasjoner som transformerer XLink til HTML. Implementasjonene som er mest relevant for avhandlingen vil bli vektlagt i dette kapittelet, dette er spesielt implementasjoner av extended XLink med mange endepunkter.

4.1 Nettlesere

XLink har ingen implisitt utforming i nettlesere. Nettleserne må derfor ha en dedikert applikasjon for håndtering av XLink. En slik applikasjon vil benyttes stilark for å tilføre informasjon om visning til dokumentet. Dette kan gjøres med de stilarkene som er diskutert i kapittel 3, eller ved å transformere dokumentet til et format som nettleseren støtter. Verken nyere versjoner av Internet Explorer eller Opera har støtte for bruk av XLink eller XPointer. I artikkelen ”on display: XML Web pages with Opera 4.0” [36] redegjøres det for muligheten til udokumentert triksing i CSS, for å få Opera til å støtte simple XLink.

4.1.1 Amaya

Amaya nettleseren/editoren har mulighet for å lage og aktivere simple XLink. Denne nettleseren bruker XLink når det skal lages lenker til XML elementer. Lenkene blir lagt til som et attributt til elementet som presenterer startpunktet til lenken. XLink kan også benyttes til og fra XML dokument [22]. Amaya har også en veldig begrenset støtte for bruk av XPointer. XPointer benyttes i Amaya for å kunne gjøre RDF annoteringer i strukturerte Web-dokument.

4.1.2 Mozilla

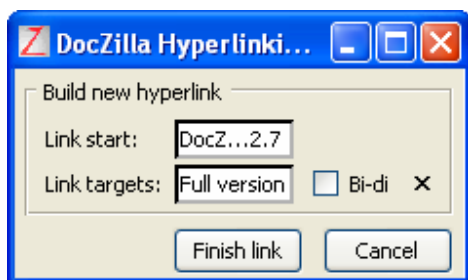
Nettleserne fra Mozilla har støttet simple XLink siden Mozilla 1.0. Disse lenkene støtter også attributtene xlink:show med verdiene "new" og "replace" og xlink:actuate med verdiene "onLoad" og "onRequest". Siden Mozilla 1.4 alpha har nettleserne fått en utvidbar prosessor for bred støtte av XPointer.[28]

4.1.3 Doczilla

Doczilla er produsert av finske CiTEC¹⁴ og er basert på en versjon av Mozilla 1.7.8, og kjører på samme "motor" som Firefox 1.5. Denne nettleseren har støtte både for simple og extended XLink. På grunn av at den er basert på Mozilla, har den også samme støtte for simple XLink. Det som er interessant med denne nettleseren er at det i en viss grad er implementert støtte for extended XLink. Nettleseren benytter CSS, DOM og JavaScript for å vise og manipulere XLink.

Tredjeparts lenking i Doczilla

I Doczilla har en mulighet til å legge til lenker i dokumenter der en ikke har skriverettigheter. XPointer og extended XLink muliggjør lenker i alle dokument som er "well formed". Dette er dokumenter av typen SGML eller XML. Lenkene blir lagret på en tjener og lagt til dokumentet før det sendes til nettleseren. Brukeren kan spesifisere et startpunkt og et endepunkt for en lenke ved å kommunisere med et grensesnitt som blir tilbudt i nettleseren. Denne aktiviseres ved å markere en tekstdel og spesifisere om dette er start eller sluttpunktet til lenken.



Figur 9 Grensesnitt for tredje parts lenker

Lenkene blir markert i nettleseren med en gul tekstboks med skriften "link". Denne legges inn foran ankeret det lenkes fra. Tittelen til endepunktet blir bare vist ved bruk av

¹⁴ CiTEC, www.citec.fi/

”tooltip”. Det er mulig å lage en-til-mange relasjoner. Dette gjøres ved å velge flere endepunkter for hvert startpunkt. Dette løses i praksis med å opprette flere startanker med forskjellige endepunkter. Det er ikke mulig å beskrive mange-til-en eller mange-til-mange relasjoner.

installed [LINKLINKLINKLINK](#) DocZilla 2.7
 ides startk points of where to naviga
 attached [Full version](#) e above!

Figur 10 Lenke fra ankeret "Doczilla 2.7" til fire forskjellige endepunkt

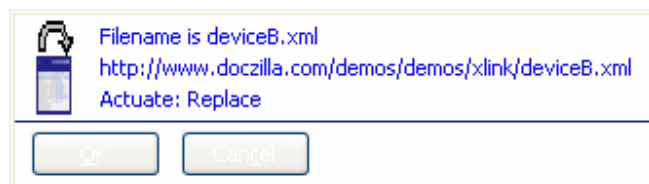
Lenkens endepunkt blir markert med blå farge, slik at brukeren kan se hva det ble lenket til. Det er også mulig å velge om lenken skal være bi-direksjonell. Det opprettes da et lenkeanker fra endepunktet tilbake til startpunkt.

[LINK](#) Full version
 We no longer distribute a public version and a commercial, full version of the browser. There's one package only, all features are in it, and it's free of char

Figur 11 Endepunkt for en bi-direksjonell lenke i Doczilla

XLink integrert i dokumentet

Doczilla presenterer funksjonalitet for bruk av XLink skrevet av forfatter. Dette er støtte for bruk av xlink:show og xlink:actuate, funksjonalitet for tooltip ved bruk av xlink:title element og for å beskrive lenker i lenkedatabase for deretter å legge dem inn i dokumentet via XPointer. Demonstrasjonene til bruk av XLink i Doczilla, presenterer forslag til en andre måter for traversering av en-til-mange relasjoner enn den presentert ved tredjeparts lenking. Denne metoden aktiviserer en meldingsdialog som informerer brukeren om alternativ lenke når brukeren skal traversere en lenke. Bruker kan da velge å traversere denne eller å gå til det dokumentet som var utgangspunktet for lenken, ved å velge ”ok” eller ”cancel”.



Figur 12 Meldingsdialog for lenker med flere endepunkt i Doczilla

4.2 Applikasjoner for avanserte lenker

Dette er applikasjoner som kan brukes som plug-in for nettleser, eller på tjenernivå for å håndtere XLink. Applikasjonene benytter XLink uten å transformere de til et annet format. Disse er derfor avhengig av at dokumentene de benyttes i er XML eller XHTML.

4.2.1 Xlip

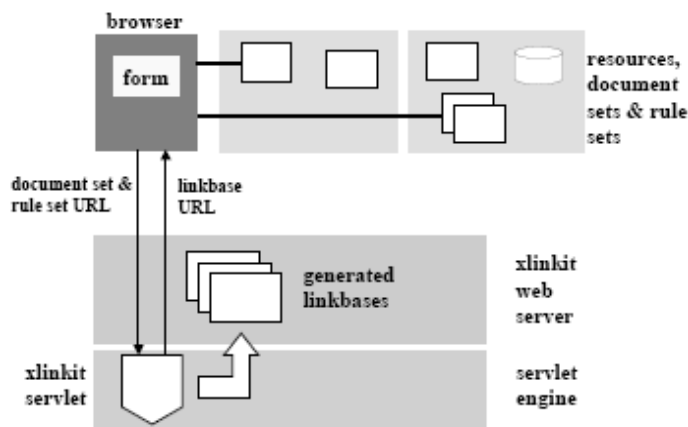
Xlip er en prosessor fra Fujitsu for håndtering av simple og extended XLink. Prosessoren har støtte for bruk av locator, resource og arc element i extended XLink. Andre enheter kommuniserer med Xlip gjennom DOM. Traverseringer fra en-til-mange noder gjøres gjennom bruk av XLink extended med flere endepunkter. Når en bruker aktiviserer et lenkeanker, vil han/hun få presentert en dialogboks som presenterer de alternative endepunktene. Brukeren kan så velge mulig den lenken som det er ønskelig å traversere. Slike lenker kan også legges inn av en bruker, og lagres separert fra dokumentet på en spesifisert tjener. Lenkene fra en slik lenkedatabase legges inn i dokumentet etter verdi spesifisert av xlink:actuate attributtet. Xlip har en begrenset støtte for XPointer. XPointer benyttes her bare for å legge til lenkene, ved tredjeparts lenking.

Xlip er ikke tilgjengelig gjennom Fujitsu, det var tidligere lagt ut en demonstrasjon av prosessoren som nå er fjernet fra Fujitsu-sidene, og det er derfor vanskelig å få testet dens funksjonalitet. All dokumentasjon om prosessoren er skrevet på Japansk, og det er ingen svar å få på email. Den eneste Engelske dokumentasjon for Xlip finnes gjennom W3C utspørring om Xlip i forhold til XLink og XPointer [26].

4.2.2 XLinkit og XTooX

Applikasjonen XLinkit benyttes for å evaluere XML-dokument opp mot et sett av regler for XLink lenker. Dette gjøres for å identifisere konsistente og ukonsistente relasjoner. XLinkit består av et Web grensesnitt og en database som vedlikeholder lenker fra bruker. Applikasjonen fungerer ved at bruker gjennom Web-grensesnittet forteller XLinkit hvilken informasjon som det ønskes å lenke sammen, og hvilke regler som skal anvendes. Applikasjonen genererer så en lenke som benyttes i kildedokumentet. Denne lenken, lenker til en database som tar brukeren videre til lenkens endepunkt. Hensikten

med dette er å eliminere arbeid for forfatteren i forhold til å skrive og vedlikeholde lenker. Lenkene har ikke noen XLink semantikk, men applikasjonen kalles XLinkit på grunn av database tilnærmingen til integrering av lenker [30].



Figur 13 Arkitektur for XLinkit

Utviklerne av XLinkit har også utviklet en extended XLink prosessor kalt XTooX som integrerer tredje parts extended XLink i XML-dokumenter. Denne applikasjonen leser dokument bestående av extended XLink og integrere dem i de dokumentene som de identifiseres som deres kilde gjennom arc elementene.

Hjemmesidene hvor XLinkit og XTOOX, www.xlinkit.com er fjernet, og det er dermed ingen tilgjengelig demo til testing.

4.2.3 X2X

X2X er en tjener som genererer lenkestrukturer mellom informasjonsressurser som er eksterne og skrivebeskyttet. Lenkene lagres i en JDBC-database som Oracle eller SQL-server og blir dynamisk lagt inn i dokumentene på forespørsel. X2X kan tolke og håndtere lenker med XLink syntaksen som er definert av W3C [18]. Systemet er lisensiert, og blir derfor ikke testet ytterligere

4.3 XLink-transformerere

XLink-transformerere er en applikasjon som gjør XLink om til et annet format. Disse systemene muliggjør bruk av lenkestrukturer som XLink i ved å transformere lenkene til HTML og deretter vise de i nettleser. Dette gjøres slik at en kan være sikker på at lenkene kan benyttes av de fleste nettlesere.

4.3.1 Goate

Goate er et system som emulerer avansert lenking i HTML ved hjelp av HTTP proxy. Dette systemet muliggjør bruk av avansert lenking i ordinære nettlesere. Systemet er utviklet av Martin Duncan, Mark Truran og Helen Ashman fra universitetet i Nottingham i 2002/2003.

Den avanserte lenkestrukturen til XLink simuleres ved hjelp av å benytte A og img i HTML, CSS og JavaScript. Løsningen er basert på prinsippet om at avansert lenking kan presenteres ved hjelp av ett stilspråk og et lavnivå lenkespråk. På denne måten presenteres bi-direksjonelle lenker som to enveislenker som peker til hverandre og flerveislenker som flere enveis lenker som peker fra samme kilde. For å få tilgang til dokumenter som har skriverestriksjon benyttes HTTP-proxyen, for så å forandre dokumentet. Normalt benyttes HTTP-proxyen for å kontrollere tilgang til Web og "caching", men den kan også benyttes til å modifisere dokumenter. Dokumentet blir så sendt til klienten som et vanlig HTML-dokument med den nye funksjonaliteten integrert i dokumentet. CSS og JavaScript benyttes for å vise lenkene i nettleseren.

Teknisk

Proxyen kjører ikke som en enkel prosess, men ved oppstart startes et n antall kopier. Alle disse barneprosessene venter på en kobling. Hver barneprosess kobler seg opp til Goate PostgreSQL database der det er lagret detaljer om lenker. Når det kobles opp mot Goate blir meldingen "parsed" og sendt videre til tjeneren. Dokumenter som ikke inneholder HTML blir sendt direkte til klienten, mens HTML-dokument blir prosessert på følgende måte;

Goate finner ut hvilken nettleser som benyttes, og informasjon om denne blir lagret til senere bruk. HTML-dokumentet blir så "parsed" til et lokalt "XMLDoc" format. Et slikt dokument består av et nummer med XMLItems hvor ett XMLItem er enten tekst, kommentar eller et element. I motsetning til de fleste XML parserne så tolererer Goate feil som ugyldige attributtverdier, feilskrevet tagger og lignende. Slike syntaksfeil blir rettet i parseren, ved å legge til eller slette tagger.

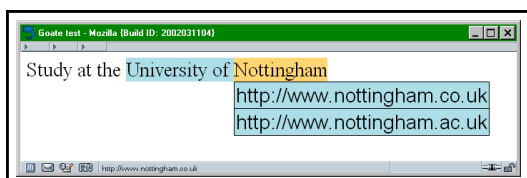
Lenkene i dokumentet blir så transformert til Goate format. Hver språkmodul skanner dokumentet som nå er XMLdoc for å finne de elementene som blir gjenkjent som lenker. Når språkmodulen finner en lenke er den ansvarlig for å evaluere lenken og oversette den til følgende form; destinasjon URI, starten på destinasjonen, slutten på destinasjonen og direksjon. Start- og ende-destinasjonen refererer til posisjonen til lenken innen måldokumentet. Direksjonsattributtet inneholder informasjon om lenken er en enveis eller flerveislenke. Når språkmodulen er ferdig med å evaluere en lenke kalles API-funksjonen med detaljene. I de tilfellene der en lenke har mange destinasjoner så repeteres kallene til APIen. Elementet som inneholder denne lenken setter et flagg for å vise at lenken er suksessfullt prosessert. Når språkmodulene har gått igjennom dokumentet slettes de elementene som viser seg å være suksessfullt prosesserte lenker. [4]

For å få lagt til de lenkene som skal legges inn i dokumentet gjøres det en SELECT-spørring i lenkedatabasen. Lenkene blir integrert i XML-dokumentet som Link XMLItem. Siste fasen av prosessen er å skanne dokumentet og konvertere lenke XMLItems til HTML-kode. Koden avhenger av type nettleser som benyttes. Lenkene blir presentert som A-elementer.

Presentasjon

Utgangspunktet for presentasjon av lenker i grensesnitt er basert på anbefalingene fra "The look of the link" [35]. Det benyttes forskjellige farger for å skille lenker med flere endepunkter fra enveislenker, og utgående lenker fra tilbakegåendelenker. Fargene legges i bakgrunn av lenkeankeret. Blå benyttes for utgående enveislenker, grå for

tilbakegåendelenker og orange for flerveislenker. Ved å skille lenketyperne fra hverandre visuelt er det enklere for brukeren å forstå lenkene. For å vise lenker med flere endepunkt benyttes en ”pop-up”-liste som aktiviseres når musepekeren traverserer over lenkeankeret. I denne listen benyttes også bakgrunnsfarge for å spesifisere hvilke lenker som er utgående og hvilke som er tilbakegående.



Figur 14 Mozilla screenshot av Goate

Problemet med å benytte et system som integrerer lenker på et proxy-nivå som Goate er at eksterne brukere må selv konfigurere deres nettleser til å benytte Goate proxyen. Dette gjør at hver bruker er avhengig av proxyen for å få vist lenkene i riktig format. Det kan være uheldig om brukeren allerede benytter en annen proxy. Dokumenter som er avhengig av Goate proxyen må inneholde en advarsel om at dokumentet må kjøres gjennom en Goate proxy for at det skal få tenkt utforming. Det er også mulig å løse dette problemet ved å videresende brukeren til en URI som automatisk kobler opp mot proxyen [3]. Presentasjonsmessig baserer Goate seg på bruk av mange farger. Dette kan virke mot sin hensikt da slik fargebruk er trekker mye oppmerksomhet, spesielt i dokumenter der lesbarhet er en viktig faktor.

4.3.2 XLink2HTML

XLink2HTML 1.0 er ett gjenbrukbart XSLT stylesheet for å representere multidireksjonale XLink-elementer i HTML. XLink2HTML består av XSLT og JavaScript som transformerer en XLink simple og extended til HTML lenker, dette kan benyttes til å presentere en-til-mange og mange-til-mange relasjoner mellom dokumenter i HTML[2]. XSLT har ikke støtte for tredjeparts lenker og lenkedatabaser, og benytter derfor ikke XPointer. For å benytte denne applikasjonen krever det at forfatter har rettighetene til de dokumentene som XLink skal implementeres i. XLink2HTML består av et XSLT dokument med integrert JavaScript.

Teknisk

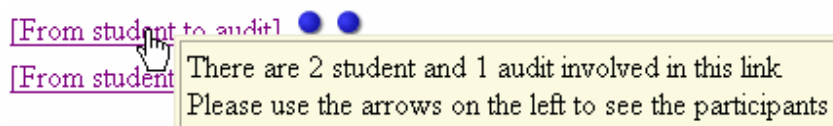
XLink2HTML tilbyr et XSLT stilark med et sett templates som tar seg av transformering av forskjellige XLink lenketyper. Templatene må kalles fra en eget stilark som må forfattes av bruker. Dette stilarket skal kalle templatene fra XLink2HTML når XLink skal transformeres til HTML. XLink2HTML inneholder templatene:

- XLink2HTML-extended oppretter først en nodeliste med alle lenkeelementene som brukeren ønsker å gjøre om til HTML. Elementet arc i lenken definerer hvilken relasjon det er mellom lenkene. Det sjekkes så om lenken har, en-til-mange eller mange-til-mange relasjon, og kaller deretter templatene XLink2HTML-one-to-many eller XLink2HTML-many-to-many.
- XLink2HTML-one-to-many templatene kaller XLink2HTML-create-list som går gjennom nodelisten og finner ut om noden er av type locator eller resource. Er det en locator så transformeres denne til HTMLs A element med href verdien fra XLinken, der tittelen til XLinken blir brukt som anker. Er det en resource mappes tittelen fra XLinken til tittel for A elementet.
- XLink2HTML-many-to-many inneholder det logiske innholdet for prosessering av mange-til-mange lenker. Denne templatene lager en liste over kilder og mål og viser i lag når templatene blir kalt. Det første denne metoden gjør er ved å benytte metoden count() telle hvor mange kilder (source) og hvor mange mål (resource) det er. Dette blir vist i et tooltip som forteller hvor mange fra- og til- elementer lenken inneholder.
- XLink2HTML-simple-link-content transformerer XLink av typen simple til HTMLs A lenke
- XLink2HTML-create-list er templatene som lager HTML-innhold av locator og resource elementene til XLink. Disse legges i en nodeliste som behandles av JavaScriptet i dokumentet.

Presentasjon

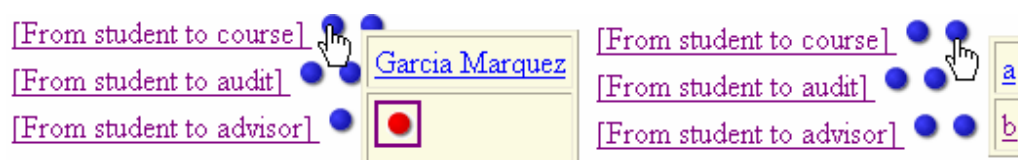
XLink2HTML transformerer XLink til HTML og presenterer denne ved bruk av JavaScript. JavaScriptet er integrert i XLink2HTML-stilarket og kalles av templatene som er beskrevet over. JavaScriptet identifiserer hvilken nettleser som er i bruk, og muliggjør tekstboks for valg av lenkene ved museklikk. Teksten som presenterer

relasjonen vises som en blå understreket lenke. Ved å klikke denne presenteres en tooltip som informerer om hvilke ressurser som inngår i lenken.



Figur 15 XLink presentert ved hjelp av XLink2HTML

Lenkenes kilde og mål blir presentert ved bruk av HTMLs `img` som legger til blå knapper på enden av lenkens anker. Brukeren finner ut hvilke ressurser som hører til hver kategori ved å klikke på knappene som aktiviserer en liste over mål og kilde elementer som inngår i lenken. Den røde knappen i listen er bare til for å lukke listen. Lenken under viser at det er en lenke fra en student med navn "Garcia Marquez" til to kurs, kurs "a" og "b".



Figur 16 Liste over antall studenter som er med i lenken, og til venstre alle kurs

På denne kan det presenteres relasjoner av typen en-til-mange, mange-til-en og mange-til-mange i grensesnittet. XLink simple presenteres som et A element i HTML, med blå font og understrek. XLink2HTML har ikke støtte for attributtene `xlink:show` og `xlink:actuate`, som definerer oppførsel ved en traversering. Ved en traversering vil den nye lenken bli presentert i et nytt vindu, som en følge av en kommando fra bruker (museklikk).

XLink2HTML er en applikasjon som gjør at forfatteren kan integrere XLink i sine HTML dokumenter. Dette gjøres ved at forfatteren selv må skrive et XSLT stilark som kommuniserer med templatene som XLink2HTML tilbyr. Denne applikasjonen vil derfor egne seg for bruk på tjenernivå. Dette kan tenkes i bruk der forfatteren har en XML-database med XLink som relasjonsmodell, og ønsker å presentere denne informasjonen i HTML. Templatene fra XLink2HTML kan da benyttes for å håndtere lenkene i XML-dokumentet. Presentasjonen av lenkene er noe mangelfull, da det er vanskelig å implisitt forstå de blå og røde knappenes funksjonalitet. Disse må klikkes på før brukeren får opp

listene over lenker. Dette gjør at det kreves to klikk før bruker får traversert en lenke. Lenkeankeret foran knappene er misvisende. Disse er utformet som lenker slik vi kjenner dem på Web, men ved klikk aktiviseres et tooltip. Dette tooltipet forklarer traverseringen som kan aktiviseres via de blå knappene.

4.4 Oppsummering

XLink har liten støtte i dagens nettlesere, den eneste nettleseren som har støtte for XLink er Doczilla. Denne foreslår en metode for skrijving og innføring av brukerlenker i XML- og XHTML-dokumenter der bruker ikke har skrivetilgang. Videre er det begrenset støtte for simple XLink i både Mozilla og Opera. Internet Explorer benytter alternativ teknologi for å oppnå det samme resultatet som ved bruk av simple XLink. Av XLink applikasjoner er de fleste ikke tilgjengelig eller lisensiert. Det er ikke funnet noen dokumentasjon på bruk av disse, men siden hjemmeområdene til produktene ikke eksisterer, antas det de ikke er i noe omfattende bruk. Det er imidlertid to løsninger som har fått noe omtale. Dette er løsningene for transformering av XLink til HTML, Goate og XLink2HTML. Felles for alle systemene er at det er stort utviklingspotensial for det presentasjonsmessige av lenkene. Bortsett fra Goate, som har utviklet brukergrensesnittet i forhold til forslag fra artikkelen "*The Look of the Link*" [35], er det ingen av løsningene er spesielt brukervennlig.

5 XLink; design og funksjonalitet på Web

”Along with providing linking data structures, XLink provides a minimal link behavior model; higher-level applications layered on XLink will often specify alternate or more sophisticated rendering and processing treatments.” - World Wide Web Consortium [2002]

Avanserte lenkestrukturer er ikke vanlig i dagens Web-baserte dokument, og en implementasjon av disse vil skape utfordringer til design av grensesnitt i hypertekst. Med XLink introduseres det mer informasjon som må integreres på en oversiktlig måte i grensesnittet. Applikasjonen som håndterer lenkene må utforme et grensesnitt som passer til konteksten som lenken befinner seg i. Mulighetene for utformingen av XLink avhenger i stor grad av strukturen og innholdet til XLinken. Det er derfor også nødvendig å se hva som må være med i en XLink for å få til forskjellig funksjonalitet. Designmessig er det viktig å tilpasse både datastrukturen og utformingen til XLinken i forhold til hvilken kontekst den står i.

5.1 Bruk av XLink på Web

For å benytte XLink i en nettleser må designet og funksjonaliteten være i riktig kontekst i forhold til hvilken type dokument lenkene skal integreres i. To verdier er spesielt viktige når det gjelder visning av lenker i dokumenter; disse er lesbarhet og synlighet. Disse to variablene er i stor grad avhengig av hverandre. Obendorf og Weinreich rapporterer at godt markerte lenker skaper lesbarhetsproblemer, samtidig som brukbarheten til lenkene svekkes når visualiseringen av lenkene blir mangelfull [34]. Er dokumentet ressursbasert med lite tekst, er ikke lesbarhet den viktigste faktoren å ta hensyn til. Skal derimot lenkene integreres i et tekstbasert HTML-dokument, er det viktig å tenke på lesbarhet til dokumentet. Grovt sett kan dokumenter på Web deles i to grupper; ressursbaserte og tekstbaserte dokument. Disse to dokumenttypene har forskjellige bruksområder, innholdet må derfor tilpasses tenkt bruk av dokumenttypen. Det er derfor viktig å identifisere typen og bruksområdet til dokumentet, før lenkens datastruktur og utforming bestemmes.

5.1.1 Ressursbaserte dokumenter

Ressursbaserte dokumenter er dokumenter som ofte inneholder metadata om en ressurs, et produkt, en innstans og lignende. Slike dokumenter finnes overalt på Web hvor dokumentene er basert på et system. Disse systemene henter informasjon fra databaser (oftest XML databaser) og presenterer ressursene i nettleser ved bruk av HTML. Netthandlere har ofte slike systemer med slik dokumentstruktur, dette finnes for eksempel hos de fleste nettbaserte bokforhandlere. Det som kjennetegner slike dokumenter er at de presenterer metadata om en ressurs som kan etterspørres. En slik ressurs har ofte mange alternative lenker med mer informasjon om ressursen eller lignende ressurser. I slike systemer vil det typisk inngå flere lenker organisert i en lenkegruppe. I en netthandler for bøker, kan det tenkes at en ressurs har mange lenker under kategoriene; ”lignende produkt”, ”kapittel”, ”tidligere utgitte verk”, ”forfattere” og lignende.

I disse ressursene er synligheten og struktureringen av lenkene viktig. Brukerne som benytter systemet er ofte ute etter å navigere seg fram til et produkt. Det er derfor viktig at brukergrensesnittet kommuniserer med brukeren på en ryddig og strukturert måte. Forfatterens mål er å presentere en ressurs og andre relaterte ressurser slik at brukeren kan navigere seg fram til det han/hun søker på en enkel måte. Lenken må være lett tilgjengelig og godt synlig. I ressursbaserte dokumenter er det ofte mange lenker som skal presenteres, lenkene bør derfor designes slik at de tar minst mulig plass i grensesnittet. Navigasjon i dokumentet vill forbedres, ved at brukeren kan søke fram hva slags informasjon han ønsker som for eksempel ”lignende verk”, for så å få presentert tilgjengelige verk. På denne måten vil en kunne ha ryddig design samtidig som det inneholder mye informasjon. Ryddige design er enklere å navigere i enn uoversiktlige design som presenterer for mye informasjon på for liten plass. Det er også viktig at en slik lenke beskrives godt siden den i ikke står i en tekstlig kontekst. Dette kan gjøres ved bruk av ”tooltip” med rik metadata, på denne måten kan antall feiltraverseringer reduseres.

SEARCH INSIDE!™

BAD BEEKEEPING

Bad Beekeeping (Paperback)
 by Ron Miksha "Bright summer afternoon. Alfalfa and wild clover blooming..." (more)
 Explore: [Books on Related Topics](#) | [Concordance](#) | [Text Stats](#) | [SIPs](#) | [CAPs](#)
 Browse: [Front Cover](#) | [Copyright](#) | [Table of Contents](#) | [Excerpt](#) | [Back Cover](#) | [Surprise Me!](#)

List Price: \$25.50
Price: **\$25.50** & this item ships for **FREE with Super Saver Shipping**. [Details](#)
Availability: Usually ships within 24 hours. Ships from and sold by Amazon.com.
 Only 5 left in stock--order soon (more on the way).

Inside This Book

SIPs: chief apiary inspector, scale hive, shipping bees, northern beekeeper, tracheal mite (more)

CAPs: Val Marie, Swift Current, Big River, David Kleinsasser, North America (more)

Browse Sample Pages: [Front Cover](#) | [Copyright](#) | [Table of Contents](#) | [Excerpt](#) | [Back Cover](#) | [Surprise Me!](#)

Search Inside This Book:

Customers who bought this item also bought

[Robbing the Bees : A Biography of Honey--The Sweet Liquid Gold that Seduced the World](#) by *Holley Bishop*

[Beekeeping for Dummies](#) by *Howard Blackston*

[Hive Management : A Seasonal Guide for Beekeepers](#) by *Richard E. Bonney*

[A Book of Bees : And How to Keep Them](#) by *Sue Hubbell*

[The Backyard Beekeeper: An Absolute Beginner's Guide to Keeping Bees in Your Yard and Garden](#) by *Kim Flottum*

[Explore similar items:](#) in [Books](#)

Figur 17 Lenking i et ressursbasert HTML-dokument. Eksempel hentet fra Amazon.com¹⁵

Utforming av grensesnitt avhenger ofte av forfatterens motivasjon, og kan i stor grad være forskjellig. I kommersielle systemer er det ofte ønskelig å kommunisere et stort varelager, og motivasjonen bak er å få brukeren til å bruke mest mulig penger. I disse systemene blir det derfor ofte mye informasjon om produkter som leseren "kan" være interessert i. Denne informasjon må "slåss" med reklame om oppmerksomhet. Dette kan fort bli meget rotete og ustrukturert, og kan fort oppleves som negativt og virke mot sin hensikt.

5.1.2 Tekstbaserte dokumenter

Tekstbaserte dokumenter er dokumenter som inneholder informasjon i form av tekst. Formålet til et slikt dokument er å kommunisere informasjon gjennom tekstlig innhold. John Morkes og Jakob Nielsen rapporterer at hovedaktiviteten til brukerne av tekstbaserte dokumenter på Web, er å skanne etter lenker [31]. Noe av årsaken til dette er fordi lenkene i disse dokumentene tar til seg mye oppmerksomhet, og "overskygger" dermed tekstens innhold. Lenker i dagens nettlesere er ofte blå og understreket, noe som gjør de veldig synlige. Dette viser seg å minske lesbarheten betraktelig. I artikkelen "Comparing

¹⁵ Amazon.com, Elektronisk netthandler fra Amerika, www.amazon.com

Link marker visualization techniques” [34], viser det seg at den blå understrekte lenken forstyrrer så mye i teksten at den fordobler feilraten ved tekstbasert oppgaveløsning, i forhold til tekst uten lenker. Annen utforming av lenkene viser å gi et bedre resultat enn de blå understrekte lenkene[34]. Lenker i dokumenter reduserer lesbarheten til dokumentene betraktelig, design av lenker er derfor meget kritisk når lenker skal integreres i slike system.

Å ikke vise lenkene uten at brukeren ønsker det er en løsning som er mye omdiskutert. Denne løsningen er anbefalt og sett på som en optimal løsning av flere hypermedieeksperter [6]. Ved å skjule lenker når de ikke er etterspurt, lar den teksten være helt ren. Tekstens lesbarhet vil da være optimal. Problemet med denne løsningen er at lenkenes brukbarhet minskes. Det vil ta brukerne lengre tid å finne og traversere de riktige lenkene [34]. Det må derfor bestemmes om nettsiden skal være enkel å skanne gjennom, eller om den skal være lettlest.

5.2 XLink funksjonalitet i nettleser

I W3C's XLink spesifikasjon er det definert en rekke elementer og attributter for XLink. Disse er beskrevet nærmere i kapittel 2 Det er også spesifisert hvilke attributter som skal benyttes i de forskjellige XLink-elementene, og hvilke elementer som kan inneholde andre elementer. Hvert element og attributt har et bruksområde definert i XLink-spesifikasjonen. Det er ønskelig å se hvordan den tenkte funksjonaliteten for XLinken kan implementeres i HTML.

5.2.1 XLink-elementer på Web

Elementene i XLink er de som spesifiserer strukturen til lenkene. For å kunne organisere XLink i HTML må strukturen til XLink simuleres ved hjelp av HTML konsepter. XSLT kan benyttes for å transformere XLink til HTML. Konseptet til avanserte lenkestrukturer støttes ikke av HTML, men ved å kombinere HTML, XSLT, JavaScript og CSS vil en kunne simulere en XLink og mange av dens funksjoner i HTML.

Tabellen ”XLink-elementer i HTML” tar for seg de elementene som er definert i XLink spesifikasjonen og hvordan disse kan benyttes til å organisere XLink-elementer i HTML.

Elementer	Funksjonalitet	Integrert i HTML
Simple	Elementet kan benyttes som A element lenker i HTML. Disse lar brukere enkelt opprette traverseringer og deretter traversere mellom to dokumenter.	simple XLink kan implementeres i HTML ved å transformere attributtene href og title til HTML A element.
Extended	Når elementet er av type extended må applikasjonen som prosesserer dette kunne støtte bruk og visning av avansert lenkestrukturer. Applikasjonen må kunne transformere XLinken til HTML som simulere XLink funksjonalitet.	Dette elementet inneholder andre element som gir informasjon om XLinken. En extended XLink med flere endepunkt kan f.eks. implementeres som flere A element.
Resource	Ved å benytte XLink resource element i form av et HTML anker, kan det refereres til en posisjon i kildedokumentet. Dette elementet benyttes også ved multidireksjonalelenker, for å beskrive kilderessursen	Dette kan implementeres som anker i HTML ved bruk av name attributtet. Dette elementet brukes også til å identifisere lenkens tilbakegående relasjon. En bidireksjonal lenke kan i HTML gjøres ved å la et A element peke tilbake til kilden.
Locator	Dette elementet inneholder referansen til måldokumentet og er det som gjør det mulig å traversere fra et dokument til et annet i et hypermediasystem.	HTML elementet <a href> kan brukes for å opprette en lenke til måldokumentet. Et XLink locator element kan inneholde andre attributter som kan gi funksjonalitet til lenkene ved bruk av DHTML.
Title	Dette elementet muliggjør avanserte tooltips med beskrivende titler. Det kan også benyttes i en dynamisk XLink for å f.eks. vise tittelen på riktig språk etter hvor brukerne kommer fra.	Ved bruk av DHTML kan title element benyttes som metadata for ressursene eller for eksempel til å gi titlene til lenkene dynamisk funksjonalitet.
Arc	arc-elementet i en extended XLink beskriver hvordan relasjonene skal prosesserer. En kan ved hjelp av denne informasjonen, strukturer og utforme design for en-til-mange, mange-til-en og mange-til-mange relasjoner i HTML.	Disse elementene beskriver den hierarkiske strukturen til ressursene, og hvordan de relateres sammen. For å gruppere lenkene hierarkisk kan XSLT benyttes for å organisere lenkene etter arc elementet.

Tabell 3 XLink-elementer i HTML

5.2.2 XLink attributter på Web

De fleste attributtene i XLink er valgfri å benytte, en kan derfor strukturere forholdsvis enkle extended XLinker. Noen attributter er avhengig av andre attributter for å ha en

mening, xlink:from og xlink:to i arc elementene er avhengige av at xlink:label er definert i locator- og resource-elementene. Tabellen "XLink attributters bruksområde i HTML" tar for seg attributtene fra XLink-spesifikasjonen og deres bruksområde ved en implementasjon i HTML.

Attributt	Beskrivelse	Bruksområde i HTML
xlink:label	Benyttes av resource og locator element for å beskrive rollen de har ved en traversering. arc- elementene bruker label verdien til å definere xlink:to og xlink:from.	Definerer hvilke lenker elementene har del i. Funksjonen til dette elementet er å sortere hvilke lenker som har del i de forskjellige traverseringene. Dette attributtet kan benyttes for å opprette lenkegrupper i HTML.
xlink:show	Bestemmer presentasjon av ressursen som blir etterspurt. Dette attributtet støttes i en viss grad av HTML. Verdiene i XLink attributtet kan derfor transformeres til attributter i HTMLs <a target> som vil gi lik funksjonalitet. Dette kan gjøres med et enkelt XSLT stilark.	Verdien "new" kan gjøres om til "blank" i HTML, denne åpner ressursen i et nytt vindu, "replace" kan transformeres til "refresh", denne henter ressursen inn i det vinduet som benyttes ved bruk av en ny URI. Attributtverdien "embeded" kan "frame-name" som laster ressursen inn på en bestemt plassering i nettleseren.
xlink:actuate	Dette attributtet bestemmer timingen på en traversering.	Javascript støtter disse attributtene og kan håndtere disse i HTML.
xlink:role	Beskriver lenkens kontekst i forhold til andre dokument på samme tjener, eller i forhold til en lenkegruppe. Kan beskrives som en URI eller et gruppenavn.	Er dette attributtet som en URI kan det benyttes som alternativ valg av lenke, eller som en "fallback" ved "broken links". Dette kan implementeres ved bruk av JavaScript.
xlink:title	Inneholder menneskelig informasjon om ressursens navn.	En menneskelig lesbar tekststreng som brukes til å presentere traverseringen.
xlink:href	Inneholder URI eller annen adresse til en ressurs. Dette attributtet må inngå i et XLink-element slik at en ressursen kan lokaliseres. Er likt HTML's <a href> element.	Attributtet href i HTML har samme funksjon som dette elementet. Det går derfor å mappe verdien fra xlink:href til dette attributtet.
xlink:type	Attributtet beskriver hvilken type XLink-elementet er. For å ha et gyldig XLink-element må dette attributtet defineres. Det kan ha verdiene simple, extended, resource, locator eller arc.	Dette attributtet gir en applikasjon forståelse av hvordan den skal prosessere elementet. En simple XLink kan for eksempel bare integreres i HTML som et <a> element, en extended XLink er avhengig av å bli behandlet på en mer avansert måte.

Tabell 4 XLink attributters bruksområde i HTML

5.3 Design av lenker

“Designers of hypertext and hypermedia materials confront two related problems, the first of which is how to indicate the destination of links and the second, how to welcome the user on arrival at that destination.” –George P. Landlow [25]

I analogi med en reise består lenker på Web av to konsepter, adgang og ankomst. Adgang er ankeret i kildedokumentet som tar leseren til en ressurs gjennom en interaksjon med hypermediadokumentet. Ressursen som leseren kommer til er lenkens ankomst. Det er to generelle spørsmål som må tas stilling til når lenker skal utformes på Web; hvor mye informasjon må knyttes til (1) ankeret ved adgang og (2) til ankomsten, slik at brukeren skal forstå lenkene og kunne vurdere om denne vil føre han til det han søker. Utfordringen er å designe anker som ikke forstyrrer lesbarheten men som er informative, og utforming av ankomst som er intuitiv i forhold til informasjonen gitt ved adgang.

5.3.1 Utforming av lenker ved adgang

“Users shouldn't have to guess or scrub the page to find out where they can click”

- Jacob Nielsen [32]

For at bruker skal forstå at det er en klikkbar lenke i teksten, må lenkens anker utformes slik at den skiller seg ut i fra den eksisterende teksten. Relasjoner på Web visualiseres normalt ved bruk av blå font og understrek. Dette stammer fra Tim Berners-Lees¹⁶ nettleser som kjørte under ”NeXT step” og Marc Andreessens Mosaic [1]. Denne utforming ble gjort av tekniske årsaker. Fargen blå ble valgt fordi den var den nærmeste fargen til svart, og understreken ble lagt til slik at lenken skulle vises på de skjermene som hadde svart/hvit display. Dette er blitt adoptert av de fleste nettleserplattformer, og kan finnes i blant annet Internet Explorer og Mozilla Firefox. Små variasjoner i hvordan lenker blir visualisert kan ha store konsekvenser for lesbarheten til et dokument. Fordelen med å benytte stilark som CSS og XSL:FO for utforming av lenkeanker er at ved bruk av disse kan en designe ankrene etter ønske. I tillegg til å designe lenkene etter hvilken kontekst de er i, er det også flere hensyn som må tas.

¹⁶ W3C.org, Who's Who at the World Wide Web Consortium, <http://www.w3.org/People/all#timbl>

Farge

Farge er ofte brukt for å skille lenker ut fra teksten. Når det skal benyttes farge for å visualisere lenker må det tas hensyn til flere faktorer. En av tolv menn er fargeblind på en eller annen måte [15]. Dette påvirker manges oppfatning av farger. Å skille farger fra hverandre er vanskelige for fargeblinde, en må derfor være forsiktig med å legge for mye informasjon i farger. Noen farger bør ikke kombineres, som for eksempel gul på hvit bakgrunn. Dette gjør det vanskelig å lese hva som står i teksten og bør unngås. Brukes farger bør en passe på å være konsistent, dette for ikke å skape frustrasjon og misforståelser. Fargebruk bør også begrenses, fordi brukerne ikke kan huske assosiasjonene mellom flere enn tre til syv farger [32].

Typografi

Ved av bruk av typografi kan lenker skilles fra vanlig tekst. Forskjellig understrekning benyttes for å visualisere hyperlenker. Understreken gjør at lenkene skiller seg fra resten av teksten. Problemet er at den forstyrrer også bokstaver som går under linjen som for eksempel "qqj"¹. En annen metode er å variere tekststørrelsen. Dette medfører at linjeskiftstørrelsen forandres, noe som kan ødelegge layouten til dokumentet. I Peter J. Browns "Guide" [37] ble det benyttet forskjellige stilarter som fet og kursiv tekst for å markere ankre. Slik formatering begrenser mulighetene for å formatere vanlig tekst i et dokument, uten at brukerne vil misforstå de som lenker. En annen teknikk er skifte av font. Dette kan være uheldig ad ikke alle fonttyper er tilgjengelig i alle system. Designere bør derfor være sikker på at fonttypen finnes i de systemene som skal benytte designet.

Andre visualiseringsteknikker

I en brukertest utført av Harald Weinreich og Hartmunt Obendorf [34] viser det seg at ved å markere lenkeanker med et overliggende markert felt (overlay), vil lesbarheten være nesten like god som ved ren tekst, og 20 prosent bedre enn ved blå understrek. Dette er en visualiseringsteknikk som er et godt alternativ til de blå understrekte lenkene, spesielt i tekstbaserte dokumenter. Lenkens synlighet blir ikke gransket ytterligere i brukertesten, det er derfor ikke godt å vite hvor god denne lenkevisualiseringen er i forhold til lett gjenfinning av lenkene.

5.3.2 Ankomst

Ankomsten er posisjonen som brukeren tas til når han/hun traverserer en lenke. Dette er ofte en ressurs eller en plassering innen ressursen. Det må opprettes en fornuftig relasjon mellom lenkeankeret og innholdet til målressursen. Brukeren må ut i fra lenkeankeret få nok forståelse av måldokumentet til å avgjøre om det er hensiktsmessig å traversere lenken. Brukeren bør derfor ankomme på en plassering i målressursen som kan assosieres med tittelen til lenken.

I dagens nettlesere er konseptet med ankomst enkelt, enten blir hele dokumentet åpnet, heller benyttes en fragmentidentifikator til å "bla" nettleseren til en bestemt posisjon i dokumentet. Selv om en fragmentidentifikator tar brukeren til en bestemt plassering i måldokumentet, så er det et kjent brukbarhetsproblem at ikke ankomst blir markert. Et problem med fragmentidentifikatorer i HTML er hvis lenkens identifikator befinner seg nært slutten av dokumentet, så kan ikke nettleseren bli tilstrekkelig langt nok ned til å vise lenkens mål øverst i vinduet [35]. Når ikke målet til traverseringen blir markert må brukeren selv identifisere materialet som forfatteren lenker til.

Tidligere hypermediasystemer som Tim Berners-Lees nettleser og tidligere versjoner av Mosaic hadde mulighet til å beskrive et eksakt fragment i en tekst og markere dette. Dette ble gjort ved at ankeret og måldokumentet definerte startpunkt og endepunkt. Ved å innkapsle målteksten på denne måten var det mulig å lage applikasjoner som opplyste innholdet til målet. Målet ble beskrevet slik, " Teksten som lenkens anker peker til...".

På denne måten kunne referansen "#mål" referere til en deltekst og ikke bare startpunktet. Problemet med denne løsningen er at forfatteren må ha skrivetilgang til måldokumentet for å kunne forfatte et anker.

5.4 Utforming av XLink i grensesnitt

Ved implementasjon av avanserte lenkestrukturer, vil det mer en noen gang stilles krav til utforming og visning av lenker. Strukturen og den rike semantikken til slike lenker fører

til at mye informasjon må vises i grensesnittet. Hvordan disse organiseres og utformes i grensesnittet er derfor avgjørende for brukervennligheten til ressursen. XLink presenterer to nye utfordringer til design av lenker i grensesnitt; semantikk og struktur.

5.4.1 Struktur

I tillegg til å kunne presentere en-til-en relasjoner gjør datamodellen til XLink det mulig å presentere lenker med relasjonstype en-til-mange, mange-til-en og mange-til-mange. Det er spesielt relasjonstypene en-til-mange og mange-til-mange som vil føre til at lenker må uttrykkes på en annen måte enn den vi er vant til på Web. Disse relasjonstypene vil føre til at mange lenker vil bli presentert i kildedokumentet. En felles egenskap for disse lenkene er at de er assosiert til hverandre selv om de har forskjellige måldokumenter. Dette kan for eksempel tenkes i forhold til en lenke til en bok, som kan ha lenker til forfatterens hjemmeside, bokforhandleren der boken kan bestilles, et utdrag av boken, og et forum der boken diskuteres. Det er nå fire lenker som må presenteres i grensesnittet. Ved bruk av HTML kan en-til-mange lenker få en slik utforming:

```
prospects for the companies' biggest clients: communications outfits
such as Cisco (CSCO:Nasdaq - news - commentary - research -
analysis), Lucent (LU:NYSE - news - commentary - research - analysis),
and Nortel (NT:NYSE - news - commentary - research - analysis).
```

Figur 18 Figur 3 tekstutdrag fra <http://www.thestreet.com>

Trestruktur

Ved å gruppere lenkene under en foreldrenode som beskriver det overordnede temaet til lenkene, vil vi få en trestruktur på lenkene. Presenteres nodene i trestruktur er det enklere å få en overordnet forståelse av konteksten til lenken. En slik struktur vil gi oss muligheten til å håndtere lenken slik at foreldrenoden er ankeret, og undernodene bare presenteres på forespørsel fra brukeren. Presentasjon av trestrukturen vil en kunne abstrahere bort flere nivå med lenker i grensesnittet. Dette vil føre til at mindre informasjon må presenteres og vi får dermed et renere brukergrensesnitt.

En slik løsning vil reise spørsmål om hvordan ankeret bør utformes for at brukeren intuitivt skal forstå at lenken inneholder alternative valgmuligheter. Det kan tenkes to forskjellige løsninger på dette problemet:

1. Gi ankeret en annen utforming enn andre lenkeanker i dokumentet. En slik lenke bør da ha en lenke slik at brukeren implisitt forstår at denne lenken har flere nivå. Et lite plusstegn benyttes blant annet i Windows for å uttrykke at et element er utvidbart. Fordelene med dette er at brukeren lett forstår hvilke lenker som har alternative endepunkt. Ulemper med denne presentasjonen er at en vil få mange forskjellig anker og dermed mye som stjeler oppmerksomhet, noe som kan være uheldig spesielt i tekstbaserte systemer.
2. Beholde samme utformingen på denne lenken som de andre lenkene i systemet, men aktivisere underelementene umiddelbart når muspekeren går over ankeret. På denne måten vil ikke de alternative lenkene vises før brukeren peker på dem. Fordeler med dette er at en vil få et homogent grensesnitt, noe som ikke vil nedsette lesbarheten til systemet. Ulempene med dette er at lenkene kan bli vanskeligere å lokalisere, uten at brukerne søker etter de i grensesnittet. I noen systemer er lenkene utformet og plassert slik at brukeren intuitivt kan forstå deres bruksområde. Et eksempel på dette er lenke anker som er designet som en knapp.

Det må tas stilling til om underelementene i lenken skal legge seg over annen tekst som finnes i grensesnittet. Dette kan i noen tilfeller virke forstyrrende, men på en annen side være med på å opprettholde konsistensen i grensesnittet. Et annet alternativ er å flytte teksten som er under lenken når lenkene aktiviseres. Dette gjøres ved å legge informasjonen mellom linjene fra der ankeret befinner seg til setningene under. Mellomrommet mellom linjene justeres automatisk etter størrelse på informasjonen som skal presenteres. Dette kan bli meget rotete om det er mange alternative lenker.

5.4.2 Semantikk

Landlow [25] belyser problemet med å gi brukeren forståelse av forholdet mellom adgang- og ankomstressursene. Brukeren må ut i fra ankeret i adgangsdokumentet, vurdere sannsynligheten for måldokumentets relevans ut i fra det som søkes. En feiltraversering bryter med brukerens flyt i søket på målet med lesningen. Dette skaper bry, fordi brukeren forlater kildedokumentets kontekst, for så å søke gjennom ressursen

som viser seg å ikke være relevant. Brukeren må så forlate måldokumentet, og returnere til kildedokumentet for å finne det fram til det forrige lesepunktet. Å assosiere forklarende materiale med hypertekstlenker for å informere leseren om måldokumentet, gis leseren et større perspektiv på traverseringen[55].

En XLink inneholder attributter og elementtypen *title* som kan tilføre lenken rik semantisk informasjon. Lenkene skrives som XML markup-språk, dette gjør at egne elementer og attributter kan benyttes i lenkene for lenkebeskrivelse. For å dra nytte av denne informasjonen bør denne kunne vises i grensesnittet slik at bruker får en bedre forståelse av måldokumentet før lenken traverseres. Plasseringen til teksten bør ligge i nærheten av ankerets plassering. Dette kan gjøres ved å benytte tooltip eller "fluid links".

Tooltip

Tooltip er en dynamisk tekstboks som kommer til syne når muspekeren er over en trigger. Denne hjelpeteksten aktiviseres automatisk før bruker klikker på elementet, og forsvinner så snart muspekeren forsvinner fra området som aktiviserte den. Tooltipen plasseres i nærheten av muspeker, slik at distansen brukeren må flytte synet minimaliseres.

Fluid links

Ved fluid links blir informasjonen vist mellom setningene i et dokument. Denne metoden benytter animasjon slik at dokumentet "flyter" når muspekeren aktiviserer innholdet. Fordelen med denne metoden er at ingen informasjon forsvinner fra grensesnittet mens informasjonen vises[55].

5.5 XLink i HTML

Strukturen til XLink kan benyttes til å opprette lenker med flere måldokument eller flere kildedokument, med et felles anker. Ved å skjule underelementene kan det spares plass i grensesnittet

5.5.1 Trestruktur i HTML

Trestruktur i HTML lages ved først å opprette et XLink-element med en referanse til navnerommet for XLink; www.w3.org/1999/XLink. Dette elementet identifiseres som `extended`. Alle ressursene som skal inngå i traverseringen defineres som `locator` element under XLink-elementet. Det defineres så et `arc`-element for hver traversering mellom ressursene.

```
<relationship XMLNs:XLink="HTTP://www.w3.org/1999/XLink">
  <entity xlink:type=resource xlink:title="Browse" xlink:label="parent"/>
  <entity xlink:type=locator xlink:title="Front cover" xlink:href="www.frontcover.com" xlink:label="child"/>
  <entity xlink:type=locator xlink:title="Copyright" xlink:href="www.copyright.com" xlink:label="child"/>
  <entity xlink:type=locator xlink:title="Table of content" xlink:href="www.content.com" xlink:label="child"/>
  <entity xlink:type=locator xlink:title="Excerpt" xlink:href="www.excerpt.com" xlink:label="child"/>
  <go xlink:type=arc xlink:from="parent" xlink:to="child" xlink:title="Browse">
</relationship>
```

Figur 19 En-til-mange relasjon med `extended` XLink

En måte å kunne benytte XLink i et HTML-dokument er å transformere må XML-koden til HTML. Dette gjøres ved å benytte et XSLT stilark. Stilarket sorterer lenkene i en uordnet HTML liste, sortert etter hvilke elementer som hører til i de forskjellige traverseringene. Alle elementene blir lagt til listen som HTML-lenkeelement. Attributtet `xlink:href` i `locator` elementene benyttes til å beskrive `<a href>` attributtet, `xlink:title` brukes for den lesbare teksten i lenken. Listen ser da slik ut:

- Browse
 - Front cover
 - Copyright
 - Table of content
 - Excerpt

Figur 20 XLink transformert til en uordnet HTML liste

På dette stadiet er det ikke spart noe plass i grensesnittet, men det benyttes en `extended` XLink til å styre struktur og innhold i listen. Fordelene med dette er at innholdet og det

som inngår i traverseringene styres av XLinken. Dette gjør at det man enkelt kan forandre listene og hva som inngår i en lenke med å forandre på arc elementet eller xlink:label-attributtene til lenkene. For å gjøre lenken mer dynamisk kan JavaScript benyttes. JavaScriptet kan brukes til å skjule underelementene i HTML-listen. Disse aktiveres/deaktiveres ved å benytte en JavaScript "event handler" [39]. På denne måten bestemmer bruker ved en kommando om lenkene skal vises eller ikke.



Figur 21 Demonstrasjons sekvens av XLink i HTML liste, med JavaScript til styring av oppførsel.

Lenkene har samme funksjonalitet som tidligere, men tar nå mindre plass i grensesnittet. Underelementene aktiviseres når muspekeren kommer over lenken. Denne funksjonen kan forandres i JavaScriptet til å bare aktivisere lenkene ved museklikk, dobbelmuseklikk og høyreklikk.

Vi har nå en XLink som transformeres til en usortert liste i HTML, underelementene i denne listen blir vist/skjult i grensesnittet ved bruk av JavaScript. Nå styrer XLink strukturen, JavaScript styrer dynamikken til lenken, og nettleserens implementasjon av HTML bestemmer utformingen. For annen utforming og stil kan et CSS-stilark benyttes. Gjennom CSS kan en kontrollere visningen av lenkene, dette er informasjon om hvordan lenken skal vises, hvor underelement skal plasseres, farger, teksttyper, størrelse og videre.



Figur 22 Demonstrasjons sekvens av XLink i HTML med CSS formatering

CSS-stilark gir oss full styring på den grafiske delen av lenken. Benyttes stilarket er listemarkeringsen i den uordnelisten fjernet. Det øverste elementet i listen har fått svart farge og er i fet skrift, mens underelementene har fått en svak grønn farge og er flyttet

under foreldre elementet. Lenkene blir understreket når muspekeren går over dem. Distanseres informasjon om stil til et CSS dokument, er applikasjonen enkel å gjenbruke og til å tilpasse til det grensesnittet som er ønsket.

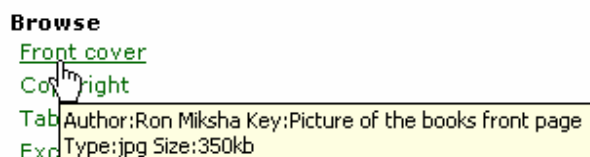
5.5.2 XLink tooltip i HTML

Semantisk informasjon fra XLink kan implementeres som et tooltip i HTML-dokumentet ved bruk av JavaScript. Av XLink-attributtene kan for eksempel xlink:title brukes i et tooltip for å vise ressursens tilhørighet i en gruppe. For å gi en bedre beskrivelse av ressursen kan et title-element opprettes for hver ressurs. Siden XLink skrives i XML kan et egendefinert XML-metadataformat for lenkebeskrivelse defineres i DTD.

```
<relationship XMLns:XLink="HTTP://www.w3.org/1999/XLink">
  <entity xlink:type=resource xlink:title="Browse" xlink:label="parent"/>
  <entity xlink:type=locator xlink:title="Front cover" xlink:href="www.frontcover.com" xlink:label="child">
    <tooltip xlink:type=title>
      <author xlink:type=title>Ron Miksha</author>
      <key xlink:type>Picture of the books front page</key>
      <type xlink:type=title>.jpg</type>
      <size xlink:type=title>350kb</size>
    </tooltip></entity>
  <entity xlink:type=locator xlink:title="Copyright" xlink:href="www.copyright.com" xlink:label="child">
    <tooltip xlink:type=title>
      <author xlink:type=title>U.S copyright office</author>
      <key xlink:type>Page with copyright laws and rights for legal copying </key>
      <type xlink:type=title>.HTML</type>
      <size xlink:type=title>unknown</size>
    </tooltip></entity>
  <entity xlink:type=locator xlink:title="Table of content" xlink:href="www.content.com" xlink:label="child">
    <tooltip xlink:type=title>
      <author xlink:type=title>Ron Miksha</author>
      <key xlink:type>The books table of content </key>
      <type xlink:type=title>.pdf</type>
      <size xlink:type=title>500kb</size>
    </tooltip></entity>
  <entity xlink:type=locator xlink:title="Excerpt" xlink:href="www.excerpt.com" xlink:label="child">
    <tooltip xlink:type=title>
      <author xlink:type=title>John Tyler</author>
      <key xlink:type>A short summary of the books content</key>
      <type xlink:type=title>.word</type>
      <size xlink:type=title>350kb</size>
    </tooltip></entity>
  <go xlink:type=arc xlink:from="parent" xlink:to="child" xlink:title="Browse">
</relationship>
```

Figur 23 Utvidet XLink med metadata

For å ta i bruk tooltip må XSLT stilarket kunne behandle de elementene som brukes. Defineres metadataelementene som et title-element, kan en lage en template i XSLT som håndterer disse elementene som tooltip. Ved bruk av JavaScript kan tooltipen integreres i HTML-dokumentet. JavaScriptet tillater en funksjon som aktiverer en tekstboks i det muspekeren går over lenkeankeret. Når tooltipen blir aktivert sendes teksten til en JavaScript metode som lager en tekstboks, og legger den i grensesnittet der muspekeren befinner seg.



Figur 24 XLink med tooltip

5.5.3 Andre mulige XLink implementasjoner i HTML

Tilbakefall

Forfatterne av HTML-dokumenter på Web har ikke kontroll på hvilke andre dokument som lenker til deres ressurser. Forandres plasseringene på dokumentene de har liggende på tjeneren vil dette føre til ugyldige lenker for de som har lenket til den tidligere plasseringen. Ved å bruke attributtet xlink:role kan det defineres en URI til ressursens gruppe på tjeneren. Ressursen kan for eksempel ha URI til område "www.tjeneradresse.no/ressurs, xlink:role vil da kunne ha adressen "www.tjeneradresse.no". Det kan derfor være interessant å implementere en tilbakefallsløsning for lenken. Med dette menes en applikasjon som ved ugyldige lenker tar brukeren til lokasjonen som er definert i xlink:role attributtet. Dette kan implementeres ved bruk av et enkelt JavaScript. Returnerer lenken en "404 error" kan JavaScriptet gi brukeren en feilmelding, og presentere en lenke med URI fra xlink:role. Ønsker ikke brukeren å følge denne lenken kan han/hun trykke vekk feilmeldingen og returnere til kildedokumentet.

Valgbare lenker

I tekstbaserte HTML-dokumenter kan det være ønskelig å kunne skru av/på lenkene for å øke lesbarheten. En slik applikasjon bør gi muligheten til å velge enkelte lenker eller ingen. XLink attributtet `xlink:arcrole` eller `xlink:title` i arc-elementet kan brukes til å gruppere lenkene. Brukeren kan så filtrere lenken etter deres gruppenavn. En gruppe kan for eksempel være forklarende lenker, lenker til figurer, ordforklaringer og lignende

Dette kan implementeres på klientnivå ved å for eksempel, ha et JavaScript som lytter til en knapp som aktiviserer/deaktiviserer lenkene i grensesnittet. Deaktiveres lenkene i grensesnittet, kan formateringen av lenken forandres via CSS til vanlig tekst. Lenkene vil da enda kunne traverseres, men de skiller seg ikke fra den andre teksten. På tjernivå kan et XSLT-stilark oppdatere dokumentet og utelate de traverseringene som er deaktivisert. Dette kan gjøres med en kommandosetning i XSLT. Problemet med å løse det på denne måten er at dokumentet må lastes inn i nettleseren på nytt for hver forandring av lenkevisningen.

6 Test case; XLink i ressursbaserte dokumentsystem

Denne implementasjonen skal utforske mulighetene med extended XLink i ressursbaserte dokumenttyper. Det er ønskelig å se hvordan en XLink kan struktureres og utformes, og hvilke fordeler en slik implementasjon vil ha for et system som allerede benytter en annen lenkemodel. Målet er å teste hva som skal til for å kunne benytte denne lenkemodulen på Web i dag, i forhold til eksisterende teknologier. Systemet XLinken skal testes mot er en prototyp av biblioteksystemet BIBSYS, der FRBR-modellen anvendes for å beskrive noder og relasjonene mellom dem.

6.1 BIBSYS FRBR-prosjektet

BIBSYS er et databasesystem som tilbyr tjenester til fag- og forskningsbibliotekenes interne drift, og gir forskere, studenter og andre tilgang til bibliotekenes ressurser. Systemet benyttes av alle universitet, alle statlige høyskoler og en rekke andre fagbibliotek i Norge [7]. Gjennom prosjektet "FRBR i bibliotekataloger", er det utviklet en prototyp av BIBSYS, hvor BIBSYS-katalogen er konvertert til FRBR. Denne prototypen er ment for å illustrere hvordan FRBR-konseptene kan benyttes i BIBSYS.

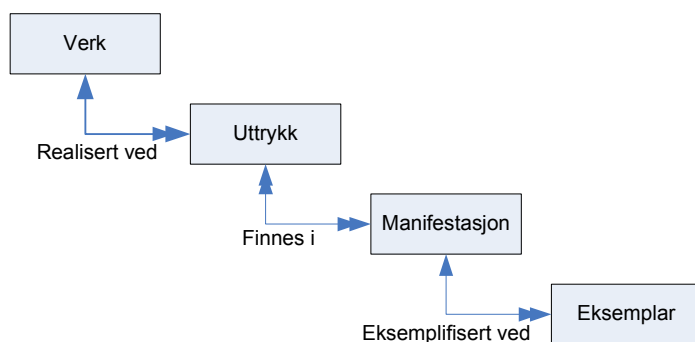
6.1.1 FRBR

FRBR (Functional Requirements for Bibliographic Records) er en konseptuell modell som representerer et generalisert syn på det bibliografiske univers. I BIBSYS-prototypen er det definert et XML-basert format av FRBR, mappingsregler og et system for konvertering av BIBSYS-MARC til FRBR [37]. Relasjonsmodellen ER (Entity relationship) benyttes for å skape en modell som speiler deler av den virkelige verden. En slik modell består av entiteter, attributter og relasjoner mellom disse. Entitetene i en slik modell deles opp i tre grupper; (1) produktene; dette er verk, uttrykk, manifestasjon og eksemplarer, (2) ansvaret; er person og korporasjon, (3) emnet; er begrep, gjenstand, hendelse og sted.

FRBR-strukturen fungerer som et rammeverk for å analysere bruk av bibliografiske opplysninger med spesiell vekt på den entiteten som brukeren er ute etter, og de attributtene og relasjonene som er relevante for brukeropp gavene. I FRBR må hvert attributt og relasjon kunne knyttes direkte opp mot brukeropp gaven de understøtter. De tilordnes derfor verdier etter hva som best støtter brukeropp gavene. Brukeropp gavene i en FRBR modell er definert som følgende:

- Å finne entiteter som svarer brukerens søkekriterier.
- Å identifisere en identitet.
- Å velge en entitet som passer brukerens behov.
- Å anskaffe eller skaffe tilgang til entiteten som er beskrevet [21].

En søkemotor vil kunne benyttes for å finne fram til en entitet, deretter benytter ofte brukerne relasjoner for å navigere fram til en entitet som passer brukerens behov. Et eksempel på en brukersekvens i en FRBR database; brukeren søker på forfatternavnet ”Ibsen Henrik” og derifra navigerer seg fram ved lenker for å finne verker som denne forfatteren er omtalt i. Relasjonens rolle i en FRBR-modell er å knytte ytterligere informasjon til ressursen for å hjelpe brukerne til å finne forbindelser til andre entiteter som er i relasjon. I FRBR er relasjonene bærere av lenker mellom entiteter, og er til for å hjelpe brukeren navigere gjennom universet som en bibliografi, katalog eller bibliografisk database representerer.



Figur 25 Entitetstyper og hovedrelasjonene i FRBR

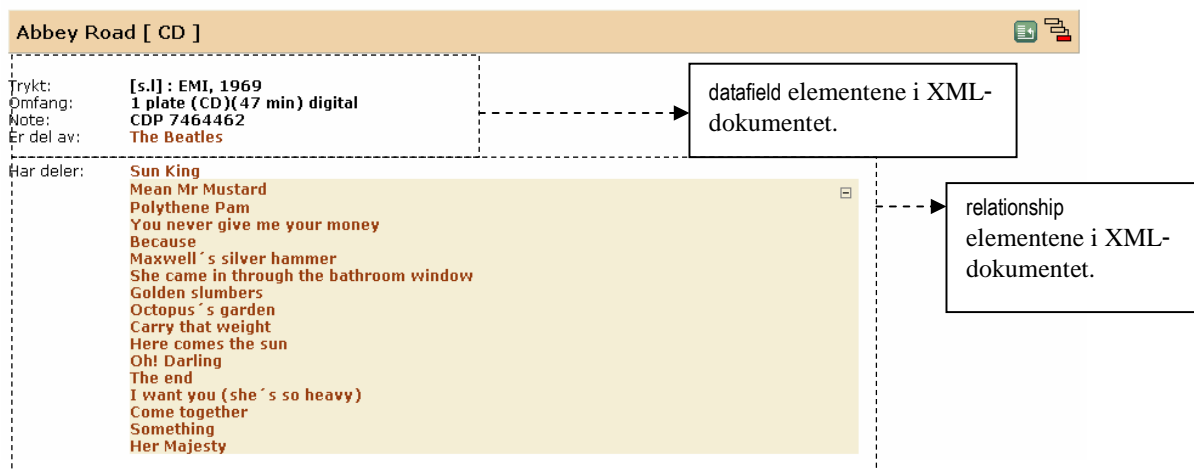
6.1.2 BIBSYS FRBRXML

All data i prototypen til prosjektet ”FRBR i bibliotekskataloger” er organisert i en XML-database der dataen blir lagret som collection bestående av flere record. En record inneholder informasjon om entiteten og hvilke relasjoner den har til andre entiteter, og består av datafield og subfield, som beskriver entiteten og relationship som beskriver relasjonene mellom entitetene. Det er spesielt elementene for relasjoner som er sentrale for implementasjonen, da det er ønskelig å strukturere disse som en avansert XLink. Elementet relationship består av attributtene rel som forteller hvilken type utgående relasjon dette er. Dette kan være relasjonsbegrep som ”er skaper av”, ”er emne for”, og lignende. Den tilbakegående relasjonen beskrives av attributtet rev. Dette elementet benyttes i systemet for å kunne søke etter relaterte entiteter, men har ikke noen rolle i presentasjon av lenkene. Beskrivelse av målressursen og forholdet den har til kilden beskrives av target. URI til målressursen defineres av attributtet href, dette attributtet inneholder en identifikator. Dokumentene blir hentet fram ved hjelp av en getDoc funksjon.

```
<relationship rel="er skaper av" rev="er skapt av" target="Verk" href="8c239033bc205ad3605621ef87e91a3b"/>
<relationship rel="er skaper av" rev="er skapt av" target="Verk" href="dce2abfb3dad001b923e82b25b00f5b4"/>
<relationship rel="er skaper av" rev="er skapt av" target="Verk" href="991f33b6845319d51cdfc969b1b53e21"/>
<relationship rel="er emne for" rev="har emne" target="Verk" href="8c20c8d037ab456c9ebbc59f988c974b"/>
<relationship rel="er skaper av" rev="er skapt av" target="Verk" href="fb7ccced2b72c68ed893c16361387465"/>
```

Figur 26 Relationship elementer i en record

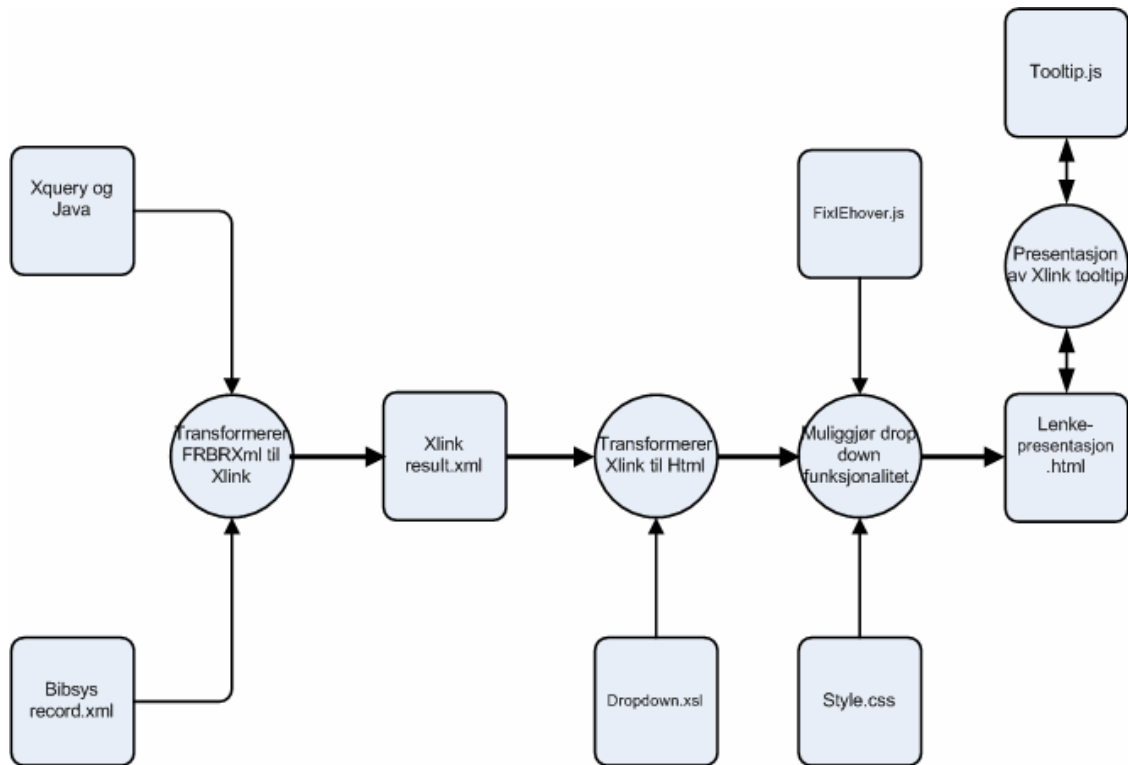
I en nettleser vises XML-dokumentene som HTML, denne transformasjonen gjøres i BIBSYS FRBR prototyp ved å benytte XSLT stilark. I ”figur 28” kan man se hvordan en record vil bli vist i en nettleser. Øverst i HTML-dokumentet er tittel og informasjon om entiteten. Her er også relasjonene representert i en rød font. En-til-mange relasjoner representeres i en maksi/minimerbarliste. Her listes alle relasjonene som hører til en gruppe, som for eksempel ”har deler”.



Figur 27 BIBSYS FRBR prototyp record i en nettleser

6.2 XLink i BIBSYS FRBR prototyp

Første steg for å kunne ta i bruk XLink i et ressursbasert dokument system som BIBSYS FRBR prototyp er å transformere den allerede eksisterende lenkemodellen til en XLink-datastruktur. Transformeringsen gjøres ved oppslag i databasen og konvertering av relationship-elementene. XLinken vil struktureres i et binært hierarki der relasjonstypene fra FRBR modellen vil være rotnodene, og lenkene løvnode. Lenkene vil deretter transformeres og legges i uordende HTML-lister på tjenernivå før dokumentet sendes til nettleser. Ved å benytte JavaScript og CSS-stilark vil lenkene utformes i grensesnittet. JavaScript vil også benyttes for å muliggjøre tooltip i grensesnittet.



Figur 28 Overordnet struktur av XLink implementasjon

6.2.1 Datastruktur

Datastrukturen til XLinken må i dette tilfellet tilpasses den allerede eksisterende lenkemodellen og informasjonen denne tilbyr. Mulighetene til XLinken vil derfor begrenses etter den informasjonen som er tilgjengelig. Det er ønskelig å benytte de mest sentrale elementene og attributtene for en extended XLink, for å se hvordan disse kan anvendes ved en implementasjon i HTML. XLinken består av elementet entity som beskriver ressursene som det lenkes til, og arc elementet go som definerer traverseringene. Elementene struktureres under elementet relationship og utgjør datastrukturen til XLinken som skal benyttes.

Relationship

I relationship elementet definerer navnerommet til XLink, dette gjør det mulig å benytte de globale attributtene for XLink. Dette elementet har attributtet xlink:type som har verdien extended, som sier at dette er en extended XLink.

Entity

Hver ressurs som er med i relasjonen presenteres ved et entity element. Elementet inneholder attributtene `xlink:type` som forteller hvilken type elementet er, `xlink:href` som inneholder en URI til elementet, `xlink:label` som benyttes for å beskrive elementets rolle ved en traversering, `xlink:title` som er lenkens menneskelige lesbare tittel, og `xlink:role` attributtet inneholder informasjon om hvilken lenkegruppe lenken tilhører.

Go

Dette elementet gis verdien `arc` og beskriver traverseringens retting og hvilke elementer som inngår i traverseringen. Elementet har her fått navnet `go` og har attributtet `xlink:type` som har verdien `arc`, attributtet `xlink:to` og `xlink:from` definerer mellom hvilke entiteter traverseringen går. Disse gis verdi lik entity-elementenes `xlink:label`. Elementet har også et attributt for å beskrive tittelen på traverseringen, dette er `xlink:title`.

Gruppering i XLink

Traverseringene av XLink grupperes ved å benytte `arc`-elementet. Denne definerer hver lenkegruppe og hvilke entiteter som er med i denne. Lenkene grupperes i to nivå, og benytter FRBR-relasjonene som rotnode og lenkens anker. I BIBSYS FRBR prototyp er denne grupperingen spesielt hensiktsmessig da alle elementene blir gruppert etter hvilken relasjon de har i forhold til kildedokumentet.



Figur 29 Gruppering av relasjoner "har del" i entiteten "Abbey road".

6.2.2 Transformering til XLink

Ved å bruke en transformeringsteknologi kan de eksisterende lenkemodellene organiseres til det XLink formatet som er ønskelig. I denne implementasjonen benyttes XQuery og Java for å gjøre oppslag og konvertere elementene fra XML databasen, til det XLink formatet som benyttes av XLink applikasjonen.

Mapping av attributter

Attributtene og elementene i BIBSYS FRBR prototypen, mappes til XLink via XQuery. Alle relasjonselementene til en entitet blir organisert i et relationship elementene i XLinken som også definerer navnerommet. Verdiene fra lenkeelementene til den tidligere lenkemodellen mappes til entity og arc element i XLink.

rel → <go xlink:title> (for rotelementet)

target → <entity xlink:role>

href → <entity xlink:href>

Figur 30 Mapping av attributter fra FRBRXml til XLink

Elementet entity kan enten være av typen locator eller resource. Målressurs får verdien locator mens kilderessursen er av typen resource. Attributtet xlink:title finnes det ingen verdi for i den eksisterende lenkemodellen. Det må derfor gjøre et oppslag i databasen for å finne entitetens navn. Attributtet xlink:label beskriver hvilke gruppe en lenke tilhører. For dette attributtet oppgis det hvilken rolle de vil spille i en relasjon i tillegg til at det autogenereres en nøkkel for hver lenkegruppe. Elementene som det lenkes fra får verdien ”p (+nøkkelverdi)” mens entitetene det lenkes til får verdi ”c (+nøkkelverdi)”. Traverseringselementet arc inneholder to attributter som spesifiserer traverseringen, xlink:from og xlink:to. Disse to attributtene definerer hvilke entity elementer som er med i traverseringen gjennom xlink:label verdiene fra entity.

```
<relationship xmlns:xlink="http://www.w3.org/1999/xlink" xlink:type="extended">
  <entity xlink:title="Abbey Road, [s.], 1969" xlink:role="Manifestasjon"
  xlink:href="6da282bbe796602d4c09d8a4885165fc" xlink:type="resource" xlink:label="p1"/>
  <entity xlink:title="Because, 1969" xlink:role="Manifestasjon" xlink:href="571455ebf28c31e55570b28eb08a761e"
  xlink:type="locator" xlink:label="c1"/>
  <entity xlink:title="I want you (she's so heavy), 1969" xlink:role="Manifestasjon"
  xlink:href="89251c9bc991867e19a544e56340a4ae" xlink:type="locator" xlink:label="c1"/>
  <go xlink:title="er del av" xlink:arcrole="er del av" xlink:from="p1" xlink:to="c1" xlink:type="arc"/>
</relationship>
```

Figur 31 XLink i BIBSYS

6.2.3 Fra XLink til HTML

Neste steg er å transformere XLinken til HTML slik at det som sendes fra tjener kan håndteres av nettleseren. Traverseringselementene til XLink transformeres derfor av en XSLT template til A-lenker i uordnede HTML lister. Templaten integreres i den allerede eksisterende stilarket som transformerer FRBRXml til HTML. Fordi lenkene organiseres samlet under et felles extended lenkeelement, må det benyttes en algoritme for å organisere lenkegruppene. Denne algoritmen strukturerer lenkene etter arc elementet og legger de elementene som er definert som xlink:from øverst i listen, og de som er xlink:to som underelementer.

```

<for hvert element velg type arc>
  <variabel Parent=xlink:from>
  <variabel Child=xlink:to>
  <variabel ArcTitle=xlink:title>
    <for hvert element velg locator eller resource>
      <if Parent=xlink:label>
        <variabel Tooltip=xlink:role>
          <a><legg elementet i øverst i den uordnede listen></a>
          <diplay tooltip på mouseover>
            <for hvert element velg locator eller resource>
              < ny tooltip>
                <if Child=xlink:label>
                  <a><legg elementet i neste nivå i den uordnede listen></a>
                  <diplay tooltip på mouseover>
                </if>
              </for hvert element>
            </if>
          </for hvert element>
        </if>
      </for hvert element>
    </for hvert element>
  </for hvert element>

```

Figur 32 Pseudo kode for lenkeorganisering i HTMLs uordnet lister

6.2.4 Utforming i grensesnitt

XLinken som leveres til nettleseren er nå bare en uordnet HTML-liste med lenker. JavaScript benyttes for å gjøre den uordende listen dynamisk. Skriptet kalles fra XSLT stilarket i BIBSYS prototypen. Det er valgbart hva som er ønskelig å benytte som anker i systemet. I denne implementasjonen defineres lenkegruppene av arc-elementet. Tittelen til lenkegruppen benyttes som navn for rotelementet til traverseringen. Dette er navn som "har del", "er del av", "er realisert av" og andre relasjoner som defineres i FRBR modellen.

Form

Utforming av lenkene defineres av CSS stilark. Lenkene bør designes etter hvilke kontekst de står i. I dette systemet som presenterer data i form av ressursbaserte dokumenter, må lenkene komme godt fram. En må derfor bestemme seg for hvilken type utforming lenkene skal få. Lenken må kommunisere interaktivitet, slik at brukerne forstår at ankeret inneholder flere underelementer. Ved å gi lenken en menybasert utforming vil lenken bli godt synlig i grensesnittet. Brukere er også vant til at menyer har flere undernivå, denne vil derfor kommunisere utvidbarhet. Ankeret blir utformet som en meny-knapp i en rektangulær boks.

Figur 33 Ankerets form

Farge

Siden denne lenken skal plasseres inn i et allerede eksisterende grensesnitt må det tas hensyn til den fargebruken som allerede eksisterer. For å få en gjennomgående design i grensesnittet bør det benyttes farger som er lik eller står i stil med de som allerede benyttes. Det er derfor ikke så mye spørsmål om hvilke farger som bør benyttes. Boksens linjer gis lik farge som hovedfargen i bakgrunnen av tittelfeltet, denne fargen blir også benyttet når muspekeren går over elementene. Tekstens farge er mørk med en gråtone, dette for at brukeren skal kunne skille mellom vanlig tekst og lenkeankerene.

Aktivisering

Lenkene i prototypen er nå organisert under hovedankerene, som er FRBR relasjonene i ER-modellen. Underelementene til lenken aktiviseres i dette systemet når musepekeren traverserer over lenkeanket. På denne måten er det enklere for brukerne å forstå at lenken inneholder flere endepunkter. Dette kan alternativt gjøres ved å benytte et venstre musknapp eller et alternativ musknapp før lenkene aktiviseres. Høyre musknapp er ofte satt av til andre funksjoner og kan derfor ikke benyttes til dette. Å benytte klikk for å aktivisere lenkene kan være hensiktsmessig der underelementene er alternative lenker.

Plassering

Lenkens anker bør få en sentral plass i grensesnittet slik at de er godt synlige. I BIBSYS FRBR prototyp er det i motsetning til andre ressursbaserte dokumenter lite som presenteres i grensesnittet. Mulighetene til plassering av lenkene er derfor mange. Lenkene presenteres som ekstra navigerbar informasjon til ressursen, og bør derfor plasseres i nærheten av metadataen for ressursen. Siden det skal benyttes utvidbare lister vil det være hensiktsmessig å plassere lenkene nederst i grensesnittet slik at underelementene ikke kommer i konflikt med annen informasjon.

I dette systemet har de fleste entitetene flere relasjoner. Lenkene må derfor organiseres i forhold til hverandre i grensesnittet. Et alternativ var å organisere lenkene til gruppene horisontalt, og vise underelementene rett under.



Figur 34 Horisontal lenkeplassering

Problemet med denne organiseringen er at ved mange lenkeelementer vil det ikke bli plass horisontalt. Legges de da på en ny linje vil det oppstå en konflikt mellom underelementene og ankerene på linje to. Det ble derfor heller valgt å organisere lenkene vertikalt, og legge underelementene på høyre side av ankerene.

Tooltip

I grensesnittet benyttes tooltip for å gi ytterligere informasjon om ressursens til brukeren. Siden lenkene transformeres fra en tidligere eksisterende lenkemodel er det begrenset hvilken informasjon som kan brukes i tooltipen. Det som ble bruket var informasjonen om hvilken type entitet resursen er. Denne informasjonen var tidligere lagret i attributtet target som nå mappes til xlink:role. Algoritmen for organisering av lenkene tillater å benytte forskjellige attributter til tooltip i hvert nivå, noe som ikke er nødvendig i denne modellen

der det ikke gis ytterligere informasjon om resursen. Dette JavaScriptet kalles også fra BIBSYSs XSLT stilark, for transformering av XML-poster.

Abbey Road [CD]

Trykt: [s.l] : EMI, 1969
 Omfang: 1 plate (CD)(47 min) digital
 Note: CDP 7464462

Har del	Sun King, 1969
Er del av	Mean Mr Mustard, 1969
	Polythene Pam, 1969
Inneholder	You never give me your money, 1969
	Back in Black, 1969
	Manifestasjon for hammer, 1969
	She came in through the bathroom window, 1969
	Golden slumbers, 1969
	Octopus ´s garden, 1969
	Carry that weight, 1969
	Here comes the sun, 1969
	Oh! Darling, 1969
	The end, 1969
	I want you (she ´s so heavy), 1969
	Come together, 1969
	Something, 1969
	Her Majesty, 1969

Figur 35 XLink i grensesnitt

6.2.5 Drøfting

Ved å transformere den eksisterende datamodellen i BIBSYS FRBR prototyp, vil vi få en lenkemodell som kan håndteres av applikasjonen som modellerer XLink i HTML. Strukturmessig vil noe av fordelene med denne modellen være at en vil slippe dupliseringer av elementer som er generelle for traverseringen, dette fordi XLink separerer traverseringen fra lenkebeskrivelsene. Vi vil i dette tilfellet slippe å duplisere attributtene rel og rev for hver traversering. Disse blir definert i arc elementet i XLinken. En ulempe med denne datamodellen er at det må defineres et element for kildedokumentet. Dette må gjøres for å ha en verdi for xlink:from slik at en gyldig traversering kan opprettes. En har derfor et ”dødt” element som i utgangspunktet ikke har

noen funksjonalitet. Presentasjonsmessig er listene oversiktlig og forståelig. Lenkeankrene er utformet som meny elementer og det er derfor intuitivt at de er klikkbare. En annen fordel med denne XLink implementasjonen er at flere lenkegrupper kan presenteres i grensesnittet, noe som ikke ble gjort med den tidligere lenkemodellen.

På grunn av at implementasjonen baserer seg på en tidligere lenkemodell vil datastrukturen til XLink være begrenset. Dette er også en begrensning på hvilken funksjonalitet som kan tilbys gjennom bruk av XLink.

- Den tidligere lenkemodellen inneholdt ikke informasjon om ressursens tittel, det måtte derfor gjøres oppslag for hvert element. Dette var en operasjon som gikk meget tregt. Det ble testet å lage oppslag i XSLT og XQuery men begge disse ble for trege. Oppslagene måtte derfor gjøres ved bruk av Java. Lenken bør derfor inneholde informasjon om tittelen for at en slik implementasjon skal være effektiv.
- Den eksisterende lenkemodellen inneholder ikke rik informasjon om ressursen. Det var derfor begrenset med informasjon som kunne presenteres i tooltip.
- Lenkemodellen inneholdt ingen informasjon om lenkens timing eller plassering. Det var derfor ikke hensiktsmessig å implementere noen løsning for bruk av `xlink:show` eller `xlink:actuate`.

Tekststrengenes størrelse ble også et problem fordi det var stor forskjell på størrelsene på disse. Størrelsen på tekstboksene til undernivået måtte derfor utvides. På denne måten kunne en få plass til mer informasjon i boksene. En alternativ løsning for dette kunne vært å dele opp tekststrengene og presentert en del av dem i tekstboksen, og resten i et tooltip.

7 Analyse

“What went wrong with pure navigation?” – George W. Furnas [19]

Dette kapitlet tar for seg XLinks datastruktur og hvilke fordeler og ulemper denne har, i forhold til bruk på Web. Det vil bli sett på utforming og bruk av XLink i ressursbaserte og tekstbaserte dokument. Implementasjonen i avhandlingen vil så bli vurdert i etter hvor generell den er i forhold til bruk av XLink.

7.1 XLink datastruktur

I forhold til XHTMLs HLink tilbyr XLink spesifikasjonen en ferdig datastruktur for organisering av avanserte lenker med mange endepunkt eller mange kildedokument. HLink kan ha en lik struktur som XLink, men bruk av HLink for å definere avanserte lenker vil kreve en ekstra prosess, av forfatteren. Dette fordi det først må spesifiseres en HLink med XLink struktur før denne kan tas i bruk. En extended XLink kan struktureres og lagres på to måter, enten som en tredjeparts lenke eller integrert i kildedokumentet. Design av datastrukturen og bruk av XPointer vil avhenge av om lenken er lagret som tredjeparts lenke eller integrert i dokumentet.

7.1.1 Tredjeparts lenke

En tredjeparts extended XLink, lagres utenfor kildedokumentet som en frittstående ressurs. Denne modellen er derfor avhengig av å definere hva som er kilderessursene, og lenkeankerets plassering i ressursen. Den store utfordringen med tredjeparts lenker er å få referert til en korrekt plassering innen kildedokumentet uten å ha skriverettigheter til dokumentet. XLink foreslår å benytte XPointer for å referere til ankerenes bestemte plassering innen et dokument. Gjennom XPointer kan en referere til plasseringer i dokument som har trestruktur og er ”well formed”, noe som gjør at XPointer ikke kan benyttes i HTML. Det kan refereres direkte til elementer og deretter benytte tekststrengsammenligning for å lokalisere ankeret. Dette gjør at XPointer er en svært nøyaktig teknologi for fragmentidentifisering. Tekststrengsammenligning i XPointer kan også benyttes til og identifiserer enkelte ord, og opprette generiske lenker til disse ved å lage ankre av alle innstansene av ordet.

Datastrukturen for extended XLink er godt tilpasset tredjeparts lenker. Dette mye på grunn av at konseptet med traversering separeres fra beskrivelsen av lenkens ressurser. Gjennom xlink:to og xlink:from attributtene i arc elementet defineres hvilke ressurser som er med i traverseringen. Det er spesielt xlink:from som muliggjør lenkebeskrivelser utenfor kildedokumentet. I tillegg til å beskrive en-til-mange relasjoner gjør dette det mulig å beskrive lenker fra mange dokument til en ressurs. Lenkene lagres som en lenkesamling på en tjener, og deles deretter for at de skal ha nytteverdi for andre brukere.

Utfordringene med å dele lenker på denne måten er først og fremst Webs størrelse. Størrelsen til Web gjør at lenker som blir publisert av tredjeparts brukere, kan være vanskelig å finne for de som ønsker å benytte lenker fra en annen bruker. Er det mange som publiserer lenker blir det fort en for stor lenkesamlinger å fordøye for brukerne. Når det lenkes til ressurser som en ikke har kontroll over så øker sjansen for at å få ugyldige lenker. For det første kan lenkene være ugyldig på to punkter, både ved kilde og mål. For det andre er større muligheter for at kildedokumentene redigeres eller flyttes uten at lenkene blir oppdatert. Lenkesamlingene må derfor vedlikeholdes ofte, slik at de ikke de "råtner"[10].

7.1.2 Lenker integrert i dokument

Lenker som er simple i XLink må integreres i kildedokumentet. Lenkens kilde er det dokumentet som lenken skrives i, og ankeret er lenkens plassering i dokumentet. XLink extended kan også integreres i kildedokumentet, dette er et mellomnivå mellom tredjeparts extended XLink og simple XLink integrert i dokumentet. Integrering av extended XLink på mellomnivå abstraherer bort deler av problemene som vil kunne oppstå ved tredjeparts lenker. Dette er spesielt problemer som er relatert til bruk av XPointer. Ved å integrere lenkene på et mellomnivå er det mulig å beskrives lenker som har relasjoner av typen en-til-mange innen dokumentet. I ressursbaserte dokumenter kan dette være spesiell nødvendig, fordi lenker ofte blir gruppert i kategorier. Disse dokumentene skrives i XML, XHTML eller i annet strukturert markup-språk, før de transformeres til HTML. Det vil da være enkelt og naturlig å skrive lenker som XLink, for deretter å integrere en applikasjon

for håndtering av lenker inn i den applikasjonen som transformerer resten av markupspråket.

Under arbeidet med utvikling av applikasjon for å støtte XLinks en-til-mange funksjonalitet, dukket det opp et problem med datastrukturen til en extended XLink. I simple XLink er adgangspunktet til lenken implisitt den plasseringen lenken befinner seg på, i en extended XLink må både kildens og målets plassering defineres i arc elementet for å få en gyldig XLink. Det vil si at selv om lenken integreres direkte i kildedokumentet må det defineres en verdi for xlink:from i arc elementet. Datastrukturen til extended XLink er for kompleks for en implementasjon på et mellomnivå, hvor lenken er integrert i dokumentet. Dette fører til at et locator eller et resource element som kan brukes til å identifisere kildedokumentet må opprettes. En ulempe med denne strukturen er at det må beskrives lenker til plasseringen i kildedokumentet, noe som fører til ekstraarbeid for brukerne når ankeret kunne ha blitt defineres implisitt fra plasseringen til lenken.

For å unngå å opprette et "dødt" element i XLinken kan elementet benyttes for å beskrive en "default" -verdi for lenken. Dette kan for eksempel være en URI til en generell forklaring av ankerets mening. En fordel med extended XLink integrert i dokumentet er at forfatteren står fritt til å trekke lenkene ut av dokumentet, og definere de i en uavhengig ressurs. Det er da opprettet en tredjeparts lenke av forfatteren til et dokument der han selv har skriverettighetene. Lenkene kan så legges inn i dokumentene på tjenernivå ved bruk av XPointer eller ved å benytte HTMLs name attributt. Integreres lenkene på et mellomnivå, vil en få all funksjonalitet som er ved tredjeparts lenker, som en-til-mange lenker og rike lenkebeskrivelser. Lenkene vil også bli enklere å vedlikeholde og gjenbruke fordi de traverseringene beskrives gjennom arc.

7.1.3 Organisering av XLink

En extended XLink med mange endepunkter framstår som en lenkegruppe, bestående av en overordnet terminologi og mange kilder som kan relateres til denne. XLink-elementene struktureres innkapslet i lenkens rotelement, denne identifiserer om lenken er av type simple eller extended. Det må derfor tas stilling til hvordan elementene skal struktureres i XLinken. Lenkene kan organiseres separat, hvor hver lenke inneholder et antall locator- og resource-elementer og et arc-element som spesifiserer traverseringen, eller så kan lenker samles under samme rotelement med flere arc-element for å definere flere traverseringer. En extended XLink kan enten struktureres slik at hver lenke bare inneholder en lenkegruppe, eller så kan flere samles under samme rotelement. Ved å gruppere alle locator og resource elementene under et felles rotelement, vil en slippe duplikasjoner av locator og resource elementene om en lenke inngår i flere lenkegruppergrupper. I ressursbaserte dokumenter kan dette være en god løsning for organisering av lenkene. Samles lenkene under samme rotelement er de enklere å vedlikeholde, fordi traverseringer kan opprette, slettes og redigeres gjennom å forandre på arc elementet. Lenkene kan også trekkes ut av ressursen og lagres lokalt. Gjøres dette vil det for eksempel være enklere å skrive applikasjoner som søker etter ugyldige lenker, og fjerner de fra traverseringene.

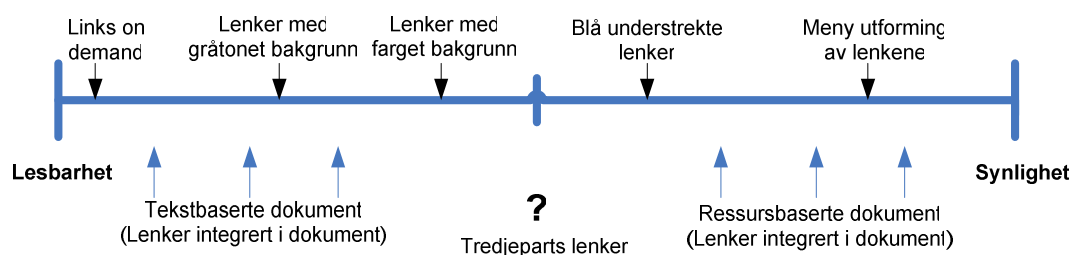
Den hierarkiske strukturen innen lenkegruppen bestemmes av arc elementet som spesifiserer et binært hierarki. I utgangspunktet støtter XLink bare hierarki på to nivå, men et dypere hierarki kan struktureres ved å benytte flere arc elementer. Et flernivå hierarki kan se slik ut:

```
<go xlink:type="arc" xlink:from="P" xlink:to="C1">
  <go xlink:type="arc" xlink:from="C1" xlink:to="C2">
```

Vi uttrykker her en lenke fra "P" til "C1", og videre til "C2". Applikasjonene som tolker XLinken må kunne håndtere flernivå lenker for at dette ikke skal bli uttrykt som to separerte lenker. Uttrykkes lenkene som trestruktur i grensesnittet vil de kunne uttrykkes i flere nivå. Dette vil være hensiktsmessig der det er mange elementer som struktureres i en menybasert utforming.

7.2 Utforming og bruk av XLink

Utformingen av en XLink bør tilpasses konteksten lenken befinner seg i. I kapittel 5 foreslås en todeling av dokumenter på Web, ressursbaserte og tekstbaserte dokument. Disse to dokumenttypene kommuniserer innhold via grensesnittet med forskjellig hensikt. Lenkene i grensesnittet er med på å underbygge målene, og må derfor utformes etter hva grensesnittet ønsker å formidle. Verdier lesbarhet og synlighet er omvendt proporsjonale. Det vil si at hvis synligheten forsterkes minskes lesbarheten og omvendt. For at lenkene skal kunne tilpasses grensesnittet, må lenkene utformes og lagres lokalt, eller ved at designeren studerer kildene lenken skal integreres i. Lenkene må derfor skrives av forfatter, og være integrert i dokumentet for at utformingen skal kunne tilpasses kontekst. Tredjeparts lenker er vanskeligere å gi en bestemt utforming fordi designeren av disse lenkene vet hvilken type dokumenter lenkene skal integreres i.



Figur 36 Utforming av lenker i grensesnitt

7.2.1 XLink i tekstbaserte dokument

Lesbarhet er et nøkkelbegrep når det er snakk om tekstbaserte dokument. Lenker i slike dokumenter bør forstyrre teksten og lesbarheten minst mulig. I tekstbaserte dokumenter står oftest lenkene i en ganske streng kontekst, noe som gjør det enklere å anta målressursens innhold. Målet til tekstbaserte dokument er å kommunisere tekst, lenker i teksten bør derfor ha forklarende eller refererende funksjon. Det er ikke ønskelig å presentere mange forskjellige ressurser til brukeren. Dette kan føre til at brukeren begynner å "browse" og dermed mister flyten i lesningen. Alternative lenker kan presenteres på slutten av dokumentet. Lenkene i slike dokumenter bør være konsis og relevant i forhold til tekstens helhet. Feiltraverseringer i slike dokumenter er kritisk,

lenkene er derfor i stor grad avhengig av rik semantisk støtte. Ankomsten er også meget viktig for lenker i et slikt dokument. Brukeren bør få presentert det relevante tekstfragmentet direkte.

I tekstbaserte dokumenter kan det være hensiktsmessig å benytte simple XLink. Dette fordi tekstbaserte dokumenter ikke bør inneholde for mange alternative ressurser. Det vil derfor sjeldent være nødvendig å benytte en lenkemodul som extended XLink for å opprette lenker i teksten. En simple XLink støtter alle attributtene spesifisert for XLink, og kan dermed inneholde rik semantikk. I tekstbaserte dokumenter vil forfatteren beskrive lenker integrert i teksten, det er derfor ønskelig at lenkene implisitt har adgang fra der de er plassert. Dette gjør det mer aktuelt å bruke en simple enn extended XLink. Utvides lenken med XPointer kan lenkens ankomst defineres nøyaktig. Med XPointer kan start og sluttpunktet for dokument defineres. Dette gjør at en kan forespørre bare en del av en tekst, trekke den ut og presentere denne for bruker. Det eksisterer bare begrenset støtte for XPointer, den er også avhengig av "well formed" dokument og kan derfor bare benyttes for å referere innen XML og XHTML – dokument.

I artikkelen *"Comparing Link Marker visualization techniques– change in reading behaviour"* [34] bevises det at markerte anker minsker lesbarheten i dokumenter. Test av usynlige lenker som bare vises på forespørsel blir beskrevet i artikkelen. Det testes her å benytte usynlige lenker som bare vises på forespørsel. Resultatet er at brukerne løser oppgaver basert på tekstens innhold 25 prosent bedre enn når ankerene blir vist. Det kommer også fram at brukerne feiler over 10 prosent mer på oppgaver basert på lenketraversering. Slike lenker støttes av XLink gjennom muligheten til å gruppere lenkene med `xlink:role`-attributtet. En vil da i tillegg til å kunne gjemme alle lenkene, og kunne identifisere noen typer lenker som en ønsker at skal vises i grensesnittet. Diskusjonen rundt bruk av lenker på forespørsel, som omhandler en slik modell vil undergrave lenkenes rolle i Webs hypertekstmodell.

7.2.2 XLink i ressursbaserte dokument

I ressursbaserte dokument er det ofte lite tekst og mange lenker. Lenkene må derfor organiseres oversiktlig slik at de enkelt kan lokaliseres av brukerne. Grensesnittet til denne dokumenttypen har måtte lide for HTMLs enkle lenkestruktur og utforming. Lenkene i ressursbaserte dokument er ofte organisert under en terminologi som for eksempel ”andre lignende ressurser”, ”andre ressurser av denne forfatteren” og lignende. Disse lenkene står ofte ikke i en bestemt kontekst, men som uavhengige menyer. Dette gjør at det er vanskeligere for brukeren å vite hva han/hun skal forvente av målressursen. Nielsen sier at et lenkeanker bør være på 3-7 ord [33], noe som kan være mye å ”presse” inn i lenkebaserte menyer. Det kan derfor være ønskelig å tilby ekstrainformasjon til brukerne i form av for eksempel tooltip.

I slike dokumenter vil det være behov for avanserte lenkestrukturer som extended XLink. XLink extended på mellomnivå har mye funksjonalitet som vil gjøre jobben til en designer av ressursbaserte dokument enklere. Ved å kunne samle lenkene med lik terminologi under en extended XLink vil et grensesnitt kunne organiseres mer effektivt og oversiktlig. Lages lenkene i trestrukturer som aktiviseres av muspeker, vil en spare mye plass i grensesnittet, og det vil være enklere for brukerne å fordøye informasjonen enn når de blir presentert for alle lenkene på en gang. Brukerne vil få et overordnet perspektiv gjennom lenkegruppene over tilgjengelige relasjoner, for deretter å søke gjennom de interessante. For å gi menystrukturene en pen design, må ankrene være korte. Det må derfor tilføres ekstra informasjon til brukeren gjennom løsninger som tooltip, som det er stor støtte for ved bruk av XLink. Lenkeankrene må utformes slik at brukeren intuitivt kan forstå at ankeret inneholder underelementer. Dette kan for eksempel gjøres ved å designe elementene som knapper.

I ressursbaserte dokumentsystem vil det være hensiktsmessig å implementere XLink på et mellomnivå. Tredjeparts lenker vil ikke være like aktuelle i disse dokumentene, fordi lenkene i ressursbaserte dokumenttyper ofte går innad dokumentsystemet, og det er lite ønsket å ha lenker som forvirrer brukeren. En vil i mindre grad være avhengig av å

benytte XPointer teknologi, om ikke forfatteren ønsker å separere lenkene fra kildedokumentet.

7.2.3 Utforming av XLink ved tredjeparts lenking

Tredjeparts lenker skaper designkonflikt på Web. Dette fordi lenkene blir lagt til av en applikasjon på klientsiden. Denne har ingen informasjon om grensesnittet og typen dokumentet lenkene skal integreres i. Lenkene kan derfor ikke tilpasse utformingen av dokumenttypen og grensesnittet. Dette er en stor utfordring siden lesbarhet og synlighet må balanseres, og alle dokumenter har forskjellig utforming i forhold til farge, font, størrelse og så videre.

Det må derfor designes en generell utforming for tredjeparts lenker som passer til alle dokumenttyper. Problemet med dette er at utformingen av lenkene aldri vil bli optimal, i forhold til lenkens kontekst. Det må inngås et kompromiss mellom ressursbaserte og tekstbaserte dokumenter.

7.3 Implementasjon av XLink for ressursbaserte dokument

Implementasjonen for håndtering av XLink i ressursbaserte dokumenttyper, er en generell XLink applikasjon for presentasjon av relasjoner med flere endepunkter. Denne applikasjonen støtter strukturering av XLink på et mellomnivå, der flere lenker inngår i samme extended element. XLink-elementene resource og locator benyttes for å beskrive ressursene som inngår i lenken, og arc elementet beskriver traverseringen mellom disse. Elementene som inngår i lenken må innkapsles i et extended element som også definerer lenkens navnerom. Elementene resource og locator må inneholde attributtene xlink:title, xlink:href og xlink:label, mens arc elementet må ha verdi for xlink:title, xlink:to og xlink:from, for at denne applikasjonen skal håndtere lenken på en fornuftig måte. Denne datastrukturen er lik den som identifiseres av XLink spesifikasjonen [53].

XSLT template benyttes for å transformere og organisere XLink elementene i HTML. En template i stilarket organiserer lenkene i et binært hierarki etter hva som er definert i XLinkens arc element. XSLT templatene styrer også hvilke verdier som skal benyttes som

tooltip. I ressursbaserte dokumenttyper som allerede har en XSLT transformasjon kan templatene integreres i stilarket, og kalles på lik måte som er gjort i BIBSYS FRBR prototypen.

Informasjon om lenkenes utforming er lagt til et eget dedikert CSS stilark. I implementasjonen i FRBR prototypen av BIBSYS, brukes xlink:title-elementene som lenkenes anker. Gjennom stilarket kan utseendet for lenkeankrene spesifiseres. CSS gjør at stilen kan spesifiseres separat, det er dermed enklere å tilpasse utformingen til lenkene etter dokumenttypen den skal integreres i. JavaScript benyttes for å styre lenkens oppførsel. Denne teknologien benyttes for å styre dynamikken til lenken. Aktivisering av lenken bestemmes av JavaScriptet. Her kan metoden som benyttes til å åpne underelementene forandres. Dette gjøres her på onMouseOver men kan også gjøres med for eksempel onClick.

Bruk av tooltips for visning av XLink attributt muliggjøres gjennom bruk av JavaScript. Attributtene som benyttes for tooltip kan forandres med et enkelt inngrep i algoritmen som organiserer lenken. Denne kan også utvides til å støtte title-element. Tooltipene aktiviseres gjennom bruk av JavaScript.

Transformerer XLinken av forfatter på tjenernivå til HTML kan lenken tilpasses kontekst og støttes av de fleste Web plattformene. Benyttes en tjener eller proxy-løsning for transformering av lenkene, kan det sendes informasjon om hvilken nettleserplattform som benyttes for å vise lenken. Forskjellige plattformer kan for eksempel ha forskjellige JavaScript implementert [4]. Det må derfor sendes en annenledes JavaScript kode etter hvilken plattform som skal benyttes. Implementasjonen av JavaScript er forskjellig fra Internet Explorer og Mozilla. Applikasjonene som er implementert for BIBSYS FRBR prototypen sjekker derfor hvilken type nettleser som benyttes for så å sende passende JavaScript til klienten.

Applikasjonen tilbyr i likhet med XLink2HTML en XSLT-transformering av XLink til HTML. XLink2HTML er det systemet som i størst grad har lik bruk av teknologi som

lenkeapplikasjonen som er beskrevet i denne avhandlingen. XLink2HTML bruker også XSLT for å transformere XLink til HTML, men benytter JavaScript til visning av lenkene og ikke CSS. I applikasjonen som er testet i BIBSYS er det lagt større vekt på utforming av lenkene i forhold til lenkens kontekst, enn i tidligere systemer utviklet for å håndtere XLink. Designforslag fra artikkelen ”*The look of the link*” [35] har blitt vurdert i forhold til implementasjonen. Løsningen med dynamiske lenkelister for presentasjon av en-til-mange relasjoner er adoptert fra denne artikkelen, og applikasjonen er et forslag til hvordan dette kan implementeres. Presentasjonen av lenken er ikke ment som å kunne benyttes som en generell utforming av XLink, da den er spesielt tilpasset ressursbaserte dokumenttyper.

Applikasjonen i forhold til W3Cs mål med XLink:

- *XLink skal være lett å benytte på Web.* Implementasjonen gjør at en forfatter kan skrive XLinker med flere endepunkt og rik semantikk, uten å måtte tenke på hvordan dette kan presenteres på Web. De fleste nettlesere støtter bruk av HTML, CSS og JavaScript.
- *XLink skal representere den abstrakte strukturen og de betydningsfulle delene ved lenking.* Ved å benytte denne applikasjonen kan lenker struktureres på andre måter enn det som er vanlig på Web.
- *XLink skal kunne ligge utenfor dokumentet der resursen ligger.* Dette er ikke implementert og er vanskelig å gjøre uten støtte for XPointer. Dette er også med på å gjøre lenker mer sårbare, siden mulighetene for broken links dobles.
- *XLink skal være lesbar for mennesker.* Applikasjonen støtter bruk av forskjellige lenkeanker med tooltip som supplement for å gi bruker forståelse av lenkene.

7.4 XLink-støtte på Web (design)

Mye av XLinks funksjonalitet kan simuleres ved hjelp av allerede eksisterende teknologi. Implementasjonen av XLink i FRBR prototypen viser at det er mulig å dele opp strukturen til XLink og presentere den i HTML. Ulempen med å benytte HTML for og presentere XLink er at det krever en applikasjon for håndtering av lenkene. Dette kan først abstraheres fra lenkeforfatterens tankemodell når det er opprettet en generell modell

for håndtering av XLink. En generell modell er vanskelig å opprette da konteksten lenken blir skrevet i vil variere. Implementasjonen i avhandlingen gir brukeren selv muligheten til å utforme lenkene gjennom et CSS dokument som er dedikert til lenkeutformingene.

De fleste nettleserne i dag støtter JavaScript, dette gjør at denne teknologien er velegnet til å gjøre XLink dynamisk i HTML – dokumenter. En ulempe med bruk av dynamisk innhold er at nettleseren ikke gir tillatelse til å vise dynamisk innhold. Dette er fordi dynamisk innhold kan være ustabil og gi virus. Noen deaktiverer også dynamisk innhold fordi dette ofte er reklame og ikke av interesse for brukeren. Bruk av JavaScript kan føre til at flere brukere ikke får presentert underelementene til lenkene.

Doczilla er den eneste nettleseren til dags dato som har støtte for extended XLink. Nettleseren støtter et bredt spekter av extended XLink. Presentasjonen av XLink i Doczilla, bryter i stor grad med alle designprinsipper for lenker i grensesnitt. Ved å markere lenker med en stor gul lenkeboks og duplisere denne etter antall endepunkter til lenken skaper Doczilla store brytninger i lesbarheten til dokumentet. Lenkeankeret har en statisk utforming og vil derfor være for stor i noen dokumenter og for liten i andre. Dette er et utformingsproblem som vil være typisk ved tredjeparts lenking. Lenkeankeret inneholder ingen semantisk informasjon annet enn ved tooltip. Det er dermed vanskelig å anta hva som er lenkens endepunkt. Doczilla foreslår også en annen måte å behandle lenker med flere endepunkter. Dette er å utforme kun et anker for hver flerveislenke og gjennom en meldingsdialog fortelle brukeren at det inngår en lenke til i traverseringen. Brukeren kan så velge om han/hun ønsker å traversere videre eller å ende stoppe i dokumentet som er lastet inn i nettleseren. Denne løsningen kan være veldig uheldig om det inngår mange endepunkt i en lenke. Løsningen baserer seg på en hierarkisk presentasjon av dokumentene basert på relevans. Det første dokumentet som presenteres til brukeren har størst odds for å være det relevante. Brukeren får ikke et overordnet perspektiv på lenkene og det er vanskelig å få forståelse for hvilke alternative lenker som inngår i relasjonen.

Skal XML benyttes på Web, bør det tilpasse seg variasjonene i veven framfor å standardisere grensesnittene. Selv om GUI-spesialister ofte påstår at forskjellige brukeropplevelser er forvirrende, så beviser Web at forskjellige design ikke alltid er like forstyrrende. Brede muligheter skaper ofte innovative design, dårlig design vil skape frustrasjon og forvirring, mens standardiserte design vil skape homogenitet. XML er et fleksibelt markup-språk, dette bør også reflekteres i design av grensesnitt. Homogent design bør derfor ikke være den logiske konsekvensen av å benytte XML i grensesnittet. CSS abstraherer det logiske nivået av en XLink struktur, og ved å forandre på dette kan brukeren styre utformingen av lenkene.

7.4.1 XLink simple VS extended

“You should use XLink whenever your application is one of hypertext linking, as XLink functionality such as power to control user interface behavior on link traversal is useful and should be implemented in a standard way to allow interoperability.”

-Tim Berners-Lee[5]

Avanserte lenker bør integreres slik at dens avanserte funksjonalitet er enkel å benytte, blir riktig visualisert og er valgbar. Valgbarheten er viktig, fordi det ikke alltid er like praktisk å benytte de avanserte lenkemekanismene. XLink deler derfor lenking i extended og simple relasjoner. Disse to har forskjellige funksjonalitet fordi de er tilpasset forskjellig bruk. XLink av typen simple tilbyr lite ekstra i forhold til den godt brukte A-lenken til HTML. Noe av det nye XLink simple skulle introdusere var bruk av xlink:show og xlink:actuate for å bedre kunne spesifisere hvordan lenker skal aktiviseres. Dette er implementert i både Mozilla og Doczilla, men med den samme effekten tilgjengelig gjennom bruk av HTML og JavaScript. Dette er mulig å implementere i blant annet nettlesere som Internet Explorer. Det er derfor ingen store fordeler med bruk av simple XLink i forhold til den lenkingen som er mulig å gjøre i HTML.

Lenker av typen extended og bruk av XPointer introduserer lenkefunksjonalitet som tidligere ikke har vært tilgjengelig på Web. Spesielt i ressursbaserte dokumentsystemer vil funksjonalitet som en-til-mange relasjoner være til nytte. Mulighetene til å skrive

lenker med rik semantikk og tilby dette til bruker som ekstra metadata for lenketraverseringene, vil gi færre feiltraverseringer og dermed forbedre navigasjon i hypertekst. Bruke av avanserte lenkestrukturer krever mer innsats av forfattere, da de må beskrive ekstra semantikk for lenkene og i noen tilfeller også se om det finnes alternative ressurser som det går å lenke til. Dette er noe som kan gjøre standarden mindre attraktiv å ta i bruk for en utvikler. På en annen hånd vil det enklere å skrive en-til-mange lenke enn mange enveis lenker. En slik lenke vil også være enklere å håndtere i et brukergrensesnitt.

8 Konklusjon

"Make it as simple as possible, but not simpler" -Albert Einstein

8.1 Oppsummering av arbeid

Avhandlingen tar med utgangspunkt i XLink for seg avanserte lenkestrukturer. Det er lagt spesiell vekt på extended XLinks struktur og hvordan denne kan brukes opp mot Web. Flere teknologier for implementasjon av XLink på Web er vurdert i forhold til bruk av støtte av mye brukte nettleserplattformer. Avhandlingen tar også for seg hvordan avanserte lenker bør utformes i forhold til den konteksten de er i. Teorien blir testet med en implementasjon av XLink med flere endepunkt og rik semantikk I BIBSYS FRBR prototyp. Implementasjonen benytter XLink, HTML, XSLT, JavaScript og CSS, og støttes av de fleste nettleserplattformer som benyttes i dag. Utformingen av denne applikasjonen har her blitt spesielt tilpasset for ressursbaserte dokumenttyper. Oppgaven har så vurdert XLinks datastruktur og mulighetene for utforming av XLink på Web.

8.2 Konklusjon

Avhandlingen har delt ressurser på Web i to dokumenttyper, ressursbasert og tekstbasert. Lenkene spiller forskjellige roller i de ulike dokumenttypene. Behovet for extended XLink i tekstbaserte dokumenter er mindre enn ved ressursbaserte dokumenter. XLink extended med mange endepunkt vil skape forstyrrelser i tekstbaserte dokumenter, det vil derfor være hensiktsmessig å benytte HTML lenker eller simple XLink, og heller begrense lenkebruken.

8.2.1 XLink i tekstbaserte og ressursbaserte dokument

Tekstbaserte dokumenter er de som vil ha mest bruk for tredjeparts lenker. Dette fordi brukeren leser disse dokumentene og prøver å få en forståelse av innholdet. Det kan da være nødvendig å tilby et supplement i form av tredjeparts lenker. Ressursbaserte dokument består ofte av lite tekst, og lenkene som er hensiktsmessig er ofte utformet av forfatter. Problemet med tredjeparts lenker er at de vil være i konflikt med lesbarheten til de tekstbaserte dokumentene. Dette spesielt siden de det er vanskelig å spesifisere en

generell utforming som passer til alle dokumenter. Utformingen til tredjeparts lenker vil derfor være meget forstyrrende for lesbarheten i tekstbaserte dokument. Fordelen med slike lenker er at de ikke hentes inn før brukeren selv aktiverer de. Tekstbaserte dokument har et behov for bruk av XPointer for å identifisere nøyaktige ankomstposisjoner i ressurser. I ressursbaserte dokumenter er ikke XPointer for å markere endepunkt like viktig. Dette er fordi brukeren ønsker å få presentert en ny ressurs ved en traversering, og ikke tilleggsinformasjon for å for eksempel forstå et konsept. Noe av grunnen til lite interessen rundt XLink, kan være på grunn av det er blitt lagt for stor vekt på tredjeparts lenker, når dette kan vise seg å være en funksjon som tar mer enn den gir.

Avhandlingen viser gjennom teori og implementasjon at det kan være hensiktsmessig å benytte extended Xlink implementert på mellomnivå i ressursbaserte dokumenttyper. Ressursbaserte dokumenttyper forfattes ofte i XML og transformeres ved bruk av XSLT. Gjennom implementasjonen i BIBSYS FRBR prototyp, vises det hvordan XLink kan integreres i det allerede eksisterende XSLT stilarket, og ved bruk av HTML lister, A-lenker, JavaScript og CSS utformes extended XLink med flere endepunkter og rik lenkebeskrivelse. Lenkene kan kjøres på de fleste nettleserplattformene som eksisterer i dag. Markedet for slike lenker på Web er spesielt ressursbaserte dokumentsystem med nettbaserte portaler. Med en økning på 45 prosent fra 2004 til 2005 på norske netthandlere [38], er det kanskje et lite håp for XLink allikevel.

8.2.2 Resultat

Erfaringene av arbeidet med denne avhandlingen kan oppsummeres med følgende punkter:

- Mye av XLink kan implementeres på Web med teknologier som er tilgjengelig.
- Det er et større behov for extended XLink i ressursbaserte dokument enn i tekstbaserte dokument.
- Tredjeparts lenker vil være avhengig av en generell utforming.
- Bruk av XPointer til å definere kildedokument ved tredjeparts lenker øker muligheten for ugyldige lenker.

- Mangler ved XLink datastruktur ved implementasjon av extended XLink på mellomnivå.
- Transformering fra en eksisterende lenke modell til XLink, kan skape problemer fordi XLink avhenger av rik semantikk.

8.3 Videre arbeid

Med videre arbeid innenfor dette feltet ville det vært aktuelt å se på mulighetene med XPointer på Web. XPointer er drøftet i avhandlingen, men det er ikke gått i dybden på spesifikasjonen. Dette er mye på grunn av tidsaspektet til oppgaven, og fordi det ikke ville vært hensiktsmessig for implementasjonene. I forhold til XLink standarden har derfor denne applikasjonen utviklingspotensialet spesielt i knyttet til XLink funksjonalitet som er avhengig av XPointer. Det ville derfor vært naturlig og videreutviklet prototypen med bruk av XPointer på Web.

Videreutvikling av prototyp i forhold til XLink spesifikasjonen:

- Støtte for xlink:show og xlink:actuate. Dette finnes det støtte for ved bruk av JavaScript, noe som allerede er implementert i blant annet Doczilla og Mozilla.
- Tredjeparts lenking kan gjøres ved bruk av denne modellen, men er ikke mulig i HTML dokument før de støtter bruk av XPointer.
- Mange-til-en og mange-til-mange relasjoner er ikke implementert. Disse relasjonene er spesielt for tredjeparts lenker. Det vil da kreve bruk av XPointer eller annen fragmentidentifikator for å spesifisere kildene.
- Bi direksjonale lenker, kan implementeres på Web ved å lenke fra målressursen tilbake til kilden. Har en ikke rettigheter til måldokumentet må det også benyttes en fragmentidentifikator her.
- Markert ankomst, er også noe som må gjøres gjennom bruk av fragmentidentifisering.

Referanser:

- [1] Andreessen, M.: NCSA Mosaic Technical Summary. NCSA, University of Illinois, 1993.
- [2] Arciniegas Fabio, XML.org Postet av forfatteren av XLink2HTML, lists.XML.org/archives/XML-dev/200008/msg00404.HTML
- [3] Ashman Helen & Duncan Martin, Goate: XLink and beyond, University of Nottingham, ACM press Juni 2002
- [4] Ashman Helen & Duncan Martin & Truran Mark, Goate, anyone?, ACM conference on Hypertext and Hypermedia August 2003
- [5] Berners-Lee Tim, Design issues XLink, W3C 2002, www.w3.org/DesignIssues/XLink.HTML
- [6] Bernstein, M.: HypertextNow: Showing Links, Eastgate Systems, 1996. www.eastgate.com/HypertextNow/archives/ShowingLinks.HTML
- [7] BIBSYS hjemmesider, www.bibsys.no
- [8] Bos Bert, Cascading Style Sheets, W3C 2005, www.w3.org/Style/CSS
- [9] Brown, P. J. Turning Ideas into Products, The Guide System, Hypertext'87 Proc., ACM Baltimore, 1987
- [10] Cailliau Robert, Hypertext in the web – a History, ACM Press 1999
- [11] Catledge L. D. & Pitkow J. E., Characterizing Browsing Strategies in the World-Wide-Web, Computer Networks and ISDN Systems, side. 1069, 1995.
- [12] Ciancarini Paolo, Folli Federico, Rossi Davide, Vitali Fabio, “XLinkProxy: External Linkbases with XLink”, Department of Computer Science, University of Bologna
- [13] Computer history museum, Xhibits: Internet History, 2006, www.computerhistory.org/exhibits/internet_history
- [14] Daniel Veilard, XML linking implementations, W3C 2005 www.w3.org/XML/2000/09/LinkingImplementations.HTML
- [15] Digma.no- artikler, foredrag og kurs om digitale medier, www.digma.no/tips/fargeblind.html

- [16] ECMA, ECMAScript for XML (E4X) Specification, 2 edition, ECMA international Desember 2005, www.ecma-international.org/publications/files/ECMA-ST/Ecma-357.pdf
- [17] Elliotte Rusty Harold & W. Scott Means, "XML in a nutshell 3rd edition", O'REILLY 9 September 2004
- [18] Empolis Bertelsmann Mohn media group, x2x intelligent linking, support.eqnet.hu/~empolis/pdf/products_x2x_en.pdf
- [19] George W. Furnas, Effective view navigation ACM 22-27 March 1997
- [20] Goodger Ben & Hickson Ian & Hyatt David & Watterson Chris, XML user interface Language (XUL), Mozilla 2001, www.mozilla.org/projects/xul/xul.html
- [21] IFLA universal Bibliographic Control and International MARC Programme, Funksjonskrav til bibliografiske poster (Oversettning av Functional requirements for Bibliographic Records), oversatt av Liv Aasa Holm
- [22] Kahan Jose / Amaya, "XML Linking Implementation Questionnaire" lists.w3.org/Archives/Public/www-xml-linking-comments/2000OctDec/0023.html
- [23] Kirjavainen Mika, XML can be displayed in a browser several ways, 2000, mia.ece.uic.edu/~papers/WWW/MultimediaStandards/XML_browsers.pdf
- [24] Kristol David, HTTP Cookies: Standards, privacy, and politics, ACM press 2001
- [25] Landow George P., Relationally Encoded Links and the Rhetoric of Hypertext, Brown University, Providence
- [26] Masatomo Goto presenting Fujitsu, XML Linking Implementation Questionnaire, lists.w3.org/Archives/Public/www-xml-linking-comments/2000OctDec/0044.html
- [27] Michael Kay, Saxon: Anatomy of an XSLT processor, www-128.ibm.com/developerworks/library/x-xslt2
- [28] Mozilla, Xml in Mozilla –XML linking and pointing, developer.mozilla.org/en/docs/XML_in_Mozilla#XML_Linking_and_Pointing

- [42] Wilde Eric & Lowe David , Improving Web linking using XLink, University of technology Sydney 2001
- [43] Wikipedia online encyclopedia, Cascading style sheets, en.wikipedia.org/wiki/Cascading_Style_Sheets
- [44] World Wide Web Consortium, HLink - Link recognition for the XHTML Family, W3C Working Draft September 2002
- [45] World Wide Web Consortium, HTML 4.0.1 specification -Links, W3C Recommendation 24.12.1999, www.w3.org/TR/1999/REC-html401-19991224
- [46] World Wide Web Consortium, Extensible markup language (XML) 1.0 (third edition), W3C Recommendation 04.02.2004, www.w3.org/TR/REC-xml
- [47] XML.com, Theodor Holm Nelson, O'REILLY 2006, www.XML.com/pub/au/53
- [48] World Wide Web Consortium, Extensible markup language (XML): Part 2. Linking, Working draft July 1997, www.w3.org/TR/WD-xml-link-970731
- [49] World Wide Web Consortium, Extensible Stylesheet Language (XSL) Version 1.0, W3C Recommendation 15 October 2001, www.w3.org/TR/xsl/
- [50] World Wide Web Consortium, Overview of SGML resources, www.w3.org/MarkUp/SGML/
- [51] World Wide Web Consortium, XHTML 1.0 the Extensible hypertext markup language (second edition), W3C recommendation januar 2000, www.w3.org/TR/2002/REC-xhtml1-20020801/
- [52] World Wide Web Consortium, XHTML 2.0, W3C working draft mai 2005, www.w3.org/TR/XHTML2
- [53] World Wide Web Consortium, XML Linking Language (XLink) Version 1.0, W3C recommendation juli 2001, www.w3.org/TR/XLink/
- [54] World Wide Web Consortium, XML Linking Language (XLink) Design Principles", W3C mars 1998, www.w3.org/TR/NOTE-xlink-principles
- [55] World Wide Web Consortium, XML Pointer Language (XPointer), W3C working draft januar 2001, www.w3.org/TR/xptr/

- [56] World Wide Web Consortium, XML Path Language (XPath) Version 1.0, W3C Recommendation november 1999, www.w3.org/TR/xptr/
- [57] World Wide Web Consortium, XML Path Language (XPath) Version 1.0, W3C Recommendation november 1999, www.w3.org/TR/xptr/
- [58] Zellweger, Chang, Mackinlay, Fluid links for informed and incremental link transition ACM Press 1998

URI sjekket: 01.06.2006
