Anne Leinonen

XML-STANDARDIPERHE JA ELEKTRONISTEN DOKUMENTTIEN ARKISTOINTI

Tietojärjestelmätieteen
pro-gradu -tutkielma
8.6.1998

Jyväskylän yliopisto
Tietojenkäsittelytieteiden laitos
Informaatioteknologian maisteriohjelma
Digitaalinen media

TIIVISTELMÄ

XML-standardiperhe ja elektronisten dokumenttien arkistointi
Leinonen, Anne
Tietojärjestelmätiede, Jyväskylän yliopisto, 8.6.1998
81 s.
Tutkielma

XML-standardiperhe on uusi dokumenttien hallintaan kehitetty standardi. XML-standardiperheen muodostavat XML-dokumenttien ominaisuudet määrittävä XML-standardi sekä sitä tukevat XLink-linkkistandardi ja XSL-tyylistandardi. XML-standardiperhe on kehitetty erityisesti verkkokäyttöön sopivaksi. Sen kehitystyö on vielä XLink- ja XSL-standardien osalta kesken.

Työssäni tarkasteli XML-standardiperhettä elektronisten dokumenttien arkistoinnissa. Analysoin arkistoinnin piirteitä verkkoympäristössä ja pohdin, miten elektroniset dokumentit tulisi arkistoida. Dokumenttien arkistoinnilta edellytetään, että dokumentit ovat löydettävissä arkistosta dokumenttaja kuvailevan metatiedon avulla. Metatieto onkin yksi arkistointiin liittyvä näkökulma tässä työssä.

XML-standardiperheen ominaisuuksia käsittelin XML-standardiperheen määrittelytiedon ja arkistointia tarkastelin kirjallisuuden avulla. Käytännössä kokeilin XML:ää kahdella eri ohjelmalla. Kokeiluilla oli tarkoitus saada tietoa XML-standardiperheen käytettävyydestä. Kokeiluissa tein XML-dokumentin, liitin siihen XSL-tyylimäärittelyt ja muodostin niistä esitettävän HTML-dokumentin, sekä kokeilin metatiedon liittämistä XML-dokumenttiin.

Analysointini osoitti, että arkistoidut dokumentit vaativat ylläpitoa. Niihin pitää liittää tekijänoikeuksien määrittämiä käyttörajoituksia, erilaisia varmistustekniikoita sekä metatietoa. XML-standardiperhe mahdollistaa erilaisen tiedon liittämisen XML-dokumentteihin, koska dokumenttien sisältö ja fyysinen koostumus voidaan jakaa pienempiin osiin. XML-dokumenttien jakaminen erilaisiin osiin vaikuttaa myös dokumenttien käsittelyyn. Käytännön kokeilut osoittivat, että XML:n käyttö riippuu ohjelmasta. XML:n perusajatusten toteuttaminen, kuten tyylimääritysten liittäminen XML-dokumenttiin, onnistuu ohjelmasta riippumatta.

AVAINSANAT: XML, XLink, XSL, arkistointi, metatieto, elektroninen dokumentti

ABSTRACT

The XML standard and archiving of electronic documents
Leinonen Anne
81 pages
Master's Thesis of Information Systems Science
University of Jyväskylä, Jyväskylä, Finland, 8.6.1998

Extensible Markup Language (XML) is a new standard under development for document management in internet, intranet and extranet environments. The XML standard has three members: XML describes a class of data objects called XML documents, XLink specifies a set of constructs to describe links between objects, and XSL is a stylesheet language. The XML standard is designed by the World Wide Web Consortium (W3C).

The main goal of the thesis is to study the XML standard and archiving of electronic documents. I have analysed the characteristics of archiving and considered, how documents should be archived. Documents should be found by metadata. That's why the metadata is one important viewpoint of the thesis.

I have used the specifications of the XML standard to define the characteristics of each member. The study of archiving electronic documents was based on the literature. I have carried out two tests with different programs to find out how XML will work. The first phase of the test was to produce an XML-document. In the second phase the XML-document and the XSL-document was combined. In the third phase the metadata was added to the XML-document. The result was examined as an HTML-document. The tests pointed out that applying XML is at the moment very dependent on the program used.

The archived documents need to be maintained. There are also restrictions to use documents, techniques to protect them from changes or withdrawal, and metadata which have to be added to them. The XML standard makes this possible because the content, the structure and the physical appearance of XML-documents may be separated from each other. The separation gives also new challenges for maintaining electronic documents.

KEYWORDS: XML, XLink, XSL, archiving, metadata, electronic documents

SISÄLLYS

1 JOHDANTO

XML (eXtensible Markup Language) on uusi standardi, joka on tarkoitettu tiedon tallennukseen ja jakeluun verkossa. XML-standardoitu tieto koostuu *XML-dokumenteista.* Dokumentteja käsitellään tietokoneohjelmilla, joita kutsutaan *XML-prosessoreiksi.* XML-standardissa (Bray, Paoli & Sperberg-McQueen, 1998) määritellään, millaisia ovat XML-dokumentit ja osittain myös se, miten XML-prosessorit toimivat. XML:ään liittyvät läheisesti XML-linkityksiä varten suunniteltu XLink-linkkistandardi (Extensible Linking Language) ja XML-dokumenttien esitystavan määrittelyyn suunniteltu XSL-tyylistandardi (Extensible Style Language). Näitä kolmea yhdessä kutsutaan *XML-standardiperheeksi.*

Kolmivaiheisen XML-standardiperheen kehittämisprojektin aloitti W3C:n (World Wide Web Consortium) työryhmä toukokuussa 1996. Ensimmäisen vaiheen tuloksena kehittämisprojekti julkaisi 10.2.1998 XML-standardin, version 1.0, virallisena W3C:n suosituksena. Toiseen vaiheeseen kuuluvan XLink-standardin alustava määritys julkaistiin huhtikuussa 1997. Kolmas vaihe on XSL-tyylistandardi. Standardointiehdotus julkaistiin elokuussa 1997. XML-standardia tukevat linkki- ja tyylimäärityksstandardit ovat vielä kehitteillä. (A Proposal, 1997; Bray & DeRose, 1997; Bray ym., 1998.)

XML-standardoinnin taustalla on kaksi aikaisemmin määriteltyä merkkauskieltä: yleinen dokumenttirakenteiden määrittelyyn ja rakenteisten dokumenttien merkkaamiseen tarkoitettu SGML (Standard Generalized Markup Language), ja verkkodokumenttien esittämiseen tarkoitettu, SGML-kielellä määritetty, HTML (Hypertext Markup Language). HTML:n vahvuus on sen sopivuus verkkojakeluun ja dokumenttien esittämiseen verkossa. World Wide Webin (WWW) laajeneminen tuo verkkoon yhä monimutkaisempia, monipuolisempia ja suurempia dokumentteja, joiden käsittelyyn HTML ei tarjoa riittäviä mahdollisuuksia. SGML:ssä tiedon rakenteinen tallennustapa puolestaan antaa dokumenttien käsittelylle monenlaisia mahdollisuuksia. Koska SGML on kehitetty ennen ny-

kyistä verkkotekniikkaa, siinä on paljon erilaisia ja harvoin käytettäviä ominaisuuksia. Yleisten SGML-sovellusten rakentaminen ei ole sen vuoksi yksinkertaista. XML:stä haluttiin verkkokäyttöön sopiva standardi tiedon esittämistä, käsittelyä ja siirtoa varten, joten siihen yhdistettiin ominaisuuksia sekä SGML:stä että HTML:stä. (Bosak 1997; Lander 1997.)

Opinnäytetyössäni tarkastelen XML-standardiperheen tarjoamia mahdollisuuksia ja rajoituksia elektronisten dokumenttien arkistoinnissa. Tarkastelu perustuu kirjallisuudesta ja käytännön XML-kokeiluista saamiini tietoihin.

Työssäni kerron aluksi XML-standardiperheen taustalla olevista standardeista, SGML:stä ja UNICODEsta, (luku 2). Sen jälkeen tarkastelen XML-standardiperhettä yksityiskohtaisemmin (luku 3), ja esittelen visioita XML:n verkkosovelluksista (luku 4). Luvussa 5 käsittelen arkistointia ja XML:ää arkistoinnissa. Luvussa 6 raportoin tulokset käytännön XML-kokeiluista.

Opinnäytetyötäni edelsi raportti "XML:n käyttö verkkosovelluksissa". Raportti syntyi Helsingin yliopiston kirjaston koordinoimassa EVA-projektissa (Elektronisten verkkojulkaisujen hankinta ja arkistointi). EVA-projektin tavoitteena on luoda menetelmät ja välineet elektronisten verkkojulkaisujen hankintaa, rekisteröintiä ja arkistointia varten, sekä selvittää pitkäaikaissäilytyksen edellytyksiä kirjastoissa.

Työssäni keskeinen käsite on *dokumentti*. Dokumentin ymmärrän Spraguen (1995) määrittelemällä tavalla: dokumentti on yhtä aihetta koskeva ja ihmisen ymmärrettäväksi tarkoitettu erilaisilla symboleilla esitetty kokonaisuus, joka tallennetaan ja jota käsitellään kokonaisuutena. Dokumentit ovat tässä työssä *elektronisia dokumentteja*, jotka on tallennettu, ja joita käsitellään, jaetaan ja käytetään *verkkoympäristössä*. Verkkoympäristössä olevat dokumentit ovat sisällöltään hyvin erilaisia. Sitä kuvaa Salmisen (1997) määritelmä: "Dokumentin sisältö koostuu osista, jotka ovat tekstiä, kuvia tai ääntä. Osa voi sisältää muita osia." Verkkoympäristöllä tarkoitan työssäni inter-, intra- ja ekstranet -ympäristöjä.

2 XML-STANDARDIPERHEEN TAUSTALLA OLEVAT STANDARDIT

XML-standardiperheen taustalla on kaksi standardia SGML ja UNICODE/ISO 10646. Ne antavat XML-standardiperheen standardeille niiden perusominaisuudet. XLink- ja XSL-standardien taustalla on vielä muita standardeja, jotka määrittelevät yksityiskohtaisemmin niiden piirteitä. Tässä luvussa käsittelen vain SGML:n ominaisuuksia ja keskeisiä käsitteitä sekä UNICODE-standardin perusajatusta. XLinkin ja XSL:n taustalla olevia standardeja tarkastelen seuraavassa luvussa.

2.1 SGML-standardi

SGML (Standard Generalized Markup Language) on kansainvälinen ISO-standardi 8879 vuodelta 1986. SGML on metakieli, jolla määritellään muita kieliä. Se määrittelee syntaksin merkkauskielille, joilla voidaan määritellä ja koodata dokumenttien rakenne. Se ei määrittele, kuinka dokumentteja käsitellään tai millaisen esitystavan dokumentit saavat. Se on tarkoitettu dokumenttien hierarkkisen rakenteen merkkauksen määrittelyyn. (Goldfarb, 1990.)

SGML-dokumentit muodostuu kolmesta osasta (DeRose, 1997, 198):

1. *Esittelyosassa* (SGML declaration) on sovelluskohtaista tietoa dokumenttien syntaksiin liittyvistä valinnoista. Siinä määritellään käytettävä merkistö sekä mm. sovelluksessa käytettävät SGML:n valinnaiset ominaisuudet. Esittelyosa voi puuttua, mutta silloin käytetään standardin oletusasetuksia (reference concrete syntax).

2. *Rakennemäärittely* (Document Type Definition, DTD) määrittelee dokumenttityypin hierarkkisen rakenteen eli kaikki kyseisen dokumenttityypin *elementit* sekä niiden hierarkkisen järjestyksen. Jokaisella elementillä on nimi, joka määrittelee elementtityypin. DTD:ssä määritellään myös kuhunkin elementtiin liit-

tyvät *attribuutit* eli lisämääreet. Jokaisella attribuutilla on nimi ja arvo. Näiden lisäksi DTD:ssä määritellään *entiteetit* ja niiden sijainti. *Entiteetti* on nimetty merkkijono tai tiedosto, johon viitataan entiteettiviitteellä. DTD:n entiteettimäärittelyssä kerrotaan entiteetin nimi ja arvo (merkkijono) tai sitten arvon sisältävä tiedosto.

3. *Sisältöosa* (document instance) on dokumenttien varsinainen sisältö, jonka rakenne merkataan tunnisteilla. Elementti nimeltään e alkaa alkutunnisteella <e> ja päättyy lopputunnisteeseen </e>. Alku- ja lopputunnisteen välissä on elementin sisältö. Attribuutit merkataan elementin alkutunnisteeseen. Sisältöosa voi olla jakautunut yhteen tai useampaan kokonaisuuteen, entiteettiin, jotka on tallennettu erikseen, ja joihin viitataan entiteettiviitteillä. Entiteettiviitteiden avulla voidaan dokumentteihin sisällyttää erilaista tietoa, kuten kuvia, graafeja, tekstikappaleita tms. Esimerkiksi jos kuva tallennetaan tiedostoon ja DTD:ssä entiteettinimi kuva liitetään tähän tiedostoon, voidaan sisältöosassa viitata kuvaan entiteettiviitteellä &kuva;. Esitettävässä dokumentissa entiteettiviite korvautuu kuvalla. (Goldfarb, 1990, 18, 27.)

SGML-standardissa on dokumenttien hallinnan kannalta seuraavia ominaispiirteitä (Salminen, 1995):

- monikäyttöisyys ja käsittelyvaiheiden automatisointi
- sovellus- ja laitteistoriippumattomuus
- pitkäaikainen säilytettävyys
- siirrettävyys
- tiedon hallinta.

Dokumenttien monikäyttöisyydellä ja käsittelyvaiheiden automatisoinnilla tarkoitetaan sitä, että SGML-dokumenttien käsittelyyn on mahdollista rakennetaa ohjelmia, jotka käyttävät hyväksi dokumenttien rakennetietoa. Dokumenteista voidaan esimerkiksi valita vain tietyt elementit ja muodostaa niistä jotain alkuperäisestä dokumenteista poikkeavaa. Samalla ohjelmalla voidaan muuttaa useita dokumentteja, ja samasta dokumentista saadaan erilaisilla ohjelmilla monia erilaisia dokumentteja. Rakenteisesta tekstistä voidaan hakea kulloinkin

tarvittavat tiedot. Tällöin vältytään koko dokumentin monenkertaiselta kirjottamiselta, paperille tulostamiselta sekä kopioimiselta. (Salminen, 1995.)

Koska dokumentit tallennetaan SGML-standardin mukaan rakenteisesti, se ei ole riippuvainen minkään tietyn järjestelmän edellyttämästä muodosta. Muunnosohjelmia ei tarvita dokumenttien siirtämiseksi järjestelmästä toiseen, jos molemmat järjestelmät tukevat SGML:ää. SGML-dokumenttien elinikä ei ole ohjelmistojen tai laitteistojen eliniästä kiinni. Dokumenttien esitystapa voi kuitenkin olla organisaation oman standardin määrittelemä. Silloin se voi olla laitteisto- tai ohjelmistoriippuva. Esitystapatietoja ei kuitenkaan tarvitse välttämättä tallentaa pysyvästi dokumentteihin, vaan ne voidaan liittää siihen esim. tulostuksen yhteydessä. SGML-standardi on sopiva pitkäaikaiseen säilytykseen. SGML-merkattu tieto on ymmerrettävissä ja tulkittavissa alkuperäisessä merkkijonomuodossa silloinkin kun käytössä ei ole sellaista ohjelmaa, joka tulkitsisi ja tulostaisi SGML-muotoisen dokumentin. (Salminen, 1995.)

SGML-standardin mukaisesti merkatut dokumentit ja siihen liittyvät määrittelyt ovat siirrettävissä ympäristöstä toiseen. Dokumenttien käyttömahdollisuudet riippuvat vastaanottajan ohjelmista ja niiden SGML-tuesta. Siirto ja sen jälkeinen dokumenttien käsittely on vaivatonta, jos vastaanottajan ohjelmassa on dokumenttityyppimäärityksen käsittelytavat määritelty samoin kuin lähettäjän. Organisaatio voi määritellä oman standardin tietojensa ja dokumenttiensa esitystavaksi. Koska SGML on kansainvälinen ja yleisesti käytetty standardi, voidaan organisaatiokohtainen standardi kehittää jo jostain valmiina olevasta sovellusalueen standardista. (Salminen, 1995.)

SGML-ohjelma tarkistaa aina, että dokumentit ovat sille määritellyn rakenteen mukaisia. Tämä lisää organisaation dokumenttien tietojen luotettavuutta. Tiedon etsintää ja löytymistä SGML-muotoinen rakenteinen tallennustapa tehostaa. Hakumekanismien käytössä on dokumenttien sisällön merkitykseen liittyvä rakenne, jota on täydennetty lisätiedoilla, attribuuteilla. (Salminen, 1995.)

Myös tekijänoikeudelliset asiat voidaan huomioida SGML-standardissa. Koska SGML-standardin mukaisesti dokumenteista erotetaan rakennemääritys, varsinainen tekstisisältö, merkkausmerkinnät ja esitysmuoto. Niille voidaan kaikkille määrittää tekijänoikeudet erikseen. (Salminen, 1995.)

2.2 UNICODE-standardi

UNICODE on kansainvälinen merkkien koodausjärjestelmä tietokoneella tapahtuvaa merkkien käsittelyä varten. Se tukee kaikkien maailman yleisimpien kielten kirjoitusmerkkejä, ideografisia merkkejä (mm. Kiinan kielen merkit) sekä symboleja. UNICODE on kansainvälisen ISO 10646-1-standardin osajoukko. (The Unicode Standard, 1996.)

UNICODE-standardin kehittelyn tavoitteena oli (The Unicode Standard, 1996):

1. *Maailmanlaajuus*: merkkijärjestelmän tuli olla niin kattava, että se sisältäisi kaikki mahdolliset merkit, joilla tietoa välitetään, huomioiden kansainväliset, kansalliset ja teollisuuden käyttämät merkit.
2. *Tehokkuus*: dokumentit, jotka muodostuvat etukäteen pituudeltaan määriteltyistä merkeistä, ovat yksinkertaisia jäsentää, koska ohjelman ei tarvitse pysähtyä etsimään erikoismerkkien määrittelyjä.
3. *Yhtenäisyys*: määritelty merkkijärjestelmä mahdollistaa dokumenttien tehokkaan lajittelun, tiedonhaun, esittämisen ja editoinnin.
4. *Yksiselitteisyys*: merkille on määritelty tietty 16-bittinen arvo, joka tarkoittaa aina juuri tätä merkkiä.

UNICODE perustuu ASCII-merkistölle. Yhtä merkkiä kuvataan 16 bitillä. Tämä mahdollistaa sen, että standardissa on eri kielille ja symboleille riittävästi yksilöllisiä arvoja. Eri kielten ja kulttuurien erityismerkkien määrittely on ollut mahdollista, eikä merkkien käyttö vaadi mitään erityistoimenpiteitä. UNICODE-standardi määrittelee numerisen arvon kullekin nimetylle merkille. Sen lisäksi se määrittelee merkkien ominaisuuksia, kuten isot ja pienet kirjaimet, sekä sisältää sellaisia vapaita koodeja, joita sovellukset voivat käyttää omiin tarkoituksiinsa. (The Unicode Standard, 1996; ks. myös The Unicode Standard: A Technical Introduction, 1996.)

UNICODE-standardi on suunniteltu tukemaan eri kielillä kirjoitettujen dokumenttien siirtämistä, käsittelyä ja esittämistä. Se tukee lähes kaikkien maailman kielten merkkejä sekä joitakin jo käytöstä jääneitä historiallisia kieliä. Siihen sisältyy myös mm. välimerkit, merkit, jotka liitetään tavallisten merkkien yhteyteen osoittamaan ääntämistä, matemaattiset ja tekniset symboolit, ja nuolet. Dokumentissa voi olla aivan vapaasti sekaisin eri kielisiä merkkejä ja symboleja. UNICODEen on varattu vapaita koodeja myös tulevaisuuden tarpeille. (The Unicode Standard, 1996; ks. myös The Unicode Standard: A Technical Introduction, 1996; Noerr, 1995.)

UNICODE-standardia ovat olleet kehittämässä eri tietokonealan yritykset, standardointiorganisaatiot ja lingvistiikan asiantuntijat. ISO 10646-standardi valmistui suurin piirtein samoihin aikoihin kuin UNICODE. UNICODEn 65 536 koodiarvoa ovat myös ISO-standardissa ensimmäiset 65 536 koodiarvoa. Loput ISO-standardin koodeista on varattu tulevaisuuden tarpeille. UNICODE-standardia ylläpidetään ja kehitetään UNICODE Consortiumissa Kaliforniassa. (The Unicode Standard: A Technical Introduction, 1996; Noerr, 1995.)

3 XML-STANDARDIPERHE

Tässä luvussa kuvaan XML-standardiperheen ominaisuuksia. Käsittelen yksitellen kaikkia standardiperheen standardeja, XML:ää, XLinkkiä ja XSL:ää. Ensiksi tarkastelen XML-standardin ominaisuuksia yksityiskohtaisemmin. Sitten esittelen XLink- ja XSL-standardit. Tarkastelen kummankin standardin taustalla olevat standardit sekä niiden ominaisuudet ja perusajatukset. Standardien esittelyn jälkeen kerron XML-standardiperheen suhteesta SGML-standardiin ja HTML-kieleen. Lopuksi teen lyhyen yhteenvedon standardiperheestä, ja tarkastelen XML-standardiperheen kehitysnäkymiä

3.1 XML-standardi

XML-standardin perusajatus ilmenee parhaiten niistä tavoitteista, jotka sen kehittämisen lähtökohdiksi asetettiin. XML-standardiin liittyvät käsitteet ja toimintaperiaatteet puolestaan määrittyvät tarkastellessani lähemmin XML-dokumentteja.

3.1.1 XML-standardin kehittelyn lähtökohta ja tavoitteet

XML-standardi on kehitetty SGML-standardista. SGML-standardoidun tiedon rakenteiseen tallennustapaan perustuvia ominaisuuksia haluttiin saada verkkokäyttöön, kuten esimerkiksi tiedon uudelleenkäytettävyys eri yhteyksissä. Toisaalta SGML:n heikkous on se, ettäSGML:ssä on paljon harvoin käytössä olevia ominaisuuksia. Sen vuoksi haluttiin SGML:ää kevyempi standardi niin käytettävyyden kuin sovellusten teonkin kannalta. Erityisesti haluttiin kehittää verkkoa monipuolisesti hyödyntävä standardi. (Bray ym. 1997; ks. myös Khare & Rifkin 1997 ; ks. myös Lander 1997.) XML:n kehittelyssä on yksityiskohtaisiksi tavoitteiksi asetettu (Bray ym. 1997):

1) XML:ää pitää pystyä käyttämään Internetissä;

2) XML:n on oltava laite- ja ohjelmistoriippumaton niin, että monet erilaiset sovellukset voivat sitä hyödyntää;

3) Yhteensopivuus SGML:n kanssa;

4) XML:ää käsittelevien ohjelmien kirjoittamisen on oltava helppoa;

5) XML-dokumenttien on oltava sellaisia, että mikä tahansa XML-prosessori osaa käsitellä niitä eli dokumenttien käsittelyyn vaikuttavia piirteitä on rajoitetusti;

6) XML-dokumenttien tulisi olla luettavia ja selkeitä;

7) XML-määrityksen pitäisi valmistua nopeasti;

8) XML-määrityksen suunnittelun on oltava tarkkaa ja huolellista;

9) XML-dokumenttien tekemisen on oltava helppoa;

10) XML:n määrittelykielen mukaiset tunnisteet saavat olla pitkiä.

3.1.2 XML-dokumentit

XML-dokumentit muodostuvat loogisesta ja fyysisestä rakenteesta. Looginen ja fyysinen rakenne muodostavat toimivan kokonaisuuden. XML-dokumentit ovat joko valideja tai hyvin muodostettuja (well-formed). Validit XML-dokumentit ovat aina myös hyvin muodostettuja. XML-prosessori tunnistaa XML-dokumenttien hierarkkisen rakenteen joko DTD:n perusteella (validit XML-dokumentit) tai elementtien alku- ja lopputunnisteiden avulla (hyvin muodostetut XML-dokumentit).

XML-dokumenttien looginen ja fyysinen rakenne

Looginen rakenne on dokumenttien sisällön rakenne. Silloin kun XML-dokumentit sisältävät DTD:n, dokumenttien looginen rakenne määritellään siinä. Looginen rakenne muodostuu (Bray ym., 1998):

- esittelyosasta (vrt. SGML declaration), joka on pakollinen valideissa XML - dokumenteissa, mutta ei hyvin muodostetuissa,
- elementeistä,
- kommenteista,
- entiteettiviitteistä,
- CDATA-osioista, joiden sisältö tulostetaan aivan täsmällisesti sellaisena kuin kirjoittaja on sen kirjoittanut, vaikka siihen sisältyisi määrittelykielen käyttöön varattuja erikoismerkkejä, ja

- sisällön käsittelyohjeista.

Kaikki nämä loogisen rakenteen erilaiset osat osoitetaan dokumenteissa tunnisteilla.

XML-dokumenttien *fyysinen rakenne* on useiden erillisten kokonaisuuksien, *entiteettien*, yhdistelmä. Kaikilla entiteeteillä on nimi ja sisältö. Sisältönä voi olla mm. kuvia, tekstiä, merkkijonoja, ääntä, videopätkiä yms. Jokainen entiteetti on tallennettu itsenäisesti. Niihin viitataan *juuressa eli dokumenttientiteetissä.* XML-prosessori aloittaa koko dokumenttien käsittelyn juuresta ja kokoaa dokumentteihin kuuluvat muut entiteetit juuressa annettujen käsittelyohjeiden mukaisesti. (Bray ym., 1998.)

Hyvin muodostettu XML-dokumentit

Hyvin muodostettuihin (well-formed) XML-dokumentteihin ei tarvitse liittyä DTD:tä. Dokumentit ovat *hyvin muodostettuja XML-dokumentteja,* jos (Bray ym., 1998):

1) niihin sisältyy vähintään yksi elementti, ja

2) juureksi tai dokumentitelementiksi kutsuttu elementti on sellainen itsenäinen elementti, joka ei ole minkään muun elementin lapsielementti. Kaikkien muiden elementtien sisällä lapsielementit ovat sallittuja, kunhan lapsielementti alkaa ja päättyy saman elementin sisällä, ja

3) ne noudattavat kaikkia syntaksissa määriteltyjä hyvin muodostettujen dokumenttien sääntöjä, joissa määritellään mm. kuinka tunnisteet kirjoitetaan täsmällisesti oikein, mitkä ovat entiteettiviittauksien rajoitukset, miten binääriseen tietoon viitataan jne., ja

4) kaikki dokumenttien entiteetit ovat hyvin muodostettuja.

Hyvin muodostettuihin XML-dokumentteihin ei välttämättä tarvitse sisältyä yhtään esittelyosaa. Niihin voi kuitenkin sisältyä ainakin yksi esittelyosa seuraavasti:

```
<?xml version="1.0"?>
```

Tässä esittelyosassa määritellään, mitä XML-versiota käytetään. Jos hyvin muodostetuissa XML-dokumenteissa on entiteettejä, ne on määriteltävä heti esittelyosan jälkeen. (Bray ym., 1998.)

Seuraava tilauslista on esimerkki hyvin muodostetusta XML-dokumenteista, jossa ei ole yhtään esittelyosaa:

```
<tilaus>
<kappalenumero>593</kappalenumero>
<kappalemäärä>200</kappalemäärä>
<kappalehinta>72</kappalehinta>
</tilaus>
```

Esimerkissä on äitielementtinä `tilaus`. Sillä on kolme lapsielementtiä: `kappalenumero`, `kappalemäärä` ja `kappalehinta`. Jokainen lapsielementti päättyy ennen kuin seuraava lapsielementti alkaa, joten millään lapsielementillä ei ole omaa lapsielementtiä. Kaikki lapsielementit päättyvät hyvin muodostetun XML-dokumentin sääntöjen mukaisesti sen vanhemman sisällä, jossa ovat alkaneetkin.

Validit XML-dokumentit

Dokumentit ovat *valideja XML-dokumentteja*, jos niihin sisältyy DTD, ja jos dokumentit ovat tämän DTD:n mukaisia. DTD:ssä määritellään dokumenteissa vaadittavat ja mahdolliset elementit, niiden väliset suhteet ja niihin liittyvät attribuutit. DTD:n avulla varmistetaan tunnisteiden tarkoituksenmukainen käyttö. Jos dokumentit ovat valideja, niin silloin ne ovat myös hyvin muodostettuja. (Bray ym., 1997.)

3.2 XLink-linkkistandardi

XLink-standardin mukaisilla linkeillä kuvataan dokumenttien tai niiden osien välisiä suhteita. Linkkistandardin kehittelyn lähtökohtana ovat olleet monipuoliset kaksisuuntaiset ja tyypitellyt linkit, joiden käyttäytyminen voidaan määritellä monipuolisesti. (Bray & DeRose, 1997.)

XLink-linkkistandardin taustalla on kolme muuta standardia joiden ominaisuuksia käsitellään seuraavaksi vain niiltä osin kuin niitä on hyödynnetty XLink-linkkistandardissa:

1. **URL** (Uniform Resource Locator) on standardi, jolla määritellään dokumenttien sijainti verkossa. XML-dokumenteissa viitatun dokumentin sijainti osoitetaan HREF-attribuutilla. Se voi saada arvokseen joko URL-standardin mukaisen osoitteen tai TEI-hankkeen mukaisen, tietyn kohdan linkittävän XPointerin, tai molemmat. Kun attribuutti saa arvokseen URL-osoitteen, dokumentit on linkkielementin sisältävän dokumentin ulkopuolella. Jos URL-osoite puuttuu, viitattu dokumentit on samassa dokumenteissa kuin linkkielementti. Tiettyyn kohtaan viitatessa, liitetään dokumentin osoitteeseen kohteen, ankkurin, nimi. Kohteen nimen on dokumentin tuottaja määrittänyt. Kohteiden nimeäminen on edellytys sille, että viittaus voidaan tehdä tiettyyn kohtaan dokumenteissa. (Berner-Lee, Masinter & McCahill, 1994; ks. myös Baker, 1995.)

2. **TEI** (Text Encoding Initiative) on kansainvälinen hanke humanististen tieteiden tiedonvälityksen ja jakelun kehittämiseksi määrittelemällä erityisen TEI DTD:n (Sperberg-McQueen & Plotkin, 1997; TEI 1997). XLink-linkkistandardi on hyödyntänyt hankkeesta tavan kohdistaa linkki rakenteisissa dokumenteissa. TEI:n mukaisen XPointerin avulla tietyyn kohtaan dokumentissa voidaan kohdistaa linkki. Dokumenttien tuottajien ei tarvitse nimetä etukäteen dokumenttien kohtia mahdollisia viittauksia varten. XPointer käyttää hyväkseen viittausten kohdistamisessa XML-dokumenttien rakennetietoa. (Bray & DeRose, 1997.)

XPointer muodostuu erilaisista avainsanojen yhdistelmistä. Näillä avainsanoilla osoitetaan dokumenteissa viitatun kohteen absoluuttinen sijainti tai sijainti suhteessa dokumentin muihin osiin tai ne osoittavat sanahakua. Avainsanoja ovat mm. ROOT, HERE, DITTO, ID (absoluuttista sijaintia osoittavat avainsanat), CHILD, ANCESTOR, PSIBLING (suhteellista sijaintia osoittavia avainsanoja), STRING (sanahakua osoittava avainsana). Näitä avainsanoja voidaan tar-

kentaa mm. esiintymän numerolla, elementtityypillä, attribuutin nimellä tai arvolla. Esimerkiksi viittaus CHILD (2, CHAP) (4, SEC) (3) linkittää dokumentin toisen luvun (CHAP) neljännen kappaleen (SEC) kolmanteen lapsielementtiin. (Bray & DeRose, 1997.)

3. **HyTime** (Hypermedia/Time-based Structuring Language) on standardoitu tapa määritellä hypertekstilinkit rakenteisissa dokumenteissa (Rutledge, 1996). XLink-spesifikaatiossa sanotaan (Bray & DeRose, 1997), että HyTimen perusteella määritellään XLink-linkkistandardissa tapoja paikallistaa erilaisia dokumentteja verkossa. Sen lisäksi sillä määritellään linkkien erilaisia semanttisia ominaisuuksia, kuten esimerkiksi kuinka linkitetty resurssi esitetään. Kipp (1997, 5) kuitenkin osoittaa artikkelissaan, ettei HyTimea ole ainakaan vielä hyödynnetty linkkistandardissa; HyTime-standardin nimi on vain mainittu.

XML-dokumenteissa linkit ilmaistaan linkkielementillä. Linkkielementtinä voi toimia mikä tahansa elementti. Linkkielementti erotetaan tavallisesta elementistä linkkielementtiin liitettävällä attribuutilla XML-LINK. Sen arvolla osoitetaan, millainen linkki on kyseessä. XML-LINK -attribuutin arvo voidaan kiinnittää tiettyyn elementtiin DTD:ssä, jolloin varsinaisissa dokumenteissa voidaan käyttää kyseistä elementtiä linkkielementtinä. Muut linkkielementtiin liitettävät tiedot määritellään dokumenteissa. Linkit tallennetaan joko linkkielementin sisältävään dokumenttiin, erilliseen dokumenttiin tai mahdollisesti tietokantaan. (Bray & DeRose, 1997.)

Linkkielementtiin voidaan (ja joissakin tapauksissa pitää) liittää seuraavat tiedot (Bray & DeRose, 1997):

- linkin rooli
- otsikko
- resurssin sijainti
- käyttäytyminen
- onko kyseessä in-line- vai out-of-line -linkki

Käyttäjä voi määritellä linkille roolin, jolla kuvataan linkin tarkoitusta. Tarkoituksena voi olla esimerkiksi yhdistää dokumentteihin kritiikkiä, taustatietoa tai

lisätietoa vaikkapa kirjoittajasta tai dokumentin versiosta. Tarkoituksena voi olla myöskin erilaisten apu- ja navigointivälineiden, kuten indeksien, sanastojen tai tiivistelmien liittäminen dokumentteihin. Samalla attribuutilla, jolla määritellään linkin rooli, voidaan määritellä myös linkin viittaaman dokumentin rooli. Linkin osoittamia dokumentteja voidaan kuvata linkissä vapaamuotoisilla otsikoilla. (Bray & DeRose, 1997.)

XML-dokumenteissa olevat linkit ovat yksinkertaisia (simple) silloin, kun ne viittaavat yksisuuntaisesti yhteen dokumentteihin. Linkit, jotka on määritelty attribuutin arvolla extended, yhdistävät monta dokumenttia. Monta dokumenttia linkitetään joko yksisuuntaisesti tai kaksisuuntaisesti. Yksisuuntaisissa linkeissä on viitatun dokumentin tiedot tallennettu linkkielementin sisältöön. Kaksisuuntaisia linkkejä ei ole välttämättä tallennettu mihinkään dokumenttiin. Niistä on muodostettu oma tiedostonsa tai ne voidaan tallentaa tietokantaan. Linkkien kokoaminen yhteen ryhmäksi helpottaa niiden hallintaa ja löytämistä. (Bray & DeRose, 1997.)

Linkin käyttäytyminen voidaan määritellä yleisellä tasolla tai hyvin yksityiskohtaisesti. Yleisellä tasolla linkin käyttäytyminen määritellään silloin, kun päätetään esitetäänkö viitatut dokumentit automaattisesti vai vain käyttäjän pyynnöstä. Viitatut dokumentit esitetään automaattisesti, kun linkki kohdataan dokumenteissa. Ei-automaattinen dokumenttien esitys vaatii käyttäjän toimia linkissä, esimerkiksi hiiren painallusta. Yksityiskohtaista linkin käyttäytymisen määrittelyä ei linkkispesifikaatio anna. (Bray & DeRose, 1997.)

Viitatun dokumentin sijoittamiseksi on kolme erilaista tapaa. Viitattu dokumentti voidaan upottaa esittämistä tai käsittelyä varten siihen dokumentin kohtaan, mistä viittaus alkaa. Toisaalta viitattu dokumentti voi korvata dokumentin siitä kohdasta alkaen, mistä viittaus alkaa. Kolmas tapa on esittää tai käsitellä viitattu dokumentti kokonaan uudessa yhteydessä, täysin erillään siitä, mistä viittaus alkaa. (Bray & DeRose, 1997.)

INLINE-attribuutilla määritetään, onko linkki tallennettu linkkielementin sisältöön vai ei. OUT-OF-LINE-linkit tallennetaan johonkin dokumentin ulkopuolelle. (Bray & DeRose, 1997.)

3.3 XSL-tyylistandardi

XSL-tyylistandardilla saadaan dokumenttien elementeille esitystapa. Se on tyylistandardi, joka on kehitetty erityisesti dokumenttien verkossa esittämistä varten. Tyylistandardin kehitystyö on vielä kesken. Täsmällisesti ei vielä tiedetä, mitä kaikkia ominaisuuksia siihen tulee sisältymään. XSL-tyylistandardi perustuu lähinnä SGML-dokumenttien esittämiseen tarkoitettuun tyylikieleen DSSSL:ään (Document Style Semantics and Specification Language) (ISO/IEC 1996; ks. myös DSSSL Online 1996; Prescod 1997), HTML-dokumenttien esittämiseen kehitettyyn CSS-kieleen (Cascading Style Sheets) ja erityisesti CSS:n laajennuksiin (Lie, 1997; Stevahn, 1997; Lie & Bos, 1996). XSL-tyylistandardissa on siis ominaisuuksia sekä CSS:stä että DSSSL:stä. (A Proposal, 1997.)

CSS-kieli on tarkoitettu erityisesti näytöllä esitettävien dokumenttien ulkoasun määrittelyyn. Sen perusominaisuuksissa onkin huomioitu elektronisen julkaisemisen erityispiirteet, kuten mm. vierityspalkit ja linkit. Se on kehitetty lähinnä HTML-dokumenttien esittämiseksi. Siinä ei ole niin paljon erilaisia muotoilupiirteitä, että se voisi hyödyntää rakenteisten dokumenttien tarjoamat mahdollisuudet. (Lie, 1997; Stevahn, 1997; Lie & Bos, 1996.)

CSS:ssä on 35 dokumenttien muotoilupiirrettä. Ne on ryhmitelty viideksi kokonaisuudeksi (kirjasin, väri ja tausta, teksti, laatikko, luokittelu). Niillä voidaan määritellä tarkasti sivujen esitystapa elementtien sijoittelun ja visuaalisuuden osalta. Muotoilupiirteet voivat periytyä elementeissä vanhemmalta lapselle. Erilaisia tyylimäärityksiä (dokumenttien tuottajan, käyttäjän) yhdistetään niin, että mahdolliset yhteentörmäykset erilaisten tyylimääritysten välillä vältetään. (A Proposal, 1997; Lie, 1997; Stevahn, 1997; Lie & Bos, 1996.)

CSS:n kyselykieli on alkeellinen. Kyselyssä elementit etsitään käsittelyä varten niitä edeltävien elementtien avulla. CSS-tyylikielellä voidaan vaikuttaa myös tulostamiseen siten että sivunvaihto, dokumenttien tulostettava alue sekä se, mikä alue dokumenteista tulostetaan milläkin medialla, määritellään tyyleillä. CSS-tyylikieli saadaan käyttöön viittaamalla suoraan tyylimäärityksiin varsinaisesta dokumentista joko LINK- tai STYLE-elementillä. CSS-tyylikieli on riittävä määrittämään yksinkertaisten XML-dokumenttien ulkoasun. Vaativampien rakenteisten dokumenttien muunnos- ja muotoilutehtäviin tarkoituksenmukaisempi kieli on XSL-tyylikieli. (A Proposal, 1997; Lie, 1997; Stevahn, 1997; Lie & Bos, 1996.)

DSSSL on monipuolinen dokumenttien esitystavan standardoitu määrittelykieli. Se on kehitetty erityisesti rakenteisten dokumenttien esitystavan määrittelyä varten. Siinä on ominaisuuksia sekä näytöllä esitettävien dokumenttien (mm. vierityspalkit, esitystila, linkit ja reunahuomautukset) että muulla tavoin tulostettavien dokumenttien esitystavan määrittelyyn. Siinä on kuitenkin niin paljon erilaisia muotoilupiirteitä että sen käyttö on monimutkaista. (ISO/IEC 1996; ks. myös DSSSL Online 1996; Prescod 1997.)

DSSSL:n kaksi yleisimpää dokumenttien käsittelytapaa ovat muunnos- ja muotoilukäsittelyt. Näiden lisäksi se määrittelee dokumenttien rakenteeseen kohdistuvan kyselykielen, jolla voidaan tehdä kyselyjä ja hakuja dokumenttien elementtien käsittelyä varten. Muunnoskäsittelyt tehdään muunnoskielen avulla. Dokumentit noudattavat muunnoskäsittelyn jälkeen jotain toista rakennemäärittelyä kuin mitä alkuperäiset dokumentit noudattivat. Muunnoskäsittelyjä käytetään mm. dokumenttien yhdistämiseen, indeksien ja sisällysluetteloiden luomiseen sekä muihin vastaaviin prosesseihin, joissa hyödynnetään dokumenttien rakenteisesti tallennettua tietoa. (ISO/IEC 1996; ks. myös DSSSL Online 1996; Prescod 1997.)

Osa muunnoskäsittelyn tehtävistä voidaan tehdä muotoilukäsittelyssä tyylikielellä, esim. sisällysluettelot. DSSSL:ssä on 300 muotoilupiirrettä ja 70 muotoilu-

luokkaa. Luokkia on mahdollisuus muotoilla, tai määrittää omia muotoiluluokkia ja -piirteitä. Muotoilupiirteet periytyvät vanhemmalta lapselle. Erilaisia tyylimäärityksiä on myös mahdollisuus yhdistää. Tyylimääritykset tehdään erillään varsinaisista dokumenteista. Ne liitetään dokumentteihin viittaamalla dokumenttien rakennemäärittelyssä haluttuun tyylimääritykseen. (ISO/IEC 1996; ks. myös DSSSL Online 1996; Prescod 1997.)

CSS:n ja DSSSL:n ominaisuuksia hyödyntäen kehitetään helppokäyttöistä kieltä rakenteisten dokumenttien verkossa esittämiseen eli XSL-tyylistandardia. XSL-tyylimääritykset ilmaistaan dokumenteissa XML-syntaksilla. XSL:ssä esitystavan määrittely tapahtuu kahden säännön, *construction rules* ja *style rules*, avulla. Construction rules vastaa lähinnä DSSSL:n muunnoskäsittelyjä, kun taas style rules määrittelee muotoilukäsittelyn. Kumpikin sääntö muodostuu kahdesta osasta: käsittelyn kohteeksi tulevan elementin tunnistamisesta ja esitystavan määrittelystä. Esitystavan määrittelyt merkataan XSL-tyylidokumentteihin. (A Proposal, 1997.)

XSL on tarkoitettu yleisten tyylimääritysten lisäksi vaativiin ja hienostuneisiin tyylimäärityksiin ilman minkään ohjelmointikielen käyttöä. Sen ominaisuuksiin kuuluu myös tyylimääritysten laajennettavuus. Dokumenttien käyttäjät voivat liittää dokumentteihin omat tyylimäärittelynsä. Tyylikielen on tarkoitus olla selkeästi luettava kieli. (A Proposal, 1997.)

3.4 SGML, HTML ja XML

XML-standardi on perusominaisuuksiltaan samanlainen kuin SGML. XML:lle ja HTML:lle puolestaan on yhteistä se, että ne on molemmat kehitetty verkkokäyttöön. Mitä uutta XML-standardi tuo suhteessa näihin standardeihin?

3.4.1 XML ja SGML

XML on SGML:n osajoukko, joten XML:ssä on paljon SGML:n ominaisuuksia. Lander (1997) on listannut seuraavat yhteiset ominaisuudet:

1) tieto tallennetaan rakenteisessa muodossa
2) elementtien hierarkian merkkaaminen edellyttää täsmällisyyttä
3) tiedon uudelleenkäytettävyys
4) tiedon modularisointi eli jakaminen pienempiin osiin.

XML:stä puuttuu suuri määrä erilaisia SGML:n ominaisuuksia. XML-spesifikaatio koostuu noin 30 sivusta, kun SGML-spesifikaatiossa on noin 500 sivua. SGML-dokumenttien jako verkossa on hankalaa myös monien erilaisten valinnaisten ominaisuuksien vuoksi. Se on hankala verkkokäytössä senkin vuoksi, että dokumentteihin täytyy liittää sekä tyylimäärittelyt että DTD. Ilman niitä SGML-dokumentteja ei voi käsitellä eikä näyttää verkossa. XML-dokumentteja voidaan jakaa verkossa myös ilman DTD:tä. Esittämistä varten XML-dokumentitkin vaativat tyylimäärittelyt. Selaimet voivat tulkita XML-dokumentteja, joista puuttuu DTD, jos XML-dokumentit ovat hyvin muodostettuja. (Lander, 1997.)

XML-standardia tukeva monipuolinen XLink-linkkistandardi tuo verkkokäyttöön sellaisia mahdollisuuksia, joita SGML:ssä ei suoranaisesti ole, mm. mahdollisuuden tallentaa linkit erillisinä tiedostoina (Bray & DeRose, 1997). XML- ja SGML-standardien välillä on myös se ero, että XML käyttää UNICODE-standardin mukaista merkistöä, kun SGML:ssä merkistöä ei ole kiinnitetty. SGML:ssä merkistö joudutaan määrittelemään kunkin sovelluksen yhteydessä, ja huomioimaan tällöin myös entiteettien merkistöt. UNICODE-standardi tukee kaikkia yleisimpiä maailman kieliä. (Goldfarb, 1995; Bray ym., 1997.)

XML:ssä ei ole kaikkia niitä ominaisuuksia rakenteen määrittelyyn kuin mitä SGML:ssä on. Bray ym. (1997) ovat luetelleet 33 erilaista SGML:n ominaisuutta, jotka puuttuvat XML:stä. XML:stä on jätetty pois minimointikeinoja, esimer-

kiksi tunnisteiden nimien minimointimahdollisuus. XML:ssä täytyy tyhjä elementti merkata määritetyllä tavalla, `<tyhjä></tyhjä>` tai `<tyhjä/>`, kun SGML:ssä tyhjää elementtiä voi osoittaa pelkkä alkutunniste. XML:stä puuttuvat mm. mahdollisuudet inkluusioon ja ekskluusioon. XML:ssä ei voi käyttää AND:iä `(&)` osoittamaan, että elementtien esitysjärjestys voi olla mikä tahansa. (Bray, 1997.)

Tässä vain muutama esimerkki XML:n ja SGML:n rakenteen määrittelyeroista. Määrittelyerot ja pienempi määrä erilaisia ominaisuuksia johtuvat siitä, että XML-sovellusten rakentaminen on haluttu tehdä helpommaksi kuin SGML-sovellusten rakentaminen. XML:n verkkosopivuutta korostavat sitä tukevat linkki- ja tyylistandardit sekä UNICODEen kiinnitetty merkistö. Molemmille standardeille on kuitenkin yhteistä niiden ajattomuus eli laitteisto-ja ohjelmistoriippumattomuus. (Bray, 1997.)

3.4.2 XML ja HTML

HTML ei ole XML:n taustalla oleva standardi. Se on tällä hetkellä laajasti käytetty merkkauskieli verkossa. XML on kehitetty erityisesti verkkokäyttöön. HTML ja XML eroavat toisistaan muutamalla olennaisella tavalla.

HTML-kieltä ei ole kehitetty määrittelemään dokumenttien rakennetta, kuten XML on. HTML:lle on määritelty DTD:t, mutta selaajat eivät tarkista, onko HTML-sivu DTD:n mukainen. HTML-sivujen ei tarvitse olla valideja eikä hyvin muodostettuja. HTML:n DTD:t määrittelevät käytettäväksi tietyn määrän elementtejä, joilla käytännössä määritellään sivun esitystapa. HTML:ää ei voi siis laajentaa dokumenttien vaatimilla elementeillä. Sillä ei voi vapaasti määritellä dokumenttien rakenneosia. (Lander, 1997.)

HTML-sivut esitetään yleensä selaimeen tallennettujen tyylimääritysten avulla. Ne on helppo ladata työasemalle palvelimelta, katsella ja yhdistää linkeillä toisiin dokumentteihin. HTML:ää on käytettykin lähinnä sivujen ulkoasun määrittelyyn. (Lander, 1997.)

XML tuo siis HTML:ään nähden dokumenttien rakenteisuuden ja laajennettavuuden. Rakenteisuuden ansiosta dokumentit ovat monipuolisesti uudelleenkäytettävissä. XML-dokumenttien rakenteisuus mahdollistaa myös tehokkaiden hakumenetelmien kehittämisen. Laajennettavuuden ansiosta dokumenttien rakenne voidaan määritellä tarpeen mukaisesti eikä olla sidottu vain tiettyihin elementteihin. Yhteistä XML:lle ja HTML:lle on tyylimäärittelyt dokumenttien esittämiseksi verkossa sekä- mahdollisuus yhdistää linkeillä dokumentteihin muita dokumentteja tai niiden osia. Sekä tyylimäärittelyt että linkkimahdollisuudet ovat kuitenkin XML:ssä monipuolisemmat kuin HTML:ssä, ja niiden käyttö on erilaista.

3.5 Yhteenveto XML-standardiperheestä ja kehitysnäkymiä

XML-standardiperhettä kehitetään verkkojakeluun sopivaksi standardiksi. Sen vuoksi siihen kuuluu varsinaisen standardin lisäksi myös erillinen linkki- ja tyylistandardi. XML perustuu SGML-standardiin. Tämä tarkoittaa sitä, että XML:ään on otettu sellaisia piirteitä SGML:stä, jotka tekevät siitä joustavasti käytettävän standardin verkossa, kuten rakenteinen tallennustapa. Siitä on jätetty pois niitä SGML:n piirteitä, jotka tekevät verkkokäytön hankalaksi. XML:n monipuolisuutta ja ajattomuutta ilmentää myös se, että se on laitteisto- ja ohjelmistoriippumaton standardi. Standardi käyttää merkkien koodauksessa UNICODE-standardia. UNICODE-standardi tukee kaikkia maailman yleisimpiä kieliä. Se helpottaa myös XML:n käyttöä, koska merkistöä ei tarvitse sovellusten yhteydessä kiinnittää.

XML-standardilla tallennetaan dokumenttien sisältö rakenteisesti. Rakenteinen tallennus takaa sen, että dokumenttien tietoja voidaan käyttää uudelleen erilaisissa yhteyksissä. Dokumentit voivat koostua itsenäisistä, erikseen tallennetuista, entiteeteistä. XML-prosessori yhdistää ne yhtenäiseksi dokumentiksi. Ennen kuin dokumentteja voidaan sanoa XML-dokumenteiksi, niiden on oltava joko hyvin muodostettuja tai valideja. Hyvin muodostettuja dokumentteja on helppo käsitellä ja siirtää verkossa, koska niihin ei välttämättä sisälly rakennemääritte-

lyä, DTD:tä. Toisaalta myös validilla dokumentilla on omat etunsa tiedonvälityksessä. Valideihin dokumentteihin sisältyvän DTD:n avulla varmistutaan, että dokumentit kuuluvat määriteltyyn dokumentitluokkaan, sekä ohjataan dokumenttien rakenteen suunnittelua.

XLink-linkkistandardi tuo verkossa olevien dokumenttien tai niiden osien yhdistämiseksi yksinkertaisten ja yksisuuntaisten linkkien rinnalle kaksisuuntaiset tyypitellyt linkit. Linkkien käyttäytyminenkin on määriteltävissä. Kaksisuuntaisten linkkien hallinta helpottuu, kun linkit tallennetaan erilliseen tiedostoon. Linkkistandardi tuo myös uusia tapoja määritellä tietty kohta dokumentissa, johon linkillä halutaan viitata.

Tyylistandardin avulla saadaan XML-dokumentit esitettävään muotoon joko näytölle tai muulle medialle. Tyylistandardilla voidaan dokumenteille määritellä monia erilaisia esitystapoja riippumatta alkuperäisestä esitystavasta. Tyylistandardiin sisältyvän kyselykielen avulla voidaan esitystapamäärittely kohdentaa vain tiettyyn tai tiettyihin elementteihin. Elementeille voidaan määritellä esitysmediasta riippuva esitystapa.

XLink- ja XSL-standardien kehitystyö on vielä kesken, joten kaikkia niiden ominaisuuksia ja mahdollisuuksia ei vielä tiedetä. XLink-linkkistandardin spesifikaation uusin versio ilmestyi tätä työtä viimeisteltäessä (Maler & DeRose, 1998). XSL-tyylistandardin spesifikaation uusin versio on tulossa.

XML:n kehitysnäkymiä

XML-standardin käyttökelpoisuuden laajentamiseksi on W3C:ssä kehiteillä tapa viitata, Namespaces in XML, dokumentin tai sen DTD:n ulkopuolella määriteltyjen elementtien nimiin. Tällöin dokumenteissa käytetävät elementit vastaavat joissakin muissa dokumenteissa olevia saman nimisiä elementtejä. Erilaisista dokumenteista tulee käytettyjen elementtien suhteen yhdenmukaisia. (Bray, Hollander & Layman, 1998.)

XML-dokumenttien rakenteen määrittelemiseksi on erillinen työryhmä kehittänyt W3C:n arvioitavaksi XML-Data skeeman. Se on dokumenttien elementtien, attribuuttien ja entiteettien sekä niiden välisten suhteiden kuvaus ja määrittely. Se eroaa DTD:stä, koska dokumenttien rakenne voidaan kuvata sillä monipuolisemmin kuin DTD:llä. Sen lisäksi XML-Data skeema kirjoitetaan XML-syntaksilla, joten DTD:n syntaksia ei tarvitse osata. XML-Data skeema on vielä kehitteillä oleva XML-standardin täydennys. Sen ensimmäinen työpaperi ilmestyi joulukuussa 1997, joten sen tarjoamat mahdollisuudet voivat vielä muuttua ja täsmentyä. (Layman ym., 1997.)

XML-dokumenttien esitystavan monipuolistamiseksi standardoidulla tavalla on W3C:ssä kehitetty Document Object Model (Byrne, 1997). XML-dokumenttien yksittäisille elementeille voidaan Document Object Modelin, DOM, avulla määritellä mm. käyttäytyminen, kuinka esitystapa huomioidaan elementissä, kuinka elementin tyylit vaihtuvat, millaisessa kanssakäymisessä elementit ovat keskenään, vaihtaa elementin sisältöä, lisätä, poistaa tai vaihtaa elementin attribuutteja. XML-dokumenttien lisäksi DOMilla voidaan määritellä HTML-dokumenttien esitystapaa.

XML:n kehitysnäkymät liittyvät siis ensiksikin dokumenttien rakenteen helpompaan ja monipuolisempaan määrittelyyn XML-Data skeeman avulla. Toiseksi XML-dokumenteille haetaan yhdenmukaisuutta elementtien Namespaces-viittaustekniikan aulla. Verkon mahdollisuuksien parempaan hyödyntämiseen viittaa DOMin kehittäminen. Samalla kun XML-standardiperhe kehittyy monipuolistuu myös verkon käyttö.

4 XML JA VERKKOSOVELLUKSIA

Tässä luvussa esittelen, millaisia sovelluksia XML:n avulla voidaan rakentaa. Aluksi tarkastelen yleisesti XML-sovelluksia. Sen jälkeen kohdistan sovellusten tarkastelun tämän työn aihepiiriin eli arkistoinnin kannalta olennaisiin hankkeisiin. Näissä hankkeissa tarkoitus on kehittää menetelmiä metatiedon liittämiseksi dokumentteihin.

4.1 Erilaisia XML-verkkosovelluksia

XML-standardoidun tiedon avulla on mahdollista rakentaa verkkoon erilaisia sovelluksia. Bosak (1997) ryhmittelee erilaiset XML -verkkosovellukset neljään ryhmään:

1) tiedon siirtoon ja välitykseen erikoistuneet sovellukset,
2) sovellukset, joissa Java-ohjelmat käyttävät XML-tietoa,
3) sovellukset, jotka esittävät samasta lähteestä eri käyttäjille erilaisia näkymiä, ja
4) älykkäät verkkorobotit, jotka hakevat henkilökohtaisten toiveiden mukaista tietoa

Seuraavassa tarkastellaan hieman tarkemmin näitä erilaisia verkkosovelluksia.

1) Tiedon siirtoon ja välitykseen erikoistuneet sovellukset käyttävät hyväkseen sovelluskohtaisia dokumenttien rakennemäärittelyjä. Standardoitu tai yleisesti hyväksytty rakennemäärittely helpottaa ja yksinkertaistaa tiedon siirtoa, kun vastaanottajalla voidaan olettaa olevan käytössä sama rakennemäärittely. Sovellusalakohtaisia rakennemäärittelyjä voivat hyödyntää mm. saman alan yritykset ja tieteenalat. Tällaisten sovellusten käyttöalueita ovat mm.:

- julkaisutoiminta
- CAD/CAM -sovellukset
- kaikki verkossa olevat kaupalliset sovellukset, jotka käyttävät hyväkseen tietokantoja, kuten tilauslistat, varaston seuranta jne.

Yksi käytännön esimerkki tähän ryhmään kuuluvista sovelluksista on kemian käyttöön kehitetty CML-kieli. Sillä on oma DTD, ja sen avulla kemian alan yritykset ja tutkijat voivat vaihtaa standardoitua tietoa verkossa. (ks. mm. Garshol, 1997; Khare & Rifkin, 1997.)

2) Java-ohjelmilla voidaan esittää ja havainnollistaa erilaisia asioita XML-standardin mukaisesti tallennetusta tiedosta. Esimerkiksi CML-kielellä määritelty kemiallinen tieto, kuten aineen molekyylirakenne, voidaan esittää selaimessa Java-ohjelmalla kolmiulotteisena mallina. Tällaisia sovelluksia voidaan hyödyntää mm.:

- erilaisessa suunnittelutyössä, kuten puolijohteiden suunnittelussa, koneiden ja rakennusten yms. suunnittelussa,
- aikatauluissa,
- kaupallisissa sovelluksissa.

Tämän ryhmän sovellukset pyrkivät erityisesti siirtämään tiedon käsittelyn aiheuttamaa kuormitusta palvelimelta työasemalle. (Bosak, 1997.) Lander (1997) haluaakin korostaa XML:n verkkokäytettävyydessä sitä, että verkossa olevat dokumentit siirretään palvelimelta käyttäjän työasemalle käsiteltäväksi. Hänen mielestään tämä nopeuttaa työskentelyä ja dokumenttien käsittelyä sekä vähentää tietoliikenneyhteyksien kuormitusta.

3) Työasemalle latautuneesta tiedosta erilaisia näkymiä päivittävästä sovelluksesta on hyvä esimerkki dynaaminen sisällysluettelo. Käyttäjä saa siitä esiin hiiren napsautuksella erilaista tietoa, mm. alaotsikot tai alaviitteeseen tallennetut tiedot näkyviin. Näkymien esittämiseen voidaan käyttää Java-sovelmia tai JavaHelp -luokkakirjastoja. (ks. myös Garshol, 1997.)

4) Älykkäät hakurobotit hyödyntävät dokumenttien rakenteisesti tallennettua tietoa. Ne etsivät käyttäjän tarpeisiin räätälöityä tietoa, Tällaiset robotit ovat Bosakin visioita XML:n tarjoamista mahdollisuuksista verkkomaailmassa. Bosak pitääkin pitkälle kehitettyjä hakurobotteja tulevaisuuden visioina. Garshol (1997) näkee, että rakenteisesti tallennettu XML-standardoitu tieto mah-

dollistaa tehokkaampien hakumenetelmien kehittämisen kuin mitä nykyiset hakumenetelmät ovat.

Yksi esimerkki XML-sovelluksesta verkossa on W3C:n kehittämä SMIL (Synchronized Multimedia Integration Language). Sen ensimmäinen spesifikaatio julkaistiin arvioitavaksi marraskuussa 1997. SMIL-kieli mahdollistaa toisistaan riippumattomien multimediaobjektien yhdistämisen synkronoidusti multimediaesitykseen. Verkkoon saadaan ilman hankalia ohjelmointikieliä monipuolisia multimediaesityksiä, joita käyttäjä voi kontrolloida, ja joihin voidaan liittää hyperlinkkejä. (Hoschka, 1997.) Toinen esimerkki XML:n hyödyntämisestä on W3C:n kehittämä PGML (Precision Graphics Markup Language). Se on kehitetty vektorigrafiikan ja graafisen suunnittelun tarpeisiin, jotta graafiset kuvat välittyisivät käyttäjälle sellaisina kuin niiden tuottaja on ne suunnitellut. PGML:nkin kehitystyö on vielä kesken. (Al-Shamma ym., 1998.)

Vaikka XML-standardiperheen kehitystyö ei vielä olekaan valmis, sen pohjalta on jo kehitetty alustavia sovelluksia. Näiden sovellusten syntyminen ilmentää hyvin sitä, että verkon mahdollisuuksia voidaan hyödyntää monipuolisesti, kun siihen saadaan sopivat välineet.

4.2 Metatietoon liittyviä XML-verkkosovelluksia

Metatieto on tietoa tiedosta, ja sen avulla tehostetaan dokumenttien löytymistä. Verkkosovellusten yhteydessä metatiedolla tarkoitetaan täsmällisemmin tietoa verkossa olevista tiedoista. Metatietoon liittyvien hankkeiden tavoitteena on yleinen sanasto dokumenttien metatiedon kuvaamiseksi ja malli metatiedon liittämiseksi dokumentteihin. Dokumentteihin liitettyä metatietoa hyödynnetään sekä hakumenetelmien ja dokumenttien hallinnan että erilaisten tietoa siirtävien ja vaihtavien sovellusten kehittämisessä. (Lassila, 1997.)

Erilaisten metatiedon määrittelemiseen liittyvien hankkeiden tavoitteena on menetelmä, joka antaa dokumenteista riippumatta tietoa siitä, mille sovellusalueelle dokumentit liittyvät tai mikä niiden sisältö on. Menetelmän avulla

pitäisi pystyä kuvaamaan mitä tahansa tietoa. RDF (Resource Description Framework) on tähän tarkoitukseen kehitteillä oleva menetelmä. Sitä kehitettäessä on huomioitu aikaisempien hankkeiden, PICS-NG:n (Platform for Internet Content Selection for Next Generation), Meta Content Framework:n ja Web Collectionin, ominaisuuksia. (Lassila, 1997.)

Kaikki nämä hankkeet on tuotu julkisuuteen vuonna 1997, mutta tarkempaa tietoa siitä, mitä hanketta tai hankkeita kehitetään edelleen, ei tällä hetkellä ole. Nämä kaikki hankkeet perustuvat tiedon tallennukseen rakenteisessa muodossa. Näissä kaikissa XML on yksi mahdollinen syntaksi metatiedon merkkaamiseksi.

Web Collectionns, PISC-NG ja Meta Content Framework

Web Collections ja PISC-NG perustuvat erilaisten ominaisuusluokkien käyttämiseen dokumenttien metatiedon kuvaamisessa. Jokainen luokka sisältää sille luokalle tyypillisiä ominaisuustyyppejä. Luokka tai luokat valitaan kuvattavan dokumenttien ominaisuuksien perusteella. Esimerkiksi `www-sivun` metatiedon kuvaamiseen voitaisiin käyttää luokkaa, jonka ominaisuustyyppinä olisivat `sivun tuottaja`, `sivun koko`, `päivityspäivämäärä` ja `sivun tuottajan sähköpostiosoite`. Koska PICS-NG:n taustalla on olio-ohjelmointi, on metatiedon määrittelyyn tullut piirteitä olio-ohjelmoinnista, esimerkiksi luokat voivat periytyä. (Hopmann, 1997; Lassila, 1997.)

Meta Content Frameworkin (MCF) perusajatus on, että dokumenteilla on ominaisuuksia, jotka jakautuvat erilaisiksi ominaisuustyypeiksi. Ominaisuustyyppien saamat arvot, kuten esimerkiksi sukunimi tai jokin luku, kuvaavat dokumentteja. Nämä ominaisuustyypit voivat itse joskus vielä jakautua ominaisuuksiksi, jotka puolestaan saavat dokumentteja kuvaavia arvoja. Esimerkiksi jos dokumentit olisi `www-sivu`, sen ominaisuuksia olisivat `koko`, `URL-osoite` ja `ylläpitäjä`. Ominaisuus `koko` voisi saada ominaisuustyypin `koko` `tavuina` ja arvoksi jonkin numerosarjan, joka kuvaisi tämän dokumenttien

kokoa. Toisaalta tämä ominaisuustyyppi, joka kertoo www-sivun koon, voitaisiin nimetä `sivunkoko`, jolloin se saisi ominaisuuksia, jotka 1) kertovat koon numeroina, 2) kertovat, että numerot ilmaisevat tavuja 3) kuvaustietoa ominaisuustyypistä, esimerkiksi miksi sivun koko vaihtuu päivittäin. MCF:n periaatteiden mukaisesti dokumenttien metatietokuvaus muodostetaan dokumenttien sisältöä kuvaavien ominaisuus-ominaisuustyyppi -ketjujen avulla. (Guha, R.V. & Bray, 1997; ks. myös Bray & Guha 1997.)

Resource Description Framework

Resource Description Framework (RDF) on kehitetty edellä kuvattujen metatietohankkeiden jälkeen. Julkisuuteen annettiin ensimmäinen versio RDF-mallin määrittelemiseksi lokakuussa 1997. RDF:n kehittämisryhmään kuuluvat Daniel, Ianella ja Miller (1997) kuvaavatkin ensimmäistä versiota lähinnä näytöksi siitä, mitä ryhmässä on ajateltu metatiedon merkitsemisestä. Ryhmä haluaakin ajatuksistaan käyttökelpoista palautetta. RDF:n yksi keskeinen tavoite on edesauttaa tiedon välityksen tehostumista verkossa kuvaamalla verkossa olevia dokumentteja täsmällisesti. (Lassila & Swick, 1997.)

RDF:n keskeinen ajatus on kuvata verkossa olevia dokumentteja dokumenttien ominaisuuksien ja niiden arvojen muodostamalla mallilla. Mallin tulisi soveltua kaikenlaisten dokumenttien kuvaukseen. Esimerkiksi dokumentti voisi olla `www-sivu`, ominaisuus olisi `www-sivun kirjoittaja` ja ominaisuuden arvo olisi `kirjoittajan nimi`. Malli saadaan tallennettavaan ja sovellusten välillä siirrettävään muotoon XML-standardin avulla. XML tarjoaa siis syntaksin, jolla määritetään RDF-syntaksi eli ne tavat, joilla dokumentteja kuvaava metatieto merkataan tallennettavaksi (serialization syntax). (Lassila & Swick, 1997.)

RDF:ään ei sisälly ennalta määriteltyä sanastoa metatiedon määrittelemiseksi kustakin dokumentista. RDF on tarkoitettu sellaiseksi syntaksiksi, jolla voidaan tallentaa erilaisten yhteisöjen määrittelemä metatietokuvaus dokumenteista. Esimerkiksi koirista kiinnostunut yhteisö määrittelee koiria käsittelevistä do-

kumenteista erilaiset ominaisuudet metatiedoksi kuin viineistä kiinnostunut yhteisö määrittelee viinejä käsittelevistä dokumenteista. Toisaalta näiden eri yhteisöjen dokumenttien ominaisuuksien kuvaukset voivat olla suhteessa toisiinsa siten, että esim. ominaisuudella `paino` tai `hinta` tarkoitetaan samaa asiaa kumassakin metatietokuvauksessa. Jokaisen yhteisön oma metatietokuvaus ja sen suhde muiden yhteisöjen metatietokuvauksiin on mahdollista toteuttaa RDF:llä. RDF:llä voi siis jokainen yhteisö määritellä omat ominaisuudet kuvaamaan dokumenttejaan, ja dokumentit säilyttävät siirrettävyytensä sovellusten välillä. (Lassila & Swick, 1997.)

RDF sopii kaikenlaisten dokumenttien metatiedon kuvaamiseen. Se on kehitetty täysin sovellusalueriippumattomaksi ja mahdollistamaan automaattisen dokumenttien käsittelyn. RDF-mallin mukainen metatieto voi hyödyntää:

1) hakumenetelmiä, koska dokumenteista on käytössä enemmän tietoa,
2) dokumenttien luettelointia, koska pystytään kuvaamaan verkossa olevien dokumenttien välisiä suhteita, esim. WWW-sivujen ja digitaalisten kirjastojen sisältöjä ja sisältöihin liittyviä suhteita,
3) tiedon siirtoa ja jakelua,
4) dokumenttien sisällön luokittelua,
5) tekijänoikeuksien kuvaamista.

RDF mahdollistaa digitaalisen allekirjoituksen, jolloin verkossa tapahtuva kaupankäynti, ryhmätyöteknologiat ja monet muut sovellukset muuttuvat luotettavammiksi ja turvallisemmiksi. (Lassila & Swick, 1997.)

Dublin Core-yhteisö on kokeillut RDF:ää. Kokeilun tekijät ovat kaikki mukana kehittämässä RDF:ää. Kokeilun tarkoituksena oli testata, kuinka hyvin Dublin Core-ominaisuudet (DC) sopivat RDF-syntaksiin, kun halutaan kuvata dokumenttien metatietoa. Tarkoitus oli myös antaa RDF:n kehittäjille palautetta DC:n kannalta. DC:ssä on 15 erilaista dokumenttia kuvaavaa ominaisuutta. Ominaisuuksien käyttö on tarkasti sisällöllisesti määritetty. Kokeilun tulos oli, että DC-kuvaus sopii nykyiseen RDF-syntaksiin silloin, kun dokumentista halutaan yksinkertainen kuvaus. Dokumentin kuvaaminen monipuolisesti DC:a käyttäen on RDF-syntaksilla hankalaa. (Daniel, Ianella & Miller, 1997.)

RDF:n kehitystyö on vielä kesken. Julkisesti RDF:stä on vasta esitelty malli, *resurssi-ominaisuus-ominaisuuden arvo*, jolla kuvataan dokumenttien metatietoa. Mallille on esitelty yksi mahdollisista syntakseista (XML), jolla voidaan määrittää ja välittää mallin mukaista metatietoa verkossa. RDF-standardista ei voi puhua vielä edes mallin osalta. Myöhemmin ilmestyy määrittelyt ainakin sille, kuinka RDF:ssä määritetään metatietoluokkia ja kuinka tehdään kyselyjä. RDF:n tulevaisuus näyttää siinä mielessä hyvältä, että sen kehitystyöllä on avainasemassa olevien organisaatioiden, kuten mm. Netscapen, Microsoftin, IBM:n, Nokian, OCLC:n tuki. (Lassila & Swick, 1997.)

5 ARKISTOINTI

Tässä luvussa keskityn XML-standardiperheen tarjoamiin haasteisiin ja mahdollisuuksiin dokumenttien arkistoinnissa. Aluksi tarkastelen arkistoinnin erityispiirteitä verkkoympäristössä. Sen jälkeen esittelen, mitä tietoa dokumenteista on arkistoitava, ja pohdin, mitä on metatieto. Lopuksi tarkastelen XML-stardiperhettä ja arkistointia.

5.1 Arkistointi verkkoympäristössä

Arkistoinnilla tarkoitetaan tässä työssä dokumenttien pitkäaikaista säilytystä digitaalisessa muodossa. Dokumentit on arkistoitava siten, että ne ovat tarvittaessa saatavilla, esitettävissä ja käytettävissä, nyt ja tulevaisuudessa alkuperäisessä muodossaan. Arkistoidut dokumentit löytyvät kuvailevan tiedon avulla. (Preserving Digital Information, 1996; Day, 1997). Uusi informaatioteknologia antaa kuitenkin dokumenteille piirteitä, jotka vaikuttavat arkistointiin verkkoympäristössä.

Arkistoitavat dokumentit voivat olla painetun dokumentin kaltaisia staattisia dokumentteja verkossa. Staattisten dokumenttien arkistointi on yksinkertaisempaa kuin *dynaamisten yhdistelmädokumenttien* arkistointi. Ne eivät vaadi ylläpitoa niin kuin yhdistelmädokumentit.

Yhdistelmädokumenttien sisältö koostuu itsenäisistä ja hajallaan olevista tietokokonaisuuksista, jotka on tallennettu eri paikkoihin. Sen lisäksi dokumentit saattavat koostua monesta osasta, koska dokumenttien sisältö, rakenne ja esitystapa on erotettu toisistaan. Dokumenttien koostuminen monesta eri osasta niin sisällöllisesti kuin ulkoisestikin lisää niiden muutosmahdollisuutta eli dynaamisuutta. Muutos koskee koko dokumenttia tai jotakin/joitakin sen osia. Dokumentit voivat aikojen kuluessa korvautua kokonaan tai osittain, niihin liitetään uusia asioita, tehdään korjauksia tai sisällytetään uusia itsenäisiä osia.

Sisällön lisäksi mm. dokumenttien otsikko, koko ja sijainti voivat muuttua. (Mackenzie Owen & Walle, 1996, 24, 66; Nielsen, 1997.)

Elektronisessa ympäristössä olevien dokumenttien arkistointia monimutkaistaa myös se, että ne voivat sisältää linkkejä. Linkkien arkistointi vaatii linkkien ylläpitoa, koska linkkien osoittamien dokumenttien sijainti saattaa muuttua hyvinkin usein. Linkkien osoittamat dokumentit ovat joskus sellaisia, että ne vaativat erilaisia ehtoja avautuakseen. Tällaisia ehtoja voivat olla esim. luvat, maksut, allekirjoitukset. Ehdot voivat muuttua dokumenttien elinkaaren aikana. Myös ehtojen vuoksi linkkit vaativat ylläpitoa. (Mackenzie Owen & Walle, 1996, 24, 66).

Dokumentteihin liittyy myös tekijänoikeustietoa. Dokumenttien tekijänoikeudet voivat olla monella eri taholla. Niitä voi olla mm. dokumenttien tuottajalla, dokumenttien arkistoijalla sekä yksilöillä ja erilaisilla yhteisöillä, jotka kokoavat tietoa. Tekijänoikeus määrittelee myös dokumenttien ja niiden eri osien käytettävyyttä. Se määrittelee, voiko dokumentteja esimerkiksi selata näytöllä, tulostaa paperille, tallentaa jollekin muulle medialle tai käyttää viittauksissa. Tekijänoikeudet asettavat ehtoja ja rajoituksia dokumenttien käytettävyydelle ja hyödynnettävyydelle. (Preserving Digital Information, 1996; Mackenzie Owen & Walle, 1996.)

Dokumenttien arkistointiin vaikuttava jatkuva muutos ilmenee myös tekniikassa. Dokumenttien arkistointiin liittyvässä tekniikassa on kaksi heikkoutta (Day, 1997):

1) Tallennusmedia, joka vanhentuu suhteellisen nopeasti - verrattuna paperiin tallennusmediana
2) Laitteistot ja ohjelmistot, joilla tuotettujen dokumenttien käyttö on usein niistä riippuvaa - laitteistot ja ohjelmistot kuitenkin häviävät ja muuttuvat nopeasti.

Näiden ongelmien voittaminen vaatii arkistolta dokumenttien jatkuvaa kopioimista uudelle tallennusmedialle tai muuntamista uusia laitteistoja ja ohjelmistoja vastaaviksi. Dokumenttien siirtäminen ja muuntaminen edellyttää tarkkaa

seurantaa, ettei dokumenteista katoa niiden aikana niiden sisältämää tietoa. (Day, 1997.)

Arkistointi verkkoympäristössä edellyttää muutoksen huomioimista. Muutos koskee sekä arkistoitavia dokumentteja että tekniikkaa, jolla dokumentit on tallennettu ja jolla ne ovat käytettävissä. Muutoksen vuoksi dokumentteja on kontrolloitava ja pidettävä yllä niiden säilyvyyttä ja toimivuutta. Muutoksen huomioimisen lisäksi dokumentit on arkistoitava niin, että tekijänoikeudet ja niistä aiheutuvat rajoitukset tulevat huomioiduiksi. Tämän kaiken lisäksi, dokumenttien on oltava löydettävissä arkistosta.

5.2 Dokumenttien arkistointi

Arkistoitavat dokumentit ovat verkkoympäristössä useinkin dynaamisia yhdistelmädokumentteja, jotka muuttuvat elinkaarensa aikana monin tavoin. Tästä aiheutuu se, että dokumenttien kokonaisuus on vaikeasti määriteltävissä. Preserving Digital Information -ryhmän (1996) mielestä dokumentit muodostuvat erilaisista asioista. Näiden asioiden tunnistaminen ja tallentaminen on keskeistä dokumenttien yksilöllisyyden ja yhtenäisyyden kannalta. He määrittelivät seuraavat asiat dokumenttien kokonaisuuden kannalta keskeisiksi:

- sisältö,
- varmistus aitoudesta,
- löydettävyys
- alkuperä ja
- asiayhteys.

Dokumenttien *sisällöllä* tarkoitetaan dokumenttien asiasisältöä. Arkistoinnissa on ratkaistava, miten dokumenttien sisältö saadaan asianmukaisesti arkistoitua niin ettei sisältö muutu. Yksinkertaisin tapa on tallentaa sisältö pelkkänä merkkijonona. Tällöin merkit voidaan aina tulkita, eikä dokumenttien sisältämä tieto ole laitteistoista tai ohjelmista riippuvainen. Toinen keino on tallennetaan sisältö sekä sisällön esitystapa ja rakenne. Esitystavan ja rakenteen tallentaminen on laitteisto- ja ohjelmistoriippuvaa, jolloin dokumenttien kokonaisuuden

säilyttäminen ja saatavilla pitäminen edellyttää ylläpitotyötä. Kolmas tapaa tallentaa dokumenttien sisältö on tallentaa dokumenttien sisältö rakenteisesti. Rakenteiset sisällöt tallennetaan merkkijonomuodossa, joten ne sopivat pitkäaikaiseen säilytyksen. Rakenteisessa tallennustavassa säilytetään dokumenttien looginen rakenne asiasisällön ohella. Rakenteisesti tallennetut sisällöt ovat käyttökelpoisia, koska niitä voidaan käytettää erilaisissa yhteyksissä ilman että alkuperäinen tieto muuttuu. (Preserving Digital Information, 1996.)

Dokumentteihin tallennetaan niiden *aitouden*, muuttumattomuuden, varmistavat menetelmät. Dokumenttien aitoudella tarkoitetaan sitä, että käyttäjä tietää, mikä painos tai versio dokumentti on, ja voi olla varma siitä. Varmistuskeinot ovat puolestaan sellaisia, että dokumenttien muuttaminen on estetty. Niillä, joilla on oikeus tehdä dokumentteihin muutoksia, on siihen myös välineet, kuten digitaalinen allekirjoitus. (Lynch, 1994; Preserving Digital Information, 1996; Mackenzie Owen & Walle, 1996.)

Dokumenttien *löydettävyyten* sisältyy sekä dokumenttien sijainti verkkoympäristössä että dokumentteja kuvaileva tieto. Dokumenttien sijainnin on oltava niin yksiselitteisiä ja selkeitä, että dokumentit löytyvät tulevaisuudessakin muiden dokumenttien joukosta. Verkossa voidaan määritellä dokumenttien sijainti dokumenttien osoitteen, URL:n avulla. Osoitteen huono puoli on, että se muuttuu aina silloin, kun dokumenttien sijaintia vaihdetaan koneesta toisen. Tähän ongelmaan on kehitteillä dokumenttien nimeen, URN, perustuva identifiointimenetelmä.

Dokumenttien löydettävyyteen liittyy tieto myös dokumenttien muista kopioista. Dokumentteihin onkin tallennettava tietoa siitä, mikä on dokumenttien alkuperäinen sijainti. Sen lisäksi dokumenteissa on oltava linkit muihin mahdollisiin verkossa oleviin kopioihin. Dokumentin sijainti kokoelmassa ja kokoelman sijainti on myös tallennettava. (Mackenzie Owen & Walle, 1996.)

Erilaisilla hakumenetelmillä, jotka perustuvat mm. bibliografioiden, luetteloiden ja indeksien hyväksikäyttöön, tekstidokumentit ovat nykyisin löydettävissä ilman erillistä dokumentteja kuvailevaa tietoakin. Dokumenttia kuvailevan tiedon liittämisellä varmistetaan muidenkin kuin tekstidokumenttien löytyminen ja hakumenetelmien kehittyminen. (Preserving Digital Information, 1996.)

Dokumenttien *alkuperän* antaa lisätietoa dokumenttien tuottamisesta sekä kertoo niistä toimenpiteistä, joita dokumenteille arkistossa on tehty ja tehdään. Dokumenttien alkuperään sisältyy tietoa siitä, kuka dokumentin on tehnyt, mikä rooli ja merkitys dokumentilla on tuottajalleen ollut, kuinka dokumentti on syntynyt, ja miksi dokumentti on tullut arkistoon. Dokumenttien arkistointiin liittyvä tieto on tietoa dokumentteihin tehdyistä muutoksista siirrettäessä dokumentteja tallennusmedialta toiselle. Se on myös tietoa dokumentteihin tehdyistä uuden teknologian edellyttämistä muutoksista. Näiden muutosten avulla dokumentit säilyttävät käytettävyytensä. Dokumenttien alkuperän arkistoiminen lisää arkiston luotettavuutta, koska alkuperän avulla dokumenttien aitous on tarkistettavissa. (Mackenzie Owen & Walle, 1996, 85; Preserving Digital Information, 1996 ; Wallace, 1996.)

Dokumenteilla on digitaalisissa ympäristöissä erilaisia *asiayhteyksiä.* Asiayhteyteen katsotaan kuuluvaksi tietoa dokumenttien teknisestä kontekstista, viittauksista, jakelumediasta ja laajemmasta sosiaalisesta kontekstista. Arkistoitaessa dokumentteja, joudutaankin arvioimaan, mitkä ulottuvuudet vaikuttavat dokumenttien kokonaisuuden säilymiseen. (Preserving Digital Information, 1996.)

Dokumenttien tekninen konteksti kertoo dokumenttien laitteisto- ja ohjelmistoriippuvuuden sekä dokumentin koosta. Dokumenttien käyttö saattaa vaatia tietyn ohjelmiston, joka puolestaan edellyttää laitteistolta joitakin asioita. Dokumenttien laitteisto- ja ohjelmistoriippuvuuden suhteen arkistoinnissa on kaksi toimintatapaa: joko tallentaa dokumentit laitteisto- ja ohjelmistoriippuvassa muodossa tai muuttaa sisältö sovituilla menetelmillä pitkäaikaiseen säilytykseen soveltuvaan muotoon. (Preserving Digital Information, 1996.)

Dokumenttien arkistointiin laitteisto- ja ohjelmistoriippuvina ehdottaa Rothenberg (1996) ratkaisuksi kapselointia. Sillä hän tarkoittaa sitä, että varsinaisen dokumentin lisäksi tallennetaan dokumentin käyttämistä edellyttävien ohjelmien rakentamisen kertovat dokumentit. Näin saadaan tulevaisuudessakin rakennettua dokumenttien käyttämistä vaativat ohjelmat.

Dokumenttien viittaukset ovat linkkejä dokumenttien sisällä tai eri dokumenttien välillä. Linkkeihin sisältyy jatkuva muutosmahdollisuus. Muutos koskee linkkien osoittamien dokumenttien sijainnin muuttumista sekä näiden dokumenttien käyttöön liittyvien ehtojen, kuten maksut ja allekirjoitukset, muuttumista. Linkkien tallennus on vaikeaa sen vuoksi, että linkit vaativat dokumenttien muutosten vuoksi jatkuvaa ylläpitoa. Linkkien ylläpitomenetelmien hankaluuden vuoksi arkistoinnissa onkin ratkaistava, ovatko dokumentit kokonaisuuksia, jos niiden sisältämien linkkien osoittamia dokumentteja ei tallenneta. (Mackenzie Owen & Walle, 1996, 24; Preserving Digital Information, 1996.)

Dokumenttien ominaisuudet ja luonne riippuvat niiden jakelumediasta. Verkkoympäristössä mm. käytettävissä oleva kaistanleveys ja tietoturvateknologian mahdollisuus määrittelevät dokumenttien ominaisuuksia verkkoympäristössä. Esimerkiksi kaistanleveyden laajentaminen mahdollistaisi koko näytön täyttävät videoesitykset, ja tietoturvateknologian kehittyminen luottamuksellisten dokumenttien jakamisen. Arkistoinnissa onkin tallennettava tietoa niistä jakelumedian ominaisuuksista, jotka määrittävät dokumenttien ominaisuuksia. (Preserving Digital Information, 1996.)

Dokumenttien laajempi sosiaalinen ympäristö vaikuttaa dokumenttien ymmärtämiseen. Esimerkiksi verkkoympäristössä olevien dokumenttien ominaisuudet riippuvat mm. kaistanleveydestä, tietoturvateknologiasta ja teknisestä infrastruktuurista, jotka puolestaan ovat seurausta erilaisista poliittisista, yhteiskunnallisista ja teknisistä päätöksistä. Dokumenteilla on myös erilaisia sosiaalisia merkityksiä. Ne on tuotettu epäviralliseen sosiaaliseen kanssakäymiseen, viralliseen asioiden hoitamiseen, viihdyttämiseen, tiedottamiseen jne. Doku-

menttien sosiaalinen konteksti muodostuu siis kaikista niistä ulottuvuuksista, jotka jotenkin ovat vaikuttaneet dokumentin muotoutumiseen juuri sellaiseksi kuin on. Näiden kaikkien ulottuvuuksien arkistoiminen on mahdotonta, joten arkistoitaessa joudutaankin päättämään, mitkä tiedot ovat olennaisia dokumenttien kokonaisuuden säilymisen tallettamiseksi. (Preserving Digital Information, 1996.)

Tässä listassa, jossa määritetään dokumenttien osatekijät, ei ole mainittu tekijänoikeuksia. Ne kuitenkin määrittävät ketkä saavat käyttä dokumentteja ja miten. (Preserving Digital Information, 1996; Mackenzie Owen & Walle, 1996.)

Preserving Digital Information -ryhmä on määritellyt, mistä asioista arkistoitava dokumentti koostuu. Verkkoympäristössä ei riitä, että tallennetaan pelkästään dokumenttien asiasisältö. Dokumenteista pyritään saamaan lisäksi talteen dokumenttien tuottamiseen ja tekijänoikeuksiin liittyvät tiedot. Sen lisäksi tallennetaan dokumenttien käyttämiseen verkkoympäristössä liittyvää tietoa. Dokumenttien tuottamiseen, käyttämiseen verkkoympäristössä sekä tekijänoikeuksiin liittyvä tieto onkin dokumentteja kuvailevaa metatietoa.

5.3 Dokumenttien metatieto

Dokumenttien metatieto sisältää dokumentteja kuvailevaa tietoa. Sen tarkoitus on tehostaa dokumenttien löytymistä. Dokumenttien indeksointiin perustuva metatieto pystyy kuvaamaan vain tekstidokumentteja. Indeksoinnin ohelle tarvitaankin dokumentteja kuvailevan tiedon tallennusta. (Desai, 1997.)

Kuvaileva metatieto sisältää erilaista tietoa dokumenteista. Se kertoo dokumentista itsestään mm. dokumentin nimen, sisällön ja tekijän sekä antaa monipuolista tausta- ja käytettävyystietoa dokumentista. Metatietoon voidaan määritellä tietoa dokumenttien tekijänoikeuksista aiheutuvista käyttörajoituksista. Siinä voidaan myös esimerkiksi digitaalisen allekirjoituksen avulla varmistaa, ettei dokumenttia eikä metatietoa voi muuttaa kuin vain sen muuttamiseen oikeutetut henkilöt. Myös dokumentteihin liittyviä kommentteja voidaan lisätä

metatietoon. Metatieto sisältää myös tietoa mm. dokumentin alkuperästä, päivityksestä sekä löydettävyydestä. (Mackenzie Owen & Walle, 1996; Desai, 1997.)

Metatiedolla voidaan kuvata myös muita kuin tekstidokumentteja. Se sopii kaikenlaisten dokumenttien, kuten kuva-, ääni-, video- ja multimediadokumenttien, kuvailuun. Metatiedolla kuvataan myös arkiston ominaisuuksista. Siinä esimerkiksi kerrotaan, kuinka paljon arkistossa on dokumentteja, ja kenen tuottamia dokumentteja arkistossa on. (Cathro, 1997; Desai, 1995; Desai 1997, 192, 203; Preserving Digital Information, 1996.)

Metatiedon määrittelemiseksi dokumentteihin on erilaisia tapoja. Mitään standardoitua tapaa tai standardoituja ominaisuuksia metatiedon määrittelemiseksi ei ole. Erilaiset yhteisöt määrittelevät metatieto-ominaisuuksia dokumenteilleen omien tarpeidensa mukaisesti. Ammattilaiset käyttävät erilaisia luettelointisysteemejä ja indeksointijärjestelmiä. Metatiedon määrittely onkin huomioitu TEI-hankkeessa. Hankkeessa kehitettiin tapa sijoittaa dokumentin metatieto dokumentin otsikkoon, headeriin. Samalla myös määritettiin dokumentin ominaisuudet, jotka otsikossa on kuvattava. TEI-dokumenttien sisältämä metatieto toimii sekä ammattilaisten apuna dokumenttien luokittelussa että hakumenetelmien käsittelemänä metatietona. (TEI, 1997; ks. myös Mackenzie Owen & Walle, 1996.)

Dokumenttien metatiedon määrittelyä varten on kehitetty erilaisia järjestelmiä. Niistä esimerkkinä on tässä kaksi erilaista järjestelmää, Dublin Core ja Semantic Header.

Dublin Core-metatieto perustuu 15:een dokumenttia kuvailevaan ominaisuuteen (Dublin Core -tallennusalusta, 1998). Dokumenttia kuvailevat ominaisuudet voi määrittää verkossa olevan Dublin Core -tallennusalustan avulla. Palvelu palauttaa ominaisuuksista kertovan HTML-koodin, jonka voi liittää HTML-dokumenttiin, `<head>` -tunnisteiden väliin. Dublin Coressa metatieto määräytyy etukäteen määriteltyjen ominaisuuksien avulla. Se on hakumenetelmien

käytössä HTML-dokumenttien otsikossa erillisenä osiona. (Dublin Core Metadata, 1997 ; Dublin Core -tallennusalusta, 1998)

Dublin Core-metatieto voidaan liittää HTML-dokumenttiin kahdella muullakin tavalla. Toinen tapa on muodostaa HTML-dokumentti tietokannassa olevasta tiedosta. Siihen liitetään makron avulla myös metatieto. Tämän on tulevaisuuden tekniikkaa, joten sen käytöstä ei vielä ole kokemuksia. Toisen tavan mukaan erillään oleva metatieto upotetaan SSI (Server Side Include) -käskyn avulla HTML-sivuun dynaamisesti. Myös tämän tekniikan käyttö on vasta kehitteillä. Kummassakin tekniikassa on kuitenkin olennaista se, että metatieto liitetään HTML-dokumentin yhteyteen. (Powell, 1998.)

Semantic Header perustuu myös 15:een valmiiksi määritettyyn dokumentin ominaisuuteen. Kyseiset ominaisuudet määritetään dokumentille verkossa olevan lomakkeen avulla. Syntynyttä metatietodokumenttia ei kuitenkaan tallenneta kyseiseen dokumenttiin vaan erilliseen tietokantaan (SHDDB, The Semantic Header Database System). Tietokannasta metatieto on yhteydessä varsinaiseen dokumenttiin. Haut kohdistetaan metatietokantaan sitä varten kehitetyllä Semantic Header -kyselylomakkeella. Metatietokannasta esitetään dokumenttien metatiedot, joista käyttäjä pääsee halutessaan varsinaiseen dokumenttiin. (Desai, 1997.)

Edellä esittämäni metatiedon määrittelytavat on tarkoitettu tavalliselle käyttäjälle, joka itse määrittelee dokumenttinsa metatiedot määritettyjen ominaisuuksien avulla. Metatiedon liittäminen dokumenttiin tapahtuu näissä esimerkeissä eri tavoin. Toisessa metatieto liitetään dokumenttiin, toisessa erilliseen tietokantaan. Varsinaista dokumenttia ei siirretä mihinkään. Se on siellä, mihin tuottaja on sen sijoittanut.

Yksi tapa määritellä metatieto on hyödyntää rakenteisten dokumenttien rakennetietoa. Dokumenttien rakennetta ilmentävät elementit. Ne ovat metatietoa sekä dokumenttien yksittäisistä osista että dokumenttien loogisesta rakenteesta. Elemetteihin liitettävät attribuutit kertovat yksittäisiin dokumenttien osiin liit-

tyvää metatietoa. Rakenteisten dokumenttien rakennemäärittely eli DTD on yksinään metatietoa dokumentin ominaisuuksista. (Böhm, Aberer, Neuhold & Yang, 1997.)

Böhmin, Abererin, Neuholdin ja Yangin (1997) tutkimuksessa tarkasteltiin DTD:n käyttöä dokumenttien etsimisessä metatiedon avulla. DTD:stä valittiin tietyt, parhaiten dokumenttityyppiä kuvaavat elementit. Niiden pohjalta rakennettiin tietokantatekniikoita hyödyntäen kyselylomakemuodostin. Se tuotti käyttäjälle graafisen kyselylomakkeen tietokannassa olevien dokumenttien etsimiseksi. Metatietoa ei siis määritelty erillisenä dokumenttina mihinkään, jota käyttäjä sellaisenaan hyödyntäisi. Kyselylomakkeella käyttäjälle esitettiin tietokannassa olevien dokumenttien metatieto-ominaisuuksia. Niiden avulla käyttäjä teki haun tietokantaan.

TEI-dokumenteissa ja Dublin Coressa metatieto tallennetaan dokumenttiin. Semantic Header ja Böhmin ym. tutkimuksessa metatietoa hyödynnettiin tietokantatekniikoiden avulla. Metatiedon tallentamisella eri paikkaan kuin varsinainen dokumentti on joitakin hyötyjä. Metatiedon käsittely on tehokkaampaa silloin kun se on koottuna yhteen paikkaan. Metatietoon kohdistetun haun mahdolliset kustannukset ja rajoitukset (aika, raha, tietoliikkenneyhteyksiin liittttyvät) ovat pienemmät kuin suoraan itse dokumentteihin kohdistetussa haussa. Sellaisten dokumenttien kuten ääni-, kuva- ja multimediadokumenttien tallentaminen on yksinkertaisempaa, kun metatieto tallennetaan erikseen. (Cathro, 1997; Desai, 1995; Desai 1997.)

Erillaiset tavat määritellä metatietoa dokumenteista johtavat siihen, ettei metatiedon käsittely ole yhdenmukaista. Eri metatietokuvauksissa olevilla saman nimisillä ominaisuuksilla ei välttämättä ole mitään tekemistä toistensa kanssa. Esimerkiksi tekijä tarkoittaa toisessa metatietokuvauksessa dokumentin kirjoittajaa ja toisessa määrätyn tehtävän hoitajaa. Tämä onkin nostanut esiin metatietokuvauksissa olevien ominaisuuksien yhdenmukaisuuden. Jos metatietokuvauksissa olevat ominaisuudet olisivat keskenään yhteensopivia ja standardoituja,

voisivat myös hakumenetelmät käsitellä kaikkea metatietoa yhteensopivasti. Yhteensopivuuden kehitystyö on aloitettu mm. W3C:n Namespaces-projektissa (Bray, Hollander & Layman, 1998). (Heery, 1997.)

Metatiedon, jolla kuvataan dokumenttien ominaisuuksia ja käyttömahdollisuuksia, merkitys dokumenttien löytymisen kannalta on moninainen. Metatietoa käytetään dokumenttien löytymisen tehostamiseksi. Sitä käytetään myös haku- ja kyselymenetelmien kehittämisessä. Erityisesti metatieto edesauttaa erilailla tietoa esittävien dokumenttien löytymistä. Metatiedon avulla voidaan kuvata dokumentteja, joita ei voida indeksoida. Dokumenttien löytymistä tehostavaa metatietoa määritetään eri tavoin dokumentteihin. Määrittelyssä hyödynnetään erilaisia metatietokuvauksia tai dokumenttien rakennetta. Metatieto ei kuitenkaan ole vielä standardoitua tietoa, joka mahdollistaisi eri tavoin määritellyn metatiedon yhteensopivan käsittelyn.

5.4 XML-standardiperhe ja arkistointi

Dokumenttien arkistoinnissa tavoitteena on tallentaa dokumenttien kokonaisuus. Dokumenttien kokonaisuus koostuu dokumenttien sisällöstä ja linkeistä sekä dokumenttia kuvailevasta metatiedosta. Dokumentit on arkistoitava niin, että ne säilyvät pitkään ja ovat löydettävissä niiden alkuperäisessä muodossa, kuten kuvana, äänenä tai tekstinä.

XML-standardiperhe perustuu dokumenttien rakenteiselle tallennustavalle, josta on monenlaista etua dokumenttien käytettävyyden kannalta. XML:ssä ei ole rajoitettu käytössä olevien elementtien määrää kiinteällä rakennemäärittelyllä. Dokumenttien rakenne onkin määriteltävissä dokumenttien tarpeita vastaavaksi. Dokumenttien sisältö voidaan tallentaa itsenäisinä osina, entiteetteinä. Entiteeteistä kootaan yhtenäinen dokumentti dokumentin juuressa olevien entiteettiviitteiden avulla. Standardiperheeseen kuluu myös dokumenttien linkityksiä määrittävä linkkistandardi ja dokumenttien esitystavan mahdollistava

tyylistandardi. XML-dokumenttien osat, dokumenttien sisältö, rakenne, esitystapa ja linkit voidaan kaikki tallentaan itsenäisesti.

Arkistoitavat XML-dokumentit ja rakennemäärittely

Arkistoitaessa dokumentteja XML-dokumenteiksi joudutaan valitsemaan tallennetaanko dokumentit rakennemäärittelyn avulla vai ilman. Jos dokumenttien tallentamiseksi valitaan rakennemäärittely, johtaa se lisävalintoihin. Tällöin joudutaan harkitsemaan, halutaanko arkiston olevan yhdenmukainen muiden arkistojen kanssa. Jos haetaan yhdenmukaisuutta, käytetään samaa rakennemäärittelyä kuin muutkin arkistot. Jos ei haeta yhdenmukaisuutta, käytetään täysin omaa rakennemäärittelyä. Samoin joudutaan ratkaisemaan, käytetäänkö kaikilla arkiston dokumenteilla samaa rakennemäärittelyä vai onko samassa arkistossa eri rakennemäärittelyn mukaisia dokumentteja. Yhden rakennemäärittelyn avulla arkiston dokumentit saadaan yhdenmukaisiksi. Usean rakennemäärittelyn salliminen vaikuttaa tiedonvälitykseen. Rakennemäärittelyn valintaan vaikuttaa myös se, mitä rakennemäärittelyjä käyttävät arkistoon dokumentteja antavat tahot, kuten esimerkiksi julkiset tahot, julkaisijat ja yksittäiset yritykset.

Rakennemäärittely vaikuttaa dokumenttien rakenteen suunnittelun nopeuteen. Jos arkistoitaessa joudutaan muuntamaan dokumentit XML-muotoon, yhdenmukainen rakennemäärittely nopeuttaa ja helpottaa muuntamista. Rakennemäärittelyllä on merkitystä myös tiedonvälityksessä. Vastaanottajan ohjelman pitäisi pystyä tulkitsemaan rakennemäärittelyn mukaisesti rakennettua dokumentteja. Koska etukäteen ei aina tiedetä, pystyykö vastaanottajan ohjelma tähän, välitetään rakennemäärittely dokumenttien ohella. Rakennemäärittelyä voidaan käyttää myös dokumenttien rakenteen muuttumattomuuden toteamiseen. Kun dokumentit on tallennettu rakennemäärittelyn avulla voidaan aina tarkistaa, ovatko arkistossa olevat dokumentit rakennemäärittelyn mukaisia.

Dokumentit voidaan tallentaa arkistoon ilman rakennemäärittelyä, jolloin ne ovat hyvin muodostettuja XML-dokumentteja. Jos dokumentit muunnetaan arkistoitaessa XML-muotoon, vaatii hyvin muodostetuiksi dokumenteiksi muuntaminen enemmän aikaa kuin rakennemäärittelyn avulla tehtävä muuntaminen. Hyvin muodostettujen XML-dokumenttien rakenne joudutaan miettimään ja suunnittelemaan jokaisen dokumenttien kohdalla erikseen. Hyvin muodostettujen dokumenttien välitys on yksinkertaisempaa, koska rakennemäärittelyä ei tarvitse välittää dokumenttien mukana. Tällöin ei ole mahdollista tarkistaa rakennemäärittelyn avulla, onko dokumenttien rakenne on säilynyt muuttumattomana pitkäaikaisessa säilytyksessä.

Rakennemäärittely voidaan tallentaa erikseen varsinaisesta dokumentista. Tällöin sen ylläpito on mahdollista. Sille voidaan määrittää tekijänoikeudet ja niistä aiheutuvat käyttörajoitukset. Myös muutosten tekoon oikeuttavat varmistustekniikat ovat liitettävissä itsenäiseen rakennemäärittelyyn.

Entiteetit XML-dokumenttien arkistoinnissa

Dokumenttien sisällön arkistoinnissa joudutaan myös valitsemaan, arkistoidaanko dokumentit kokonaisena vai itsenäisinä osina, entiteetteinä. Valinnassa olisi huomioitava, että verkkoympäristössä dokumenttien arkistointi edellyttää jatkuvaa dokumenttien ylläpitoa ja vaihtamista tallennusmedialta toisella. Tämän lisäksi dokumenttien käyttömahdollisuuksia rajoitetaan tekijänoikeuksilla. Dokumenttien muuttumattomuutta ja muutosoikeutta osoittamaan dokumentteihin liitetään erilaisia varmistuskenoja.

Kun dokumentit on jaettu entiteetteihin on dokumenttien ylläpito yksinkertaisempaa. Ylläpito voidaan kohdistaa muutoksen kohteena olevaan entiteettiin, eikä koko dokumenttia tarvitse käsitellä. Dokumenttien sisällön eri osien tekijänoikeuksien ja niiden mukaisten käyttörajoitusten määrittäminen ja ylläpito onnistuvat, kun eri osat on tallennettu itsenäisinä. Myös eri osien muuttumat-

tomuutta ja muuttamisoikeutta ilmaisevat varmistuskeinot voidaan määritellä yksiselitteisemmin, kun dokumentti on jaettu selkeästi osiin.

Arkistoituja dokumentteja joudutaan siirtämään tallennusmedialta toiselle tekniikan muuttuessa. Entiteetteihin jaetun dokumentin siirtäminen on turvallista, koska juuressa mainitaan kaikki dokumenttiin kuuluvat entiteetit. Näin ollen tieto kaikista entiteeteistä on tallella, jos juuri on tallella.

XLink-linkit XML-dokumenteissa

Dokumenttien sisältöön voidaan liittää tietoa XLink-linkkien avulla. Linkkien tallentaminen on ollut ongelmallista, koska dokumenttien sijainnin muututtua linkit eivät enää toimi. XLink-linkkistandardin mukaisesti linkit voidaan ryhmitellä ja tallentaa dokumenteista erilleen. XLinkit voivat myös olla kaksisuuntaisia. Näin ollen dokumenttien sijainnin muuttuessa voidaan etsiä kaikki dokumentteihin viitanneet linkit, ja muuttaa ne uuden osoitteen mukaiseksi. Erillään olevia linkityksiä on mahdollista ylläpitää ilman, että dokumenttien sisältöön tarvitsee puuttua. XLink-linkkistandardin mukaisiin linkkeihin määritetään tietoa siitä, missä suhteessa linkeillä yhdistetyt dokumentit ovat toisiinsa. Samanlaista tietoa sisältävät linkit voidaan ryhmitellä yhteen. Tämä puolestaan on käyttäjän kannalta tiedon etsimistä helpottava tekijä.

Mahdolliset muutokset sisällössä tai rakenteessa tai jossain muussa dokumentteihin liittyvässä asiassa voidaan liittää dokumentteihin XLink-linkkien avulla, jos ne katsotaan tarpeellisiksi. Tällöin ei tarvitse välttämättä muuttaa alkuperäistä dokumentteja eikä niihin liittyviä varmistustekniikoita.

XLink-linkkistandardi mukaiset linkit sopivat pitkäaikaiseen säilytykseen, koska niiden ylläpito on mahdollista. Koska linkit voidaan tallentaa erilleen varsinaisesta dokumentista, voidaan niihin myös määritellä tekijänoikeustietoa, käyttörajoituksia sekä muuttamisoikeuteen liittyviä varmistustekniikoita.

Tyylistandardi

XML-dokumenttien esitystapa on mahdollista arkistoida erilleen dokumenttien sisällöstä XSL-tyylistandardilla. Tällöin niihin on määriteltävissä tekijänoikeustieto, käyttörajoitukset sekä muuttamisoikeuden varmistamiskeinot. Samoin niiden ylläpito on mahdollista ilman, että varsinaisia dokumentteja tarvitsee käsitellä. Tyylistandardin avulla voidaan määritellä alkuperäisistä dokumenteista uusia dokumentteja, joiden esitystapa poikkeaa alkuperäisestä sisällön säilyessä ennallaan. Arkistoinnin kohteena voi olla haluttaessa useita esitystapaversioita. XML-standardin käyttökelpoisuutta arvioitaessa tuo XSL-tyylistandardi siihen yhden tärkeän ominaisuuden: tyylistandardi on kehitetty erityisesti verkossa olevien XML-dokumenttien näytöllä esittämistä varten. Tällöin XML-dokumentit saavat näytöllä esitystavan ilman ohjelmistoriippuvia tyylikieliratkaisuja. Pitkäaikaisen säilytyksen kannalta ongelmia voi tietenkin syntyä silloin, kun näyttötekniikat muuttuvat. Vanhat tyylimääritykset eivät ehkä olekaan silloin enää toimivia. Tässäkin tilanteessa dokumenttien sisältö pitäisi kuitenkin olla esitettävissä uusien tyylimääritysten avulla.

Metatieto XML-dokumenttien arkistoinnissa

Dokumenttien arkistointi edellyttää varsinaisen sisällön ohella monenlaisen muun tiedon, metatiedon, tallentamista dokumenteista. Metatietoa voidaan tallentaa myös dokumenteista erilleen tallennetuista rakennemäärittelystä, linkeistä ja tyylimäärityksistä. Metatietoa määritetään ja liitetään XML-dokumentteihin kolmella eri tavalla:

1) hyödynnetään XML-dokumenttien rakennetta,
2) rakennetaan metatiedosta oma XML-dokumentti tai
3) ilmaistaan metatieto metatietosovelluksilla, kuten RDF:llä.

Seuraavaksi tarkastelenkin lähemmin näitä mahdollisuuksia.

1) XML-dokumenttien metatieto voidaan arkistoida suoraan XML-dokumenttien rakenteesta. Tällöin määritetään erilliset metatietoelementit, joissa on tietoa dokumenteista. Sen lisäksi metatiedoksi määritetään jotkin dokumenttien ominaisuuksia selvästi kuvaavat elementit, elementtien attribuutit tai rakennemäärittely kokonaisuudessaan. Esimerkiksi tietokantatekniikoita hyväksikäyttäen saadaan näin määritetty metatieto hyödynnetyksi dokumenttien etsimisessä. Tällaisen metatiedon ylläpito ja siirtäminen toiselle tallennusmedialle tapahtuu varsinaisten XML-dokumenttien rakennemäärittelyn ylläpidon ja siirtämisen yhteydessä. Metatietoon liittyvät tekijänoikeudet ja niistä aiheutuvat käyttörajoitukset voidaan määritellä rakennemäärittelyn tekijänoikeuksien ohella, samoin kuin oikeudet muutosten tekemisiin.

2) Rakennettaessa metatiedosta oma XML-dokumentti määritetään aluksi metatietodokumenttien rakenne. Rakenne voi olla jonkin metatietokuvauksen, esimerkiksi Dublin Core tai Semantic Header, mukainen. Jos metatietoon halutaan liittää lisätietoa, esimerkiksi käytetyn metatietokuvauksen ulkopuolista metatietoa, voidaan se liittää metatietodokumenttiin linkkien avulla.

Metatietodokumentti voidaan rakentaa myös entiteeteistä. Tällöin ominaisuudet, joista metatietodokumentti muodostuu, ovat kukin oma entiteettinsä. Entiteettien käyttö edellyttää, että entiteetit ovat jonkin DTD:n mukaisesti määritettyjä ja XML-standardin mukaisia.

Metatietodokumentti voidaan liittää varsinaiseen dokumenttiin joko linkeillä, tai se voi olla yksi varsinaisen dokumentin entiteeteistä. Myös erilaiset tietokantatekniikat mahdollistavat dokumentin ja metatiedon tallentamisen erillisinä. Linkkien ylläpidon helpottuminen XLink-linkkistandardin avulla mahdollistaa linkkien käytön dokumenttien ja metatiedon yhdistämisessä. Kun linkit ovat kaksisuuntaisia, voidaan dokumenttien sijainnin muuttuessa muuttaa myös metatiedon tietoja. Jos metatieto tallennetaan dokumentin entiteettinä, on sen oltava jonkun DTD:n mukainen, jotta se voidaan yhdistää dokumenttiin.

Metatietodokumentin ylläpito on mahdollista, kun se on itsenäinen dokumentti. Silloin ovat myös sekä tekijänoikeudet että käyttörajoitukset ja muutoksenteko-oikeudet määritettävissä. Jos metatieto jakautuu vielä pienempiin osiin kuten linkkeihin ja entiteetteihin, myös niihin voidaan liittää käyttöön ja muutoksentekoon liittyvää tietoa.

3) Kolmas tapa määrittää metatietoa XML-dokumenttiin on käyttää sitä varten kehitteillä olevia sovelluksia, kuten RDF:ää. RDF:n sisällä voidaan käyttää erilaisia metatietokuvauksia. Sovellusten kehitystyö on vielä kesken, joten niiden ominaisuudet voivat vielä täsmentyä ja muuttua.

Laitteisto- ja ohjelmistoriippumattomuus XML-dokumenttien arkistoinnissa

Dokumenttien pitkäaikainen säilyminen edellyttää myös sitä, etteivät dokumentit katoa, kun laitteistot ja ohjelmistot kehittyvät ja muuttuvat (ks. mm. Marcouxn ja Sévignyn 1997; Preserving Digital Information, 1996). XML-dokumenttien tallennusmuotona soveltuu pitkäaikaiseen säilymiseen. Vaikka käytössä ei tulevaisuudessa olisikaan ohjelmaa, joka osaisi tulostaa XML-dokumentteja, ne voidaan aina tulostaa merkkijonomuodossa. Merkkijonomuodossakin XML-merkattujen dokumenttien sisältö on ymmärrettävissä ja tulkittavissa. Toisaalta, tulostava ohjelma voidaan toteuttaa uudessa ympäristössä, koska dokumenttien sisältö, rakenne, esitystapa ja linkitykset on määritelty sovellusriippumattomalla, standardoidulla tavalla.

XML-standardilla on pitkäaikaisen säilymisen kannalta myös se etu, että sen merkistö on kiinnitetty UNICODE-standardiin. Näin vältytään merkistön erilliseltä valinnalta, ja eri kielillä tuotetut dokumentit ovat myös kaikkialla ymmärrettävissä.

XML-dokumenttien siirtämiseksi järjestelmästä toiseen ei tarvita muunnosohjelmia, jos kumpikin järjestelmä ymmärtää XML-standardiperheeseen kuuluvia standardeja. Järjestelmien tuki XML-standardille on mahdollista, koska standardi on julkinen, ja se on eri ohjelmistotuottajien käytettävissä. Tuki on toden-

näköistä myös sen vuoksi, että XML-standardin kehitystyöllä on ollut laitteisto- ja ohjelmistotuottajien kuten Microsoftin, Netscapen, Sunin ja Hewlett-Packardin tuki (Homer, 1997; XML Overview, 1997.)

Lyhyesti yhteenkoottuna

XML-standardiperhe tarjoaa arkistoinnille mahdollisuuden jakaa dokumentit eri tasoilla osiin. Dokumenteista voidaan ensiksi erottaa sisältö, rakenne, tyylit ja linkit. Sen lisäksi dokumenttien sisältämä tieto voidaan jakaa elementteihin, entiteetteihin ja linkkeihin. Tällainen dokumenttien ja niiden sisältämän tiedon paloittelu mahdollistaa arkiston dokumenttien ylläpidon. Sen lisäksi se helpottaa tekijänoikeuksien ja käyttömahdollisuuksien sekä erilaisten varmistustekniikoiden määrittelyä niihin hyvinkin yksityiskohtaisesti. Tiedon paloittelu mahdollistaa myös sen, että dokumentteista voidaan määrittää yksityiskohtaista tietoa ja dokumentteihin voidaan liittää tietoa erilaisin menetelmin. XML-standardiperhe mahdollistaa merkkijonomuotoisella tallennustavalla dokumenttien pitkäaikaisen säilymisen.

6 XML JA XSL KÄYTÄNNÖSSÄ

Tässä luvussa tarkastelen MS Word (versio 6.0) -tekstinkäsittelyohjelmalla editoimani dokumentin muuntamista XML-dokumentiksi ja sen muotoilemista HTML-dokumentiksi. Muunnan XML-dokumentin näytöllä esitettäväksi HTML-dokumentiksi XSL-tyylimäärityksen avulla. Kokeilujen tarkoituksena on hankkia tietoa siitä, millaisia mahdollisuuksia ja rajoituksia XML-standardiperhe tällä hetkellä tarjoaa dokumenteille arkistoinnin näkökulmasta. Standardiperheen XLink-osaa en voinut kokeilla, sillä linkkejä tukevia ohjelmia ei ole vielä käytössä. Käytännön kokeiluissa keskityn dokumentin sisällön eri osien, entiteettien, esitystavan ja metatiedon yhdistämiseen, sekä dokumentin toimivuuden tarkasteluun ilman rakennemäärittelyä.

6.1 XML-työkaluista

Etsin XML-työkaluja Internetistä maaliskuussa 1998. Tällöin kaupallisia XML-työkaluja oli tarjolla mm. ArborTextin ADEPT 7 sekä Enigman INSIGHT Fact Sheet. ArborTextin ADEPT 7 on tarkoitettu XML- ja SGML-dokumenttien tuottamiseen. Enigman INSIGHT Fact Sheet on elektroninen julkaisuympäristö, joka tukee SGML- ja XML-dokumenttien hallintaa. Monen valmistajan erilaiset kaupalliset dokumenttien hallintaan tarkoitetut ohjelmat tukevat XML-dokumenttien käsittelyä, vaikka heillä ei ainakaan vielä ole varsinaisia XML-työkaluja. Esimerkkinä seuraavat:

- Banff: OmniMark (tuki XML-dokumenttien hallintaan),
- Inso Corporation: DynaWeb (muuntaa DynaText elektroniset kirjat XML-dokumenteiksi),
- Frontier 5-ympäristö (ympäristö dokumenttien luontiin, julkaisuun ja ylläpitoon),
- Poet Software: Poet Content Management Suite (SGML- ja XML-dokumenttien hallintaan kehitetty ympäristö).

XML-jäsentäjiä oli usealla valmistajalla ilmaisjakeluohjelmana, mm.

- Balise: Release 4.0 (jäsentää XML-dokumentit sekä ilman DTD:tä että DTD:n kanssa),

- Microsoft:
1) XML Parser in C++ (jäsennys ilman DTD:tä),
2) XML Parser in Java (jäsennys DTD:n kanssa),
- DataChannel: DXP-DataChannel XML parser (DTD:n vaativa jäsentäjä),
- Textuality: Lark XML-prosessori (ilman DTD:tä toimiva jäsentäjä).

Internetissä oli maaliskuussa 1998 vähän julkisesti saatavia XML-ohjelmistoja. Tällöin löytyivät seuraavat tuotteet:

- Vervet Logic LLC: XML Pro, versio Beta 1.0b. XML Pro on XML-editori, jolla voi käsitellä vain editorin mukana tulleita esimerkkidokumentteja. Sitä ei tuolloin voinut käyttää omien dokumenttien käsittelyyn

- ArborText: XML Styler 1.0; versio 2.0 ilmestyi myöhemmin. Kyseessä on XSL-editori, joka on tarkoitettu XSL-tyylitiedostojen luomista ja muotoilemista varten.

- Microsoft: msxsl-prosessori. XML-prosessori , joka pystyy yhdistämään XML-dokumentteihin XSL-tyylimäärittelyt ja esittämään dokumentit HTML-dokumentteina näytöllä.

Käytännön kokeiluissani käytin ArborTextin XSL-editoria ja Microsoftin msxsl-prosessoria.

6.2 MS Word-dokumentista näytöllä esitettävään XML-dokumenttiin

Tein käytännön kokeilut maaliskuussa ja toukokuussa 1998. Kokeilin XML:ää kahdella eri ohjelmalla. Kokeiluissa oli tavoitteenani saada tietoa siitä, kuinka XML-dokumentti voidaan tuottaa, kuinka XML-dokumenttiin saadaan yhdistetyksi XSL-tyylimäärittelyt, ja voidaanko XML-dokumentti esittää näytöllä. Kokeilut oli tarkoitus tehdä hyvin muodostetulla XML-dokumentilla ilman DTD:tä.

6.2.1 Dokumentin muuntaminen ja käytetyt ohjelmat

Tässä kohdassa käsittelen vaihe vaiheelta, kuinka muunnin MS Word-dokumentin XML-dokumentiksi, ja mitä ohjelmaa siinä käytin. XML-kokeiluihin valitsin MS Word (versio 6.0) -tekstinkäsittelyohjelmalla editoimani EVA-projektin raportin "XML:n käyttö verkkosovelluksissa". Sen jälkeen ku-

vaan XSL-tyylimääritysten tekoa. XSL-tyylimääritysten lopussa esittelen lyhyesti kokeilujen aikataulun eri ohjelmien käytön osalta.

Dokumentin hierarkisen rakenteen suunnittelu

MS Word-dokumentin muuntaminen XML-dokumentiksi vaati hierarkisen rakenteen suunnittelua. Dokumentin hierarkisen rakenteen suunnittelussa sovelsin textbk.dtd:tä (Veijola, 1998). Tämä DTD perustuu yleisempään docbook.dtd:hen (The Davenport Group, 1997). Kokeilussa käytetty DTD, textbk.dtd, ei ihan sellaisenaan vastannut dokumentin rakennevaatimuksia, varsinkaan lähdeluettelon osalta. Rakenteeseen tein seuraavat lähdeluetteloa koskevat muutokset:

- lisäsin siihen yhden tunnisteen ilmaisemaan päivämäärää, jolloin verkkolähteeseen on viitattu
- muutin tekijöiden ja julkaisun nimen paikkaa.

Koska kokeiluissa oli tarkoituksenani käyttää hyvin muodostettua XML-dokumenttia, käytin DTD:tä apuna vain dokumentin hierarkisen rakenteen muodostamisessa ja elementtien nimeämisessä. DTD:stä oli apua myös dokumentin rakenteen merkkaamisessa.

Dokumentin rakenteen merkkaaminen

Muunnin MS Word-dokumentin XML-muotoon seuraavasti:

1. Muunnin Word-dokumentin tekstitiedostoksi.
2. Merkkasin dokumentin textbk.dtd:n mukaisilla elementeillä NT-EMACS +PSGML -ohjelmalla.

Toimivia, julkisesti saatavilla olleita XML-editoreita ei löytynyt maaliskuussa 1998, joten päädyin käytettävissä olleeseen NT-EMACS +PSGML:ään. NT-EMACS +PSGML -ohjelma on SGML-dokumenttien muokkausohjelmisto. Sillä voidaan muokata mitä tahansa DTD:tä käyttävää dokumenttia. NT-EMACS +PSGMLin käytön etuna on sen rakenteen merkkaamista avustava editori. Do-

kumenttia merkatessa editori käyttää hyväksi dokumentin DTD:ssä määriteltyä hierarkista rakennetta. (Staflin, 1996.)

Kun olin muuntanut dokumentin merkatuksi dokumentiksi, ajoin sen EMACSin automaattisella sgmls-jäsentimellä, eikä virheilmoituksia tullut. Dokumentti oli tämän jälkeen SGML-dokumentti. Muunnin sen XML-dokumentiksi lisäämällä dokumentin alkuun esittelyosan `<?XML version="1.0"?>` ja poistamalla DOCTYPE-määrittelyn. Nämä toimenpiteet riittivät, koska DTD:stä ei käytetty mitään sellaista, mikä olisi ollut XML-standardin vastaista.

Dokumentti entiteeteiksi

Jotta voisin kokeilla XML-dokumentin muodostumista erillisistä entiteeteistä, jaoin dokumentin pienempiin osiin. Jaoin dokumentin tarkoituksellisesti eri kokosiin osiin. Kansilehti ja luku 1 muodostivat yhden entiteetin, luku 2 toisen , luvut 3, 4, 5 ja 6 kolmannen ja lähdeluettelo neljännen entiteetin. Dokumentissa ei ollut yhtään kuvaa, joten kaikki entiteetit olivat sisällöltään tekstiä.

Tallensin kaikki entiteetit XML-muotoon antamalla kullekin oman nimen ja lisäämällä nimen jälkeen XML-muotoa osoittavan .xml-tunnisteen, esimerkiksi `3456luku.xml` ja `lahteet.xml`. Tallensin entiteetit tekstitiedostoina. Entiteettien alkuun ei tule esittelyosaa `<?XML version="1.0"?>`. Entiteetit olivat siis itsenäisiä hyvin muodostettuja XML-dokumentteja, joissa ei ollut yhtään esittelyosaa. Kaikissa entiteeteissä korvasin ä- ja ö-kirjaimet `&auml;` :lla ja `&ouml;`:lla, koska entiteeteistä koottu dokumentti oli tarkoitus esittää HTML-dokumentteina näytöllä.

Entiteettien XSL-tyylimääritykset

Tein kaikille muodostamilleni entiteeteille erikseen tyylimääritykset ArborTextin XML Style 1.0 -editorilla (ArborText, 1998). Editorissa on online käyttöopas ja graafinen käyttöliittymä. Käyttöopas kuvaa editorin mahdollisuuksia, mutta

ei anna selkeitä käyttöohjeita. Editorilla voi tehdä yksinkertaisia tyylimäärityksiä kappaleille ja otsikoille. Esimerkiksi editorilla tehty yksinkertainen määritys otsikon koolle ja tyylille olisi:

```
<xsl>
  <rule>
      <target-element type="head"/>
      <P font-size="14pt" font-weight="bold">
          <children/>
     </P>
  </rule>
</xsl>
```

Mihinkään monimutkaisempaan, kuten esimerkiksi listojen tekoon, editori ei kyennyt. Niinpä käytin editorilla aikaansaamiani tyylimäärityksiä pohjana, muutin niitä hieman ja määrittelin niihin tekstieditorilla (MS Word, versio 6.0) täydennyksiä mm. listojen osalta. Tyylimääritysten määrittelyssä käytin apuna W3C:n A Proposal for XSL -tyylimäärittelyä (1997) ja Microsoftin XSL Tutorial -opasta (1998).

Seuraavan esimerkin avulla käyn läpi tarkemmin XSL-tyylimääritysten tekoa. Esimerkissä on tyylimäärittely ensimmäisen tason alaotsikon esitystavalle:

```
<xsl>
<rule>
  <element type="section">
    <element type="section">
      <target-element type="head/>
    </element>
  </element>
  <P font-size="14pt" font-weight="bold">
    <children/>
  </P>
</rule>
</xsl>
```

Tässä määrittelin ensimmäisen tason alaotsikolle (`head`) vain kirjasinkoon ja -tyylin. Niiden lisäksi voisi määritellä mm. ylä-, ala- ja sivumarginaalit, kirjasinlajin, sisennykset, keskitykset ja automaattisen kappalenumeroinnin. XSL-syntaksi vaatii, että tyylimäärityksessä on oltava määritys `<children/>` juuri tässä muodossa. Sillä lähinnä tarkoitetaan, että käsiteltävän elementin lapsielementit käsitellään myös kyseisten tyylimääritysten mukaan. Tämä määritys ei

kuitenkaan estä määrittämästä käsiteltävän elementin lapsielementeille omia esitystapoja syntaksin mukaisesti. Alku- ja lopputunniste `<rule>` tarkoittaa, että niiden sisällä on jotakin elementtiä koskeva XSL-tyylimääritys.

Esimerkissä yksilöin käsiteltävän elementin sitä edeltävien elementtien eli vanhempien avulla. Tässä tapauksessa käsiteltävä elementti on siis ensimmäisen tason alaotsikko (`head`). Jotta käsittelyyn tulisi oikean tason otsikko, määrittelin otsikon tason vanhempien eli kahden hierarkisessa suhteessa olevan `section`-elementin avulla. Tunnistin `P` ilmoittaa, miten ensimmäisen tason alaotsikko esitetään näytöllä. Tunnistimen attribuutit osoittavat, millaisia muotoilupiirteitä alaotsikkoon liittyy.

Tunnistimien valinta riippuu siitä, mitä tunnistimia XML-dokumentin ja XSL-tyylimääritykset yhdistävä ohjelma tukee. Tätä esimerkkiä on käytetty Microsoftin msxsl-ohjelmalla, joka tukee lähes kaikkia HTML:n tunnistimia. Toisessa kokeilussa käytetty xslj ja Jade eivät tukeneet esimerkiksi tunnistinta `P`, joten jouduin muuttamaan tyylimäärityksiä tunnistimienkin osalta toisessa kokeessa.

Kirjoitin jokaiselle entiteetille oman tyylimääritykset ja tallensin kunkin tyylimääritykset omana tiedostonaan muotoon `nimi.xsl`. Näin pystyin kokeilemaan maaliskuussa erilaisten tyylimäärityksen tekoa ja liittämistä entiteetteihin (=hyvin muodostettuun XML-dokumenttiin), kunkin entiteetin esittämistä näytöllä sekä dokumentin kokoamista entiteeteistä. Toukokuussa päähuomio oli entiteeteistä muodostuvan dokumentin kokoamisessa ja metatiedon liittämisessä dokumenttiin. Maaliskuussa käytin XML-kokeissa Microsoftin msxsl-ohjelmaa (The Microsoft XSL Processor, 1998) ja ArborTextin XML Styleria, versio 1.0 (ArborText's XML Styler, 1998). Molemmat ohjelmat olivat julkisesti saatavilla. Toukokuussa käytin Jyväskylän yliopiston palvelimelle asennettuja ohjelmia xslj (Thompson, 1998) ja Jaden versiota 1.1.1. (Clark, 1998).

6.2.2 Työskentely msxsl-ohjelmalla

Microsoftin msxsl-prosessori yhdistää XML-dokumentin ja XSL-tyylimääritykset sekä muodostaa niistä näytöllä esitettävän HTML-dokumentin (The Microsoft XSL Processor, 1998). Ohjelmalla ei ole graafista käyttöliittymää, joten sitä käytetään MS-DOS-kehotteilla. Ohjelmassa msxsl on selkeät ohjeet, kuinka yhdistäminen tapahtuu:

```
-i <input file name>
-s <style file name>
-o <output file name>
```

Esimerkki komennosta:

```
msxsl -i 3456luku.xml -s 3456luku.xsl -o 3456luku.html
```

Käytössäni oli siis XML-muodossa olevia hyvin muodostettuja dokumentteja (entiteetit) ja ArborTextin XML Styler-editorin pohjalta itse tehdyt tyylimääritykset. Kaikista yksittäisistä entiteeteistä sain erilaisia tyylimäärityksiä kokeilemalla esitystavaltaan moitteettomia HTML-dokumentteja. XML-dokumenttien ja tyylimäärittelyjen yhdistäminen sekä XML-dokumenttien muuntaminen XSL-tyylimäärittelyjen avulla näytöllä esitettäväksi HTML-dokumentiksi onnistui Microsoftin msxsl-ohjelmalla.

Dokumentin kokoamisen entiteeteistä aloitin määrittelemällä juuren. Siinä määrittelin kaikki dokumenttiin sisällytettävät entiteetit ja niiden sijainnin. Sen lisäksi yhdistin entiteettien tyylimääritykset. Yhdellä tyylimäärityksellä määrittelin kaikille mahdollisille, yhtenäisessä dokumentissa oleville, elementeille esitystavan.

Kun sitten yritin muodostaa msxsl-ohjelmalla entiteeteistä yhden kokonaisuuden, se ei onnistunutkaan. Tarkoituksenani oli, ettei dokumentilla olisi määriteltyä DTD:tä eli dokumentti olisi vain hyvin muodostettu XML-dokumentti. Ohjelma antoi kuitenkin virheilmoituksen DTD:n puuttumisesta. Lisäsin juureen yksinkertaisen DTD:n, `gradu`, joka sai sisältää mitä tahansa:

```
<?XML Version="1.0"?>
```

```
<!DOCTYPE gradu[
<!ELEMENT gradu ANY>
<!ENTITY alku SYSTEM "1luku.xml">
<!ENTITY luku2 SYSTEM "2luku.xml">
<!ENTITY luvut3456 SYSTEM "3456luku.xml">
<!ENTITY lahteet SYSTEM "lahteet.xml">]>
<gradu>
&alku;
&luku2;
&luvut3456;
&lahteet;
</gradu>
```

Ohjelma ei tämän jälkeen antanut virheilmoituksia. Raportti ei kuitenkaan tulostunut kokonaisena HTML-dokumenttina näytölle. Näytölle ei tulostunut yhtään mitään, koska HTML-tiedosto oli tyhjä. Koska en löytänyt valmistajan WWW-sivuilta enkä erilaisista keskusteluryhmistä mitään ohjetta tai apua ongelman ratkaisemiseksi, en voinut jatkaa koetta.

Dokumentin sisällön jakaminen entiteetteihin oli mahdollista, mutta entiteeteistä en pystynyt Microsoftin msxsl-ohjelmalla yhdistämään yhtenäistä kokonaisuutta. Näin ollen en edes kokeillut metatiedon, metatietoentiteetin, lisäämistä dokumentteihin. Dublin Coren mukaista metatietoa voi liittää HTML-dokumenttiin. Kustakin kokeilussa mukana olleesta entiteetistä muodostettiin oma HTML-dokumentti. Jos olisin sijoittanut ne verkkoon, olisin voinut liittää niihin Dublin Coren-tallennusalustalla (1998) määriteltyä metatietoa.

Kokeilun materiaalina olleen raportin lukujen jakaminen itsenäisiksi entiteeteiksi, jotka toimivat hyvin muodostettuina XML-dokumentteina ilman rakennemäärittelyä, onnistui hyvin. Onnistuin yhdistämään hyvin muodostettuihin XML-dokumentteihin niiden tyylimäärittelyt ja muuntamaan ne HTML-dokumenteiksi Microsoftin msxsl-ohjelmalla. Näissä XML-dokumenteissa ei ollut muita entiteettejä kuin ä ja ö. Kun yritin yhdistää entiteetit yhdeksi dokumentiksi, ohjelma vaati toimiakseen rakennemäärittelyn. Lisäsin rakennemäärittelyn juureen. En saanut kuitenkaan Microsoftin msxsl-ohjelmalla yhtenäistä dokumenttia, joka olisi muodostettu entiteeteistä ja esitetty näytöllä HTML-dokumenttina.

6.2.3 Työskentely xslj- ja Jade-ohjelmilla

Thompsonin (1998) kehittämä xslj-ohjelma kääntää validit XSL-tyylitiedostot valideiksi DSSSL-tyylitiedostoiksi, joita voidaan käyttää XML-dokumenttien muuntamisessa Jaden (Clark, 1998) avulla. Jade on DSSSL-tyylikielen sovellus. Tässä kokeilussa ohjelmia käytetään HTML-dokumentin tuottamiseen hyvin muodostetuista XML-dokumenteista.

Ohjelmilla xslj ja Jade ei ole graafista käyttöliittymää. Niitä käytetään MS-DOS-kehotteilla. XML-dokumentin muuntaminen HTML-dokumentiksi tapahtuu kahden eri komennon avulla, joista tässä esimerkki:

```
xslj -h -c catalog kokodoc.xsl >kokodoc.dsl
```

Kun tämän komennon jälkeen ei tule enää tiedostoa kokodoc.xsl koskevia virheilmoituksia, voidaan antaa seuraava komento:

```
jade -2 -c catalog -d kokodoc.dsl -t sgml kokodoc.xml
>kokodoc.html
```

Kun tiedostoa kokodoc.xml koskevat virheilmoitukset loppuvat, näytölle tulostuu haluttu HTML-dokumentti.

Käytössäni oli aikaisempaa kokeilua varten määritelemäni juuri, `kokodoc.xml`, ja tyylimääritys, `kokodoc.xsl`. Juuressa määrittelin yhdistettävät entiteetit, ja tyylimäärityksessä määrittelin esitystavan dokumentin elementeille. Varsinaisen kokeilun aloitin ohjelmien käytön järjestyksen mukaisesti muuntamalla XSL-tyylimääritykset DSSSL-määrityksiksi. Virheilmoituksessa ohjelma valitti DTD:n puuttumisesta tyylimääritysdokumentista.

Lisäsin kokodoc.xsl-dokumentiin rakennemäärityksen `xsl.dtd` sekä esittelyosan `<?XML version='1.0' RMD='none'?>`. Tyylimäärityksestä tuli näin ollen XML-dokumentti, mutta se tallennettiin edelleen XSL-muodossa, `kokodoc.xsl`. Tämän jälkeen virheilmoitus ilmoitti, etteivät tyylimäärityksessä käytetyt elementit ja attribuutit ole xsl.dtd:n mukaisia. Rakennemääritys

xsl.dtd sisältää htmlfos.dtd:n, joten elementit oli muutettava htmlfos.dtd:n mukaisiksi. Dokumenttihan oli tarkoitus esittää HTML-dokumenttina. Kun kaikki elementit ja attribuutit olivat htmlfos.dtd:n mukaisia virheilmoituksia ei enää tullut ja pääsin seuraavaan komentoon.

Käytin komennossa samaa juurta, minkä olin määritellyt msxsl-ohjelmalle. Virheilmoituksessa valitettiin, ettei entiteeteissä olevia elementtejä eikä attribuutteja ole määritelty. Määrittelin uuden juuren (liite 1), jossa määrittelin dtd:n nimeltään `gradu` ja kaikki sen elementit sekä attribuutit eli kaikki yhtenäiseen dokumenttiin tulevat elementit ja attribuutit. Tämän jälkeen ohjelma ei hyväksynyt linkki-elementin attribuuttia. Jouduin muuttamaan sen DSSSL:n syntaksin mukaiseksi muodostuneeseen .dsl-dokumenttiin. Tämän jälkeen ei tullut enää virheilmoituksia, ja sain näytölle entiteeteistä muodostetun XML-dokumentin HTML-dokumenttina. HTML-dokumentin esitystapa ei ollut täysin moitteeton. XSL-tyylimäärityksiä muokkaamalla ja molemmat komennot uudelleen läpikäymällä muutin HTML-dokumentin esitystapaa.

6.2.4 Metatiedon liittäminen XML-dokumenttiin

Yhtenäisen dokumentin muodostaminen entiteeteillä onnistui ohjelmilla xslj ja Jade. Sen vuoksi päätin kokeilla myös metatiedon liittämistä entiteettinä dokumenttiin. Täytin ensiksi Dublin Core-metatietolomakkeen (Dublin Core - tallennusalusta, 1998) raporttia koskevilla tiedoilla. Sain lomakkeen, joka kehoitettiin liittämään editorilla Web-sivun alkuun. Minä tallensin sen tekstitiedostona, XML-muodossa, samaan hakemistoon, missä kokeiluissa käyttämäni ohjelmat ja dokumentit olivat.

Viittauksen metatietoentiteetin yhdistämisestä dokumenttiin sijoitin tyylimääritykseen, `kokodoc.xsl`. Jos metatietoentiteettiviittaus olisi sijoitettu kokodoc.xml-dokumentin alkuun, se olisi toimiakseen vaatinut seuraavia toimenpiteitä:

1) Kokodoc.xml-dokumentissa olisi pitänyt määritellä se DTD, jonka mukaisesti metatietodokumentti (Dublin Core-metatietodokumentti) on tehty.

2) Tyylimäärityksessä olisi pitänyt määritellä kaikkien metatietoentiteetin elementtien ja attribuuttien käsittely niin, että ne eivät näy HTML-dokumentissa.

3) Tyylimäärityksessä olisi oltava keinot määritellä tarkasti, mihin kohtaan HTML-dokumenttia metatietoentiteetti sijoitetaan.

Tyylimäärityksessä viittasin metatietoentiteettiin seuraavasti: Määrittelin tyylimäärityksen alussa, että xsl.dtd:hen kuuluu entiteetti `meta`, ja lisäsin entiteettiviiteen `&meta;` alku- ja lopputunnisteen `<HEAD>` väliin:

```
<?XML version='1.0' RMD='none'?>
<!DOCTYPE xsl SYSTEM "xsl.dtd"[
<!ENTITY meta SYSTEM "meta.xml">]
...
          <HEAD>
                &meta;
                <TITLE>...</TITLE>
          </HEAD>
...
```

Ensimmäisen xslj-komennon jälkeen virheilmoitus valitti metatietoentiteetin:

- attribuuteista
- umlauteista
- lopputunnisteiden puuttumisesta
- elementistä LINK

Muutin metatietoentiteetissä (Dublin Core-metatietodokumentissa) attribuuttien nimet pieniksi kirjaimiksi ja lisäsin tyylimäärityksen rakennemääritykseen ä- ja ö-kirjaimien määrittelyt. Tämän jälkeen xslj-ohjelman virheilmoitus valitti lopputunnisteiden puuttumisesta, LINK-elementistä ja sen attribuuteista. Yritin vielä yhdistää metatietoentiteetin dokumenttiin lisäämällä ylläolevaan tyylimääritykseen `literal`-elementin osoittamaan, että `meta`-niminen entiteetti sijoitetaan osoitettuun paikkaan dokumentissa:

```
...
          <HEAD>
                <literal>&meta;</literal>
                <TITLE>...</TITLE>
          </HEAD>
...
```

Tämäkään ei auttanut. Virheilmoituksessa sanottiin, ettei DTD salli metatieto-entiteettiä tähän, ja valitti samoista asioista kuin aikaisemminkin. Kokeilu loppui tähän.

Tällaisen Dublin Core-metatiedon liittäminen XML-dokumenttiin ei onnistu xslj-ohjelmalla. Liittäminen saattaisi onnistua, jos metatietodokumentti olisi htmlfos.dtd:n mukainen. XSL-tyylikielen DTD, xsl.dtd, sisältää htmlfos.dtd:n. Ohjelma xslj puolestaan tukee XSL-tyylikieltä. Tällöin xslj-ohjelma tietää, kuinka htmlfos.dtd:n mukaisia dokumentteja käsitellään. Dublin Core-metatietodokumentissa oli elementti LINK, jota ei ole määritelty htmlfos.dtd:ssä. Sen lisäksi siitä puuttuivat XML-standardin pakolliset lopputunnisteet. Nämä olivat syitä, joiden vuoksi Dublin Core-metatiedon liittäminen XML-dokumenttiin ei onnistunut xslj-ohjelmalla.

6.3 Johtopäätöksiä XML:n ja XSL:n käytöstä

Tehtäessä johtopäätöksiä XML-standardiperheestä näiden kokeilujen perusteella, täytyy huomioida muutama johtopäätöksiin vaikuttava asia. Tein kokeiluja standardiperheellä, joka ei ole täysin valmis. Yhtä standardiperheen jäsentä, XLinkkiä, en voinut kokeilla ollenkaan, koska sitä tukevia ohjelmia ei vielä ole. Tyylistantardikaan, XSL, ei ole vielä täysin valmis. Sen lisäksi kokeilumateriaalina oli vain tekstimateriaalia. Tekstimateriaalin käsittelyssä DTD:n merkitys korostui. Dokumentin kokoaminen sisällöltään erilaisista dokumenteista, kuten kuvista, äänestä ja videosta, olisi ollut toisenlainen prosessi. Kokeilussa oli mukana vain kolme ohjelmaa, joista kaksi kuuluu kiinteästi yhteen. Toisenlaisilla ohjelmilla tulokset olisivat voineet olla toisenlaisia ja/tai samoihin tuloksiin olisi päästy erilaisten vaiheiden kautta. Metatiedon osalta johtopäätöksiä pelkistää se, että kokeilussa oli mukana vain yksi tapa määrittää metatieto.

XML-työkalut

XML-työkalut vaikuttavat olennaisesti XML-dokumenttien tuottamiseen ja hallintaan. XML-työkalujen valikoima on tällä hetkellä vielä hyvin suppea. XML-dokumenttien tuottamiselle onkin kaksi erilaista tapaa. XML-dokumentteja voi tuottaa joko puhtaasti XML-työkalujen avulla suoraan. Julkisesti saatavilla olevia ja maksullisia työkaluja löytyy jonkun verran tälläkin hetkellä. Toinen tapa on sama, jota minä käytin, eli muuntaa dokumentit SGML:n avulla XML:ksi. SGML-dokumenttien tuottamiseksi on jo olemassa erilaisia ohjelmia.

Dokumenttien muuntaminen XML-dokumenteiksi SGML:n avulla ei tietystikään ole ajatuksena tarkoituksenmukainen. Tällä hetkellä se on hyvä ja nopea tapa tehdä kokeiluja. Ohjelma, joka käyttää rakennetta ohjaavaa editoria, on käyttökelpoinen dokumenttien merkkauksessa, mutta se edellyttää määriteltyä DTD:tä. Toisaalta, vaikka dokumentit haluttaisiin tallentaa ilman DTD:tä, sen rakenteen suunnittelu hierarkiseksi, elementtien nimeäminen sekä dokumenttien merkkaaminen käy nopeammin, jos käytössä on jokin DTD.

Käytettävissä ollut ArborTextin XSL-tyylieditori XML Styler, versio 1.0, oli hankala käyttää. Sitä oli vaikea oppia käyttämään, eikä käyttöoppaasta ollut mitään apua. XML Stylerista ilmestyi vapaasti ladattava uusi versio 2.0 sen jälkeen kun olin kokeiluni tehnyt. Tyylimäärittelyt oli helppo ja nopea kirjoittaa W3C:n A Proposal for XSL (1997) ja Microsoftin XSL Tutorialin (1998) avulla msxsl-ohjelmalle. Ohjelmat xslj ja Jade vaativat enemmän aikaa ja huolellisempaa perehtymistä tyylimäärityksiin. Ne vaativat myös kaikkien dokumentissa olleiden elementtien ja attribuuttien määrittelyn, eivätkä nämä ohjelmat tukeneet kuin rajoitettua määrää HTML-tunnisteita. Tyylimäärityksiä kirjoittaessani opin ymmärtämään, mitä dokumenteille tyylimäärityksillä tapahtuu, ja miten niistä tulee esitettäviä dokumentteja.

Yhteenvetona voidaan XML-työkaluista todeta, että tällä hetkellä julkisesti saatavilla olevat ohjelmat ovat hankalia käyttää. Niiden käyttöliittymät ovat kehittymättömiä, ja ohjelmat toimivat puutteellisesti. Ohjelmien käyttö vaatii hyvää

perehtyneisyyttä XML-standardiin, ehdotukseen XSL-standardiksi sekä xsl.dtd:hen. Ohjelmiin ei ole kunnon käyttöoppaita. Vaikka ohjelmia onkin sekä julkisesti että maksullisesti saatavilla, valinnanvaraa on vähän.

Rakennemäärittelyn merkitys käytännössä

Rakennemäärittelyn eli DTD:n merkitys korostui tässä kokeilussa. Ilman rakennemäärittelyä muodostettujen XML-dokumenttien käyttö onnistuu silloin kun kyse on itsenäisistä dokumenteista ts. XML-dokumentit eivät ole minkään toisen dokumentin entiteettejä. Käytettävän ohjelman on tällöin oltava sellainen, että se mahdollistaa dokumenttien käsittelyn ilman DTD:tä. Tässä kokeilussa msxsl-ohjelma mahdollisti hyvin muodostetut XML-dokumentit kun taas xslj- ja Jade -ohjelma eivät. Ohjelmalla msxsl liitin hyvin muodostettuihin dokumentteihin tyylimääritykset ja sain niistä näytöllä esitettäviä HTML-dokumentteja. Hyvin muodostetut dokumentit voivat myös sisältää yksittäisiä entiteettiviitteitä, kuten esimerkiksi entiteetti `&auml;` osoittamaan ä-kirjainta.

Jos halutaan muodostaa kokonainen dokumentti entiteeteistä, on entiteettien oltava jonkin rakennemäärittelyn mukaisia tai juuressa on määriteltävä rakennemäärittely. Tätä vaativat molemmat kokeilussa olleet ohjelmat. XML-dokumentin muodostaminen useasta entiteetistä onnistui vain xslj- ja Jade-ohjelmilla. Yhtenäisen dokumentin muodostaminen oli monivaiheinen prosessi. Se edellytti tässä kokeilussa DTD:n määrittelyä juureen. Sen lisäksi dokumentin esitystavan muokkaaminen vaatii perehtyneisyyttä ja aikaa, koska kaikki dokumentissa olevat elementit ja attribuutit oli määriteltävä tyylimäärityksessä.

XSL-tyylimäärityksistä

XSL-tyylimääritysten liittäminen XML-dokumenttiin ja XML-dokumentin muuntaminen niiden avulla näytöllä esitettäväksi HTML-dokumentiksi ei ollut periaatteessa ongelmallista. Eri ohjelmissa siihen liittyi eri vaiheita ja eri asioiden huomioimista.

Tyylimääritys oli tässä kokeilussa tallennettuna samassa tiedostossa kuin XLM-dokumentti. Se oli XML-dokumentti, joka oli tallennettu tekstitiedostona muotoon .xsl. XSL-dokumentiin ei sisältynyt mitään viitettä siitä, mihin dokumenttiin kyseiset tyylimääritykset liittyvät. XML- ja XSL-dokumenttit yhdistin kirjoittamalla komentoon kummankin dokumentin nimen.

Tämän kokeilun perusteella, dokumentilla voi olla erilaisia erikseen tallennettuja tyylimäärityksiä. Tyylimääritysten tallentaminen ei vaadi mitään erityistietojen tallentamista. Tyylimääritysten on vain oltava XSL-tyylistandardin syntaksin mukaisia. Käytössä on kuitenkin oltava ohjelma, joka pystyy yhdistämään dokumenttit ja tyylimääritykset, ja muuntamaan ne esitettävään muotoon.

Metatieto XML-dokumentissa

Metatiedon liittäminen XML-dokumenttiin oli ongelmallista, ja vaatii enemmän ja erilaisia kokeiluja. Tässä kokeilussa metatieto oli erillinen dokumentti, joka yritettiin liittää XML-dokumenttiin entiteettinä. Metatietodokumentti ei alunperin ollut XML-dokumentti. Kokeilujen tuloksena ilmeni kuitenkin, että metatietodokumentin tulisi olla XML-standardin mukaista. Siihen tulisi liittyä myös tietoa rakennemäärittelystä, DTD:stä, jolla sen rakenne on määritetty. Silloin ainakin ohjelma, joka vaatii dokumenteilta rakennemäärityksen toimiakseen, voisi käsitellä metatiedon. Toisaalta metatiedon liittäminen XML-dokumenttiin voidaan tulevaisuudessa ratkaista kehitteillä olevan linkkistandardin tai metatietosovellusten, kuten RDF:n, avulla.

XML- ja XSL-dokumenttien arkistoinnista

Tämän hetken informaatioteknologian vuoksi arkiston täytyy tallentaa XML-dokumenteista XSL-tyylimääritysten avulla muodostetut HTML-dokumentit. Toinen vaihtoehto on tallentaa ne ohjelmat, jotka yhdistävät varsinaiset dokumentit ja tyylimääritykset ja muodostavat niistä esitettävät dokumentit. Tämän hetken selaajat eivät vielä tue XML-standardiperhettä niin hyvin, että arkisto

voisi tyytyä pelkkien XML-dokumenttien ja XSL-tyylimääritysten arkistointiin. Pitkäaikaista säilytystä ajatellen ohjelmien rakentamisen dokumentointi XML-muodossa mahdollistaisi näiden ohjelmien saatavuuden tulevaisuudessakin.

Käytännön kokeilut osoittivat XML-standardiperheen käytön olevan vielä hyvin ohjelmasidonnaista. Yhdellä ohjelmalla voidaan käsitellä XML-dokumentteja tavalla, mikä ei toisella ohjelmalla olekaan mahdollista. Ohjelmien tallentaminen onkin erityisesti arkistoinnin kannalta olennaista. Itse dokumenteissa joudutaan ohjelmistoriippuvuuden vuoksi huomioimaan XML-standardin vaatimusten lisäksi ohjelmien erityiset vaatimukset.

7 YHTEENVETO

Tarkastelin tässä työssä uutta ja osittain kehitteillä olevaa standardiperhettä XML:ää. Rajasin XML:n tarkastelun arkistoinnin näkökulmaan. Arkistoinnilla tarkoitan dokumenttien pitkäaikaista säilytystä siten, että dokumentit löytyvät tarvittaessa arkistosta. XML-standardiperhettä ja arkistointia tarkastelin kirjallisuuden avulla. Käytännön kokeilla sain lisätietoa XML:stä vaikka linkkistandardia ei vielä voinutkaan kokeilla.

Tässä yhteenvedossa esittelen tiivistelmän tuloksista, joita sain sekä analysoimalla arkistointia ja XML-standardiperhettä että käytännön XML-kokeiluilla. Sen jälkeen arvioin tutkimuksen tekotapaa ja esittelen jatkotutkimusmahdollisuuksia.

Tiivistelmä tuloksista

Arkistoitaessa dokumentteja XML-dokumentteina voidaan valita arkistointi joko rakennemäärittelyn kanssa tai ilman sitä. Valittaessa rakennemäärittelyä, joudutaan ratkaisemaan, mitä rakennemäärittelyä käytetään, ja käytetäänkö arkistossa yhtä vai useampaa rakennemäärittelyä. Rakennemäärittelyä käytettäessä joudutaan tiedonvälityksen varmistamiseksi välittämään rakennemäärittely dokumenttien kanssa. Rakennemäärittely kuitenkin helpottaa dokumenttien rakenteen suunnittelua ja mahdollistaa dokumenttien rakenteen muuttumattomuuden toteamisen. Ilman rakennemäärittelyä tiedonvälitys on yksinkertaisempaa, koska välitettävänä on vain XML-dokumentti. Dokumenttien rakenteistaminen on kuitenkin hitaampaa eikä dokumenttien rakenteen muuttumattomuus ole niin helposti todettavissa.

Dokumenttien sisältö voidaan jakaa entiteetteihin, osiin. Dokumenttien eri osat, sisältö, rakenne, linkit ja tyylit voidaan erottaa toisistaan. Tämä mahdollistaa näiden osien ylläpidon. Dokumenttien ylläpito arkistossa on välttämätöntä, koska dokumentit saattavat muuttua elinkaarensa aikana, ja dokumenttien tal-

lennusmedia muuttuu tekniikan kehittymisen myötä. Dokumenttien käyttörajoitukset, niiden muuttumattomuutta ja muutosoikeutta osoittavat tekniikat voidaan myös yksiselitteisesti määritellä eri osille.

XLink-standardin mukaiset linkit voidaan tallentaa erikseen varsinaisesta dokumentista. Ne ovat kaksisuuntaisia linkkejä. XLink-standardin mukaisten linkkien ylläpito on mahdollista. XSL-tyylistandardilla saadaan haluttu esitystapa ja vaihtoehtoiset esitystavat dokumenteille ilman että se haittaa dokumenttien säilyvyyttä.

Dokumenteille joudutaan määrittämään verkkoympäristössä monenlaista metatietoa. Metatieto auttaa dokumenttien löytymistä ja hakumenetelmien kehittämistä. Metatiedon määrittelemiseksi XML-dokumentista on kolme erilaista tapaa. Metatieto voidaan määritellä suoraan XML-dokumenttien rakenteesta ja hyödyntää sitä siitä käsin. Rakennemäärittelyssä on tällöin määritelty erilliset metatietoelementit ja ne dokumentin loogisen rakenteen elementit ja attribuutit, joita käsitellään metatietona. Toinen tapa on liittää XML-dokumenteihin metatietodokumentti, jossa metatieto määritetään. Se on itsenäinen XML-dokumentti, joka voidaan liittää varsinaiseen dokumenttiin entiteettinä. Metatietodokumentti on mahdollista liittää dokumenttiin myös linkillä. Kolmas tapa on hyödyntää kehitteillä olevia metatietosovelluksia kuten RDF:ää.

XML tarjoaa myös laitteisto- ja ohjelmistoriippumattoman, aina tulkittavissa olevan, tallennusmuodon. XML on kiinnittänyt merkkijärjestelmänsä UNICODEen, joka tukee lähes kaikkien kielten merkkijärjestelmiä.

Käytännön kokeiluissa oli mukana kolme ohjelmaa, Microsoftin msxsl sekä xslj (Thompson, 1998) ja Jaden versiota 1.1.1. (Clark, 1998). XLink-standardin antamien mahdollisuuksia kokeilu ei vielä ole mahdollista, koska sitä tukevia ohjelmia ei ole. Käytännön kokeilujen avulla paljastui, että XML:n käyttö on vielä hyvin ohjelmistoriippuvaa: se mitä toisella ohjelmalla pystyi tekemään ei ollutkaan mahdollista toisella. Julkisesti saatavilla olevia XML-työkaluja on vielä vähän.

Käytännön kokeiluissa muunsin MS Word (versio 6.0) -dokumentin XML-dokumentiksi, liitin siihen XSL-tyylimääritykset ja muodostin niistä HTML-dokumentin näytöllä esitettäväksi. Tämä onnistui molemmilla ohjelmilla (xslj:tä ja Jadea käytettään yhdessä). XML-dokumentin muodostaminen entiteeteistä ja sen esittäminen HTML-dokumenttina ei onnistunut msxsl-ohjelmalla mutta onnistui xslj- ja Jade-ohjelmilla. En voinut liittää XML-dokumenttiin metatietoa, joka oli itsenäinen entiteetti, xslj- ja Jade -ohjelmalla. Käytössäni ollut metatietodokumentti ei ollut XML-dokumentti eikä siinä ollut viitettä käytetystä DTD:stä.

Käytännön kokeilujen tuloksena saatiin, että XML- ja XSL-dokumenttien arkistoiminen erillisinä dokumentteina on mahdollista. Ne voidaan liittää toisiinsa ja niistä saadaan HTML-dokumentti, jos käytössä on siihen sopiva ohjelma. XML-dokumentti voidaan arkistoida entiteetteinä. Jos metatieto halutaan liittää XML-dokumenttiin, sen on oltava XML-standardin mukainen ja siinä käytetty DTD on oltava tiedossa. Koska XML:n käyttö on ohjelmistoriippuvaa, XML- ja XSL-dokumenttien arkistointi edellyttää myös sellaisten ohjelmien arkistointia, jotka pystyvät tekemään halutunlaisen liittämisen. Toinen vaihtoehto on arkistoida XML-dokumenttien ohelle niistä muodostetut HTML-dokumentit.

Käytännön kokeilujani rajoitti linkkistandardia tukevien ohjelmien puute. Sain kuitenkin kokeiltua kahdella erilaisella ohjelmalla XML:n käyttöä. Oman oppimiseni ja ymmärtämisen kannalta näillä kokeilla oli suuri merkitys. Ne myös vahvistivat sitä tietoa, mitä olin XML:n määrittelystä lukenut. Standardiperheen valmistuttua ja käytössä olevien ohjelmien monipuolistuttua erilaisilla käytännön kokeiluilla on varmasti suurempi osuus standardiperhettä koskevissa tutkimuksissa.

Työni tulokset ovatkin lähinnä kuvailevaa tietoa XML:stä. Tämä työ antaa viitteitä, miten XML:ää ja arkistointia tai jotakin muuta XML:n sovellusaluetta saatetaan tutkia. Tämän työn arvo on enemmänkin siinä, että tämä antaa materiaalia ratkaistavien ongelmien muotoilemiseen ja tutkittavien kohteiden täsmen-

tymiseen. Päätösten teon pohjaksi tämän työn tulokset eivät tällaisenaan ole riittävät. Käytännön kokeiluja tarvitaan paljon enemmän.

Jatkotutkimusmahdollisuuksia

Yleisesti ottaen XML-standardiperhe on uusi ja osittain vielä kehitteillä oleva standardi. Tutkimusmahdollisuuksia itse standardiperheessä on paljon. Standardiperheen laajennuksiakin on jo ilmestynyt, kuten Namespaces, XML-Data skeema ja DOM. Jatkuvasti ilmestyvät XML-sovellukset antavat myös hyviä mahdollisuuksia kartoittaa XML-standardiperheen olemusta.

Erityisen mielenkiintoisia veratailuasetelmia saa asettamalla vastakkain XML:n ja SGML:n, HyTimen ja XLinkin, DSSSL/CSS:n ja XSL:n. Vertailemista riittää näiden ominaisuuksissa: kuinka paljon ne itseasiassa eroavat toisistaan ja mitä ne antavat dokumenteille, jota toinen ei voi antaa. Standardien eroista saa tietoa myös käytettävyystutkimuksilla vertaamalla standardin käyttöä tekijän vs. käytttäjän kannalta, soveltuvuutta erilaisiin sovelluksiin ja erilaiseen kontekstiin, kuten liike-elämään, julkishallintoon, viihdeteollisuuteen, teollisuuteen, tutkimukseen ja tieteeseen. Tässä työssä koin ongelmalliseksi sen, etten pystynyt keskittymään pelkästään XML:n ja SGML:n dokumenteille antamien ominaisuuksien vertailuun. Se olisi antanut mielestäni työlle sitä hyödynnettävyyttä ja dynaamisuutta, joka siitä nyt puuttuu.

SGML-sovelluksissa dokumenttien ominaisuuksista korostui rakennemäärittelyn merkitys. Rakennemäärittely on tärkeä ominaisuus mm. sekä tiedonvälityksessä että dokumenttien löytymisessä. XML:ssä dokumenttien rakenne voidaan määritellä rakennemääritelyn kanssa tai ilman sitä. Tähän sisältyykin paljon erilaisia selvitysmahdollisuuksia, koska XML-dokumentin kokoaminen entiteeteillä ei onnistunut ilman rakennemäärittelyä. Selvitettävä olisi, milloin kannattaa käyttää rakennemäärittelyä ja milloin olla ilman, mihin rakennemäärittely tai sen puuttuminen todellisuudessa vaikuttaa. Jos XML-Data skeemasta tulee sellainen dokumenttien rakenteen määrittelymahdollisuus, jota myös ohjelmat

tukevat, tarjoaa se vaihtoehdon DTD:lle. Siitä myös syntyy hyviä tutkimusmahdollisuuksia DTD:n ja XML-Data skeeman vertaamiseksi.

Tämä työ herätti myös filosofisempiakin kysymyksiä. Mieleeni nousi kysymys, hyödyntävätkö rakenteiset dokumentit todellakin verkkoa optimaalisella tavalla, vai onko kyseessä kuitenkin käsitteen paperidokumentti siirtäminen verkkoon rakenteisen dokumenttien muodossa? Toisaalta kysymyksen voisi esittää myös niin päin, että kahlitseeko käsitys paperidokumentista niin, ettemme osaa kehittää dokumenteille standardeja, jotka hyödyntäisivät verkkoa optimaalisesti ja kehittäisivät myöskin sitä? Mitä uutta DOM, täysin uudenlainen dokumenttien osien esittäminen näytöllä, tuo dokumenttikäsitteeseen?

Arkiston näkökulmasta on myös paljon tutkittavaa. Metatiedon liittäminen on vielä hyvin epäselvä asia. Linkkejä ei tässä työssä pystytty kokeilemaan ollenkaan. Rakenteeseen liitetty metatieto olisi vaatinut isomman tutkimushankkeen tai keskittymisen pelkästään siihen. Entiteetillä metatiedon liittämistä kokeiltiin, eikä se onnistunut. Metatiedon liittäminen tarvitsee syvällisempää tutkimusta. XML-dokumenttien metatiedon käytettävyystutkimuksista saataisiin tietoa siitä, mistä kaikista XML-dokumenttien osista, entiteetit, rakennemäärittely, linkit, sisältö ja dokumentteihin liittyvä muu tieto, metatietoa olisi määriteltävä.

XML-standardiperheen ominaisuuksien tarkastelu arkistoinnissa selkeästi toimijan näkökulmasta, esimerkiksi arkistoon dokumenttaja arkistoitavaksi hakevan ja tallentavan, arkistoa ylläpitävän ja arkistoon dokumentteja tarjoavan näkökulmasta antaisi hyödynnettävää tietoa XML-sovellusten rakentamiseen. Arkistoa käyttävien kokemusten kartoituksesta saataisiin tietoa tekijänoikeuksien mukaisten käyttörajoitusten ja muuttumattomuus- sekä muutoksentekooikeus tekniikoiden toimivuudesta, jota voitaisiin hyödyntää verkossa olevien arkistojen luotettavuuden kehittämiseen. Myös tiedonhakutapojen, hakukoneet ja navigointi, dokumenteille asettamien vaatimusten tarkastelu toisi lisää tietoa XML-standardiperheen ominaisuuksista arkistoinnin näkökulmasta.

XML:ään on siis helppo liittää vaikka minkänlaista tutkimusta, koska se on uusi standardi, josta ei vielä tiedetä paljon mitään. Mielenkiintoista olisikin saada selville, mitä uutta XML-standardiperhe tuo dokumenttien hallintaan, ja mitkä mahdollisesti ovat sen puutteet.

Lopuksi

Tutkimusaiheena XML on hyvinajankohtainen. Uutta tietoa tuli koko ajan lisää, ja se oli jätettävä tämän työn ulkopuolelle, kuten esimerkiksi linkkistandardin uusin versio, joka ilmestyi tämän työn viimeistelyvaiheessa. Minun oli pakko vetää raja käsiteltäviksi otettavista asioista johonkin, jotta sain työni valmiiksi.

Toisaalta tämä aihe oli minulle jopa liian laaja näin lyhyessä ajassa tehtäväksi opinnäytetyöksi. Jos hallitsee edes osan XML-standardiperheen taustalla olevista standardeista kohtalaisesti, tämä aiheen laajuus olisi ihan hyvä. Nyt täytyi liian lyhyessä ajassa omaksua paljon erilaisia standardeja, niiden ominaisuuksia ja niiden välisiä suhteita. Tämän lisäksi mukana oli vielä täysin uuden ja tuntemattoman käsitteen "arkistointi" selvittely ja omaksuminen. Tuntuu edelleenkin siltä, että pelkkä XML:n tarkastelu olisi ollut tarpeeksi laaja. XLinkin ja XSL:n olisi voinut jättää pois. Tällöin opinnäytetyöstä olisi ehkä tullut yhtenäisempi kokonaisuus. Olisin voinut paremmin keskittyä XML:n ominaisuuksiin, sekä sen ja SGML:n erojen ja yhtäläisyyksien pohdiskeluun. Ehkä silloin olisin saanut arkistoinninkin käsittelyyn syvyyttä ja asiantuntemusta. Tämmöiseen syvällisempään pohdiskeluun ei nyt jäänyt aikaa, kun piti huomioida niin monen asian käsittely, ja saada työ kohtuullisessa ajassa valmiiksi.

Aihe oli kuitenkin minulle mitä täydellisin. Minulla oli mahdollisuus oppia XML-standardiperheeseen tutustumisen ohella paljon dokumenttien hallintaa liittyvistä muista standardeista, kuten SGML, HyTime, DSSSL, CSS. Sen lisäksi minulle tarjoutui mahdollisuus ymmärtää, mistä dokumenttien hallinnassa oikein on kyse.

LÄHTEET

A Proposal for XSL [online]. W3C, 1997 [viitattu 1.12.1997]. NOTE-XSL.html. Saatavilla www-muodossa: <URL:http://www.w3.org/TR/NOTE-XSL-970910>.

Al-Shamma, N., Ayers, R., Cohn, R., Ferraiolo, J., Newell, M., de Bry, R.K., McCluskey, K., Evans, J.: Precision Graphics Markup Language (PGML) [online]. W3C, 1998 [viitattu 23.4.1998]. W3C Note 10-April-1998 NOTE-PGML-19980410. Saatavilla www-muodossa: <URL:http://www.w3.org/TR/1998/NOTE-PGML-19980410.html>.

ArborText's XML Styler [online]. ArborText, 1998 [viitattu 23.5.1998]. Saatavilla www-muodossa: <URL:http://www.arbortext.com/xmlstyler/>.

Baker, D.W. :A Guide to URLs [online]. 1995 [viitattu 20.11.1997]. Saatavilla www-muodossa: <URL:http://www.ust.hk/ccst/techinfo/webdevelop/html/url-guid.htm>.

Berner-Lee, T., Masinter, L. & McCahill, M.: Uniform Resource Locators (URL) [online]. 1994 [viitattu 20.11.1997]. Saatavilla www-muodossa: <URL:http://ds.internic.net/rfc/rcf1738.txt>.

Bosak, J.: XML, Java, and the future of the Web [online]. 1997 [viitattu 1.9.1997]. Saatavilla www-muodossa: <URL:http://sunsite.unc.edu/pub/sun-info/standards/xml/why/xmlapps.htm>.

Bray, T. & DeRose, S.: Extensible Markup Language (XML): Part 2. Linking [online]. W3C, 1997 [viitattu 2.10.1997]. W3C Working Draft July-31-97. Saatavilla www-muodossa: <URL:http://www.w3.org/TR/WD-xml-link>.

Bray, T. & Guha, R.V.: An MCF Tutorial [online]. W3C, 1997 [viitattu 15.12.1997]. Saatavilla www-muodossa: <URL:http://www.w3.org/TR/NOTE-MCF-XML/MCF-tutorial.html>.

Bray, T., Hollande, D. Layman, A.: Namespaces in XML [online]. W3C, 1998 [viitattu 20.4.1998]. Working Draft by the W3C. Saatavilla www-muodossa: <URL:http://www.microsoft.com/xml/>.

Bray, T., Paoli, J. & Sperberg-McQueen C.M.: Extensible Markup Language (XML) [online]. W3C, 1997 [viitattu 9.9.1997]. W3C Working Draft 07-Aug-97. Saatavilla www-muodossa: <URL:http://www.w3.org/TR/WD-xml-970807>.

Bray, T., Paoli, J. & Sperberg-McQueen C.M. Extensible Markup Language (XML) 1.0 [online]. W3C, 1998 [viitattu 19.3.1998]. W3C Recommendation 10-

February-1998. Saatavilla www-muodossa:
<URL:http://www.w3.org/TR/REC-xml>.

Byrne, S.: Document Object Model (Core) Level 1 [online]. W3C, 1997 [viitattu 28.10.1997]. W3C Working Draft 9-October-1997 WD-DOM/level-one-core-971009. Saatavilla www-muodossa: <URL:http://www.w3.org/TR/WD-DOM/level-one-core971009>.

Böhm, K., Aberer, K., Neuhold, E.J. & Yang, Xiaoya. Structured document storage and refined declarative and navigational access mechanisms in HyperStorM. The VLDB Journal, 1997, 6, 296-311.

Cathro, W.: Metadata: An Overview [online]. 1997 [viitattu 28.11.1997]. Esitys seminaarissa "Matching Discovery and Recovery", Austraalia. Saatavilla www-muodossa: <URL:http://www.nla.gov.au/nla/staffpaper/cathro3.html>.

Clark, J.: Jade - James' DSSSL Engine [online]. 1998 [viitattu 20.5.1998]. Saatavilla www-muodossa: <URL:http://www.jclark.com/jade/>.

Daniel, R. Jr., Ianella, R. & Miller, E. : Expressing the Dublin Core in the Resource Description Framework: Suggestions based on an early examination of the problem [online]. 1997 [viitattu 19.12.1997]. Saatavilla www-muodossa: <URL:http://www.acl.lanl.gov/~rdaniel/RDF/DC/ExpDC_2.html>.

Day, M.: Extending metadata for digital preservation [online]. 1997 [viitattu 4.1.1998]. Saatavilla www-muodossa:
<URL:http://www.ariadne.ac.uk/issue9/metadata/>.

DeRose, S.J., The SGML FAQ Book. Understanding the Foundation of HTML and XML. Kluwer Academic Publishers, Norwell, Massachusetts, 1997.

Desai, B.C.: Indexing and Searching Virtual Libraries [online]. 1995 [viitattu 23.1.1998]. The white paper prepared for CIC Forum: America in the Age of Information, Bethesda, MD, July 1995. Saatavilla www-muodossa:
<URL:http://www.cs.concordia.ca/~facult.../forum95/forum95-bcd-Importan.html>.

Desai, B.C., Supporting Discovery in Virtual Libraries. Journal of the American Society for Information Science 1997, 48, 190-204.

DSSSL Online Application Profile [online]. 1996 [viitattu 12.12.1997]. Saatavilla www-muodossa: <URL:http://sunsite.unc.edu/pub/sun-info/standards/dsssl/dssslo/do960816.htm>.

Dublin Core Metadata [online]. Last Modified 1997-11-02 [viitattu 31.5.1998]. Saatavilla www-muodossa:
<URL:http://purl.oclc.org/metadata/dublin_core/>.

Dublin Core -tallennusalusta [online]. 1998 [viitattu 18.5.1998]. Saatavilla www-muodossa: <URL:http://hul.helsinki.fi/cgi-bin/dc.pl>.

Extensible Markup Language [online]. 1997 [viitattu 22.9.1997]. Saatavilla www-muodossa:
<URL:http://www.microsoft.com/standards/xml/xmlintro.htm>.

Fausey, J. & Shafer, K., All My Data Is in SGML. Now What? Journal of the American Society for Information Science, 1997, 48 (7), 638-643.

Garshol, L.M.: Introduction to XML [online]. 1997 [viitattu 1.9.1997]. Saatavilla www-muodossa:
<URL:http://www.ifi.uio.no/~larsga/download/xml/xml_eng.html>.

Goldfarb, C.F., The SGML Handbook. Oxford University Press. Inc., New York, 1990.

Guha, R.V. & Bray, T.: Meta Content Framework Using XML [online]. 1997 [viitattu 10.12.1997]. Saatavilla www-muodossa:
<URL:http://www.w3.org/TR/NOTE-MCF-XML/>.

Heery, R.: Metadata Corner. Naming names: metadata registries. Julkaisussa: Ariadne (Web version) [online], 1997, N:o 11 [viitattu 31.5.1998]. Saatavilla www-muodossa: <URL:http://www.ariadne.uk/issue11/metadata>.

Homer, M.: Netscape Open Standards Guarantee [online]. Netscape, 1997 [viitattu 4.11.1997]. Saatavilla www-muodossa:
<URL:http://www.netscape.com/flash5/com...lumns/intranet/open_standards.htm>.

Hopmann, A.: Web Collections using XML [online]. W3C, 1997 [viitattu 15.12.1997]. Saatavilla www-muodossa:
<URL:http://www.w3.org/TR/NOTE-XMLsubmit.html>.

Hosschka, P.: Synchronized Multimedia Integration Language [online]. W3C, 1997 [viitattu 24.11.1997]. Saatavilla www-muodossa:
<URL:http://www.w3.org/TR/WD-smil>.

ISO/IEC 10179:1996.: International Organization of Standardization, Information Technology - Text and office systems - Document Style Semantics and Specification Language (DSSSL) [online]. 1996 [viitattu 12.12.1997]. Saatavilla www-muodossa: <URL:http://0ccam.sif.novell.com:8080/dsssl/dsssl96>.

Khare, R. & Rifkin, A.: X Marks the Spot. eXtensible Markupp Language opens the door to the motherlode of automated Web applications [online]. 1997 [viitattu 1.9.1997]. Saatavilla www-muodossa:
<URL:http://www.cs.caltech.edu/~adam/papers/xml/x-marks-the-spot.html>.

Kipp, N.A., Will XML-Linking Be Useful for Me - As I'm Building My Very Own Digital Library?. The SGML Newsletter <TAG>, 1997, 10 (12), 4-7.

Lander, R.: XML: The New Markup Wave [online]. 1997 [viitattu 6.10.1997]. Saatavilla www-muodossa:
<URL:http://www.csclub.uwaterloo/u/relander/Academic/XML/xml_html>

Lassila, O.: PICS-NG Metadata Model and Label Syntax [online]. W3C, 1997 W3C [viitattu 10.12.1997]. NOTE 1997-05-14. Saatavilla www-muodossa:
<URL:http://www.w3.org/TR/NOTE-pics-ng-metadata>.

Lassila, O. & Swick, R.R.: Resource Description Framework (RDF). Model and Syntax [online]. W3C, 1997 [viitattu 10.12.1997]. Saatavilla www-muodossa:
<URL:http://www.w3.org/TR/WD-rdf-syntax/>.

Layman, A., Jung, E., Maler, E., Thompson, H.S., Paoli, J., Tigue, J., Mikula, N.H. & De Rose, S.: XML-Data [online]. Microsoft, 1997 [viitattu 20.4.1998]. Saatavilla www-muodossa:
<URL:http://www.microsoft.com/standards/xml/xmldata.htm>.

Lie, H.W.: CSS Printing Extensions [online]. W3C, 1997 [viitattu 12.12.1997]. W3C Working Draft 26-June-97. Saatavilla www-muodossa:
<URL:http://www.w3.org/TR/WD-print>.

Lie, H.W. & Bos, B.: Cascading Style Sheets, level 1 [online]. W3C, 1996 [viitattu 12.12.1997]. W3C Recommendation 17 Dec 1996. Saatavilla www-muodossa:
<URL:http://www.w3.org/TR/REC-CSS1>.

Lynch, C.A., The Integrity of Digital Information: Mechanics and Definitional Issues. Journal of the American Society for Information Science, 1994, 45 (10), 737-744.

Mackenzie Owen, J.S. & Walle, J.v.d., Deposit collections of electronic publications. Libraries in the Information Society. European Commission, DG XIII-E/4. Luxembourg, 1996.

Maler, E. & DeRose, S.: XML Linking Language (XLink) [online]. World Wide Web Consortium, 1998 (viitattu 3.6.1998). World Wide Web Consortium Working Draft 3-March-1998. Saatavilla www-muodossa:
<URL:http://www.w3.org/TR/1998/WD-xlink-19980303>.

Marcoux, Y. & Sévigny, M., Why SGML? Why Now? Journal of the American Society for Information Science, 1997, 48 (7), 584-592.

Nielsen, J.: Publications on the Internet: publication types and a section on access conditions [online]. 1997 [viitattu 19.12.1997]. Saatavilla www-muodossa:
<URL:http://www.purl.dk/rapport/html.uk/part4.htm>.

Noerr, P.L.: Character sets and UNICODE [online]. 1995 [viitattu 10.11.1997]. Saatavilla www-muodossa:
<URL:http://www.ua.ac.be/KB/pn/pnoerr0.html>.

Prescod, P.: Introduction to DSSSL [online]. 1997 [viitattu 12.12.1997]. Saatavilla www-muodossa:
<URL:http://itrc.uwaterloo.ca:80/~papresco/dsssl/tutorial.html>.

Preserving Digital Information [online]. 1996 [viitattu 21.1.1998]. Report of the Task Force on Archiving of Digital Information commissioned by The Commission on Preservation and Access and The Research Libraries Group, Inc. Saatavilla www-muodossa:
<URL:http://www.rlg.org/ArchTF/tfadi.index.htm>.

Powell, A. Dublin Core Management. Julkaisussa: Ariadne (Web version) [online], 1997, N:o 10 [viitattu 31.5.1998]. Saatavilla www-muodossa
<URL:http://www.ariadne.ac.uk/issue10/dublin/>.

Rothenberg, J. Metadata to Support Data Quality and Longlivety [online]. Proceeding of the First IEEE Metadata Conference, 1996 [viitattu 1.6.1998]. Saatavilla www-muodossa
<URL:http://www.computer.org/conferen/meta96/rothenberg_paper/ieee.data-quality.html>.

Rutledge, L.: HyTime: ISO 10744 Hypermedia/Time-based Structuring Language [online]. 1996 [viitattu 3.1.1998]. Saatavilla www-muodossa:
<URL:http://dmsl.cs.uml.edu/standards/hytime.html#PUBLIC>.

Salminen, A., Elektroninen teksti: mitä se on?. SGML-seminaari. Eduskunnan kirjaston seminaari 12.12.1994 ja 14.12.1994. Eduskunnan kirjaston tutkimuksia ja selvityksiä 2. Helsinki, 1995.

Salminen, A.: Hajautettu hypermedia [online]. 1997 [viitattu 24.11.1997]. Saatavilla www-muodossa:
<URL:http://www.cs.jyu.fi/~airi/digmed/tkod54/kasitteet.html>.

Sprague, R.H., Electronic document management:challenges and opportunities for information systems managers. MIS Quarterly, 1995, 19 (1), 29-49.

Sperberg-McQueen, C.M. & Plotkin, W.: Text Encoding Initiative [online]. 1997 [viitattu 3.1.1998]. Saatavilla www-muodossa:
<URL:http://www.uic.edu/orgs/tei/index.html>.

Staflin, L.: Information about PSGML. 1996 [viitattu 23.5.1998]. Saatavilla www-muodossa: <URL:http://www.lysator.liu.se/projects/about_psgml.html>.

Stevahn, R.: Positioning HTML Elements with Cascading Style Sheets [online]. W3C, 1997 [viitattu 12.12.1997]. W3C Working Draft 19-Aug-1997. Saatavilla www-muodossa: <URL:http://www.w3.org/TR/WD-positioning>.

TEI Guidelines for Electronic Text Encoding and Interchange [online]. 1997 [viitattu 3.1.1998]. Saatavilla www-muodossa: <URL:http://etext.virginia.edu/TEI.html>.

The Davenport Group: Maintainers of the DocBook DTD [online]. 1997 [viitattu 25.5.1998]. Saatavilla www-muodossa: <URL:http://www.ora.com/davenport/>.

The Microsoft XSL Processor [online]. Microsoft, 1998 [viitattu 23.5.1998]. Saatavilla www-muodossa: <URL:http://www.microsoft.com/xml/>.

The Unicode Standard [online]. Unicode, 1996 [viitattu 20.11.1997]. Saatavilla www-muodossa: <URL:http://www.unicode.org/unicode/uni2book/u2.html>.

The Unicode Standard: A Technical Introduction [online]. Unicode, 1996 [viitattu 12.5.1998]. Saatavilla www-muodossa: <URL:http://www.unicode.org/unicode/standard/principles.html>.

Thompson, H. S.: xslj: An XSL to DSSSL Translator [online]. University of Edinburgh, 1998 [viitattu 23.5.1998]. Language Technology Group, HCRC, University of Edinburgh. Saatavilla www-muodossa: <URL:http://www.ltg.ed.ac.uk/~ht/xslj.html>.

Veijola, R. Rakenteisen julkaisemisen prosessi. Pro gradu-tutkielma, Tietojenkäsittelytieteiden laitos, Jyväskylän yliopisto, (tulossa 1998).

Wallace, D., Archives and the Information Superhighway: Current Status and Future Challenges. The International Information & Library Review, 1996, 28, 79-91.

XML Overview [online]. Microsoft, 1997 [viitattu 22.9.1997]. Saatavilla www-muodossa: <URL:http://www.microsoft.com/standards/xml.htm>.

XML White Paper [online]. Microsoft, 1997 [viitattu 22.9.1997]. Saatavilla www-muodossa: <URL:http://www.microsoft.com/standards/xml/xmlwhite.htm>.

XSL Tutorial [online]. Microsoft, 1998 [viitattu 10.2.1998]. Saatavilla www-muodossa: <URL:http://www.microsoft.com/xml/xsl/tutorial/tutorial.htm>.

LIITE 1: Ohjelmassa Jade (versio 1.1.1) käytetty juuri

```
<?XML VERSION="1.0" ?>
<!DOCTYPE gradu[
<!ELEMENT gradu ANY>
<!ELEMENT section ANY>
<!ELEMENT para ANY>
<!ELEMENT head ANY>
<!ELEMENT numberedlist ANY>
<!ELEMENT listitem ANY>
<!ELEMENT itemizedlist ANY>
<!ELEMENT terminaltext ANY>
<!ELEMENT example ANY>
<!ELEMENT title ANY>
<!ELEMENT pubyear ANY>
<!ELEMENT lastname ANY>
<!ELEMENT biblioentry ANY>
<!ELEMENT author ANY>
<!ELEMENT authgroup ANY>
<!ELEMENT firstname ANY>
<!ELEMENT info ANY>
<!ELEMENT ulink ANY>
<!ATTLIST ulink url CDATA #REQUIRED>
<!ELEMENT pagenums ANY>
<!ELEMENT subtitle ANY>
<!ELEMENT publisher ANY>
<!ELEMENT textbk ANY>
<!ELEMENT header ANY>
<!ELEMENT name ANY>
<!ELEMENT courseinfo ANY>
<!ELEMENT university ANY>
<!ELEMENT bibliography ANY>
<!ENTITY Auml  CDATA "&#196;" -- capital A -->
<!ENTITY Ouml  CDATA "&#214;" -- capital O -->
<!ENTITY auml  CDATA "&#228;" -- small a -->
<!ENTITY ouml  CDATA "&#246;" -- small o -->
<!ENTITY alku  SYSTEM "1luku.xml">
<!ENTITY luku2 SYSTEM "2luku.xml">
<!ENTITY luvut3456 SYSTEM "3456luku.xml">
<!ENTITY lahteet SYSTEM "Lahteet.xml">
]>
<gradu>
&alku;
&luku2;
&luvut3456;
&lahteet;
</gradu>
```

LIITE 2: Kokeilussa käytetty metatietodokumentti

```
<!-- Alla on lomakkeen tiedoista jäsennetty kuvaus      -->
<!-- 'Liimaa' editorilla kuvailu Web-sivun alkuun       -->
<!-- HEAD-elementin sisälle                             -->
<!-- ---------------------------------------------------- -->
 <META NAME="DC.Title" CONTENT="XMLN k&auml;ytt&ouml; verkkosovelluksissa">
<LINK REL=SCHEMA.dc HREF="http://purl.org/metadata/dublin_core_elements#title">

<META NAME="DC.Creator" CONTENT="Anne Leinonen">
<LINK REL=SCHEMA.dc HREF="http://purl.org/metadata/dublin_core_elements#creator">

<META NAME="DC.Creator.Address" CONTENT="saanle@cc.jyu.fi">
<LINK REL=SCHEMA.dc HREF="http://purl.org/metadata/dublin_core_elements#creator">

<META NAME="DC.Subject" CONTENT="XML">
<LINK REL=SCHEMA.dc HREF="http://purl.org/metadata/dublin_core_elements#subject">

<META NAME="DC.Subject" CONTENT="XLL">
<LINK REL=SCHEMA.dc HREF="http://purl.org/metadata/dublin_core_elements#subject">

<META NAME="DC.Subject" CONTENT="XSL">
<LINK REL=SCHEMA.dc HREF="http://purl.org/metadata/dublin_core_elements#subject">

<META NAME="DC.Description" CONTENT="XML-standardiperheen tarkastelu arkistoinnin
kannalta">
<LINK REL=SCHEMA.dc
HREF="http://purl.org/metadata/dublin_core_elements#description">

<META NAME="DC.Publisher" CONTENT="EVA-projekti">
<LINK REL=SCHEMA.dc
HREF="http://purl.org/metadata/dublin_core_elements#publisher">

<META NAME="DC.Date" CONTENT="(SCHEME=ISO8601) 1998-01-31">
<LINK REL=SCHEMA.dc HREF="http://purl.org/metadata/dublin_core_elements#date">

<META NAME="DC.Type" CONTENT="Text">
<LINK REL=SCHEMA.dc HREF="http://purl.org/metadata/dublin_core_elements#type">

<META NAME="DC.Format" CONTENT="(SCHEME=IMT) text/html">
<LINK REL=SCHEMA.dc HREF="http://purl.org/metadata/dublin_core_elements#format">
<LINK REL=SCHEMA.imt HREF="http://sunsite.auc.dk/RFC/rfc/rfc2046.html">

<META NAME="DC.Identifier" CON-
TENT="http://www.jyu.fi/%7Esaanle/eva/kokodoc.html">
<LINK REL=SCHEMA.dc
HREF="http://purl.org/metadata/dublin_core_elements#identifier">

<META NAME="DC.Language" CONTENT="(SCHEME=ISO639-1) fi">
<LINK REL=SCHEMA.dc
HREF="http://purl.org/metadata/dublin_core_elements#language">

<META NAME="DC.Date.X-MetadataLastModified" CONTENT="(SCHEME=ISO8601) 1998-
05-18">
<LINK REL=SCHEMA.dc HREF="http://purl.org/metadata/dublin_core_elements#date">
```