

Tommi Koivisto

**RESURSSIEN IDENTIFIOINTI
DIGITAALISESSA OIKEUKSIEN HALLINNASSA**

Tietojärjestelmätieteen
pro gradu -tutkielma
18.12.2002

Jyväskylän yliopisto
Tietojenkäsittelytieteiden laitos
Jyväskylä

TIIVISTELMÄ

Koivisto, Tommi Markus

Resurssien identifiointi digitaalisessa oikeuksien hallinnassa / Tommi Koivisto

Jyväskylä: Jyväskylän yliopisto, 2002.

69 s.

Tutkielma

Tässä työssä tarkastellaan resurssien identifiointia digitaaliseen oikeuksien hallintaan kehitettävässä järjestelmässä. Tavoitteena on selvittää, miten resurssit tulisi identifioida tässä järjestelmässä ja mitkä asiat vaikuttavat käytettävän identifiointijärjestelmän valintaan. Tutkimustulosten perusteella pyritään myös luomaan kuva siitä, onko Open Digital Rights Language -kielen (ODRL) käyttämä identifiointimekanismi, Uniform Resource Identifier (URI), riittävä sovellusalueelle.

Aiheen käsittely alkaa digitaalisen oikeuksien hallinnan esittelystä tutkimusalueena. Tämän jälkeen käsitellään resurssien identifiointin ja niihin osoittamisen keskeisimpiä kysymyksiä. Lisäksi esitellään useita identifiointimekanismeja. Yhdeksi tutkimuksen tärkeimmistä aiheista nousee, miten resurssin sisältä voidaan rajata osia ja miten näille osille voidaan luoda oma osoitteensa. Tutkielma on käsitteellisteoreettinen ja sen lähdemateriaalina on käytetty aiheesta kirjoitettuja artikkeleita sekä teknisiä spesifikaatioita.

Tutkimuksen perusteella vaikuttaa siltä, että URI on riittävän monipuolinen identifiointimekanismi resursseille. Sen sijaan tiedot kehitettävän järjestelmän yksityiskohdista ja käytettävästä teknologiasta eivät riitä tunnisteen tarkan muodon määrittämiseen. Tutkimustulokset toimivatkin parhaiten aiheesta tehtävän jatkotutkimuksen pohjatietoina.

AVAINSANAT: digitaalinen oikeuksien hallinta, tunnisteen, identifiointi

ABSTRACT

Koivisto, Tommi Markus

Resource Identification in Digital Rights Management / Tommi Koivisto

Jyväskylä: University of Jyväskylä, 2002.

69 p.

Master's thesis

This thesis looks at resource identification in digital rights management. The objective of the study is to examine how resources should be identified in a digital rights management system and which factors have an effect on deciding which identification scheme is appropriate for the application area. Further discussion concentrates on Uniform Resource Identifier (URI), which is the identification mechanism used with Open Digital Rights Language (ODRL).

The study starts with an introduction to digital rights management as a field of research. Further discussion covers some of the most essential issues in resource identification and creation of resource locators. In addition, the study presents several identification mechanisms. One of the most important issues in this study is how to identify fragments of certain resources and how to create addresses for these fragments. The study is conceptual and the sources of information include articles on the subject, as well as technical specifications.

The conclusion of this study is that although URI is versatile enough for identifying resources in a digital rights management system, the information provided about the details of a rights management system is not sufficient for defining an exact format for identifiers. The results serve best as basics for further studies in the same domain of study.

KEYWORDS: digital rights management, identifiers, identification

SISÄLLYS

1	JOHDANTO	6
1.1	Tutkimuksen lähtökohdat.....	6
1.2	Keskeiset käsitteet.....	8
1.3	Mitä on digitaalinen oikeuksien hallinta?	10
1.4	Miksi tätä kaikkea pitää tutkia?.....	12
2	RESURSSIEN IDENTIFIJOINTI.....	13
2.1	Toiminnallinen granulariteetti.....	13
2.1.1	Abstraktiosta teoskappaleeseen.....	14
2.1.2	Kokoomateoksesta komponentteihin	17
2.2	Pysyvät tunnisteet	18
2.3	Tunnistejärjestelmien arviointi.....	19
3	IDENTIFIJOINTIJÄRJESTELMÄT	21
3.1	Kansainväliset standardinumerot	21
3.1.1	International Standard Book Number (ISBN).....	22
3.1.2	International Standard Serial Number (ISSN)	24
3.1.3	International Standard Text Code (ISTC).....	25
3.1.4	International Standard Technical Report Number (ISRN).....	25
3.2	Serial Item and Contribution Identifier (SICI).....	27
3.3	Formal Public Identifier (FPI).....	29
3.4	Digital Object Identifier (DOI)	30
3.5	Universal Unique Identifier (UUID).....	32
3.6	Kansallisbibliografian ID-tunnus (NBN).....	33
3.7	Uniform Resource Name (URN)	34
3.8	Järjestelmien sisäisistä tunnisteista	35
3.9	Identifiointijärjestelmän valinta	38
4	RESURSSIEN PAIKANTAMINEN	39
4.1	Uniform Resource Locator (URL).....	39
4.2	HTML: Katkelmatunnisteet ja URI	40
4.3	Extensible Markup Language ja resurssien paikantaminen.....	42
4.3.1	XML Path Language (XPath)	44
4.3.2	XML Pointer Language (XPointer)	47
4.4	Extensible Hypertext Markup Language (XHTML).....	49
4.4.1	XHTML:n syntaksi	50
4.4.2	Katkelmatunnisteet XHTML:ssä	50
4.5	Havainnot	51

5	TUNNISTEEN RESOLUUTIO.....	52
5.1	Resoluutiopalvelut	52
5.2	ODRL digitaalisessa oikeuksien hallinnassa	54
5.2.1	ODRL:n oikeuksien hallinnan malli	54
5.2.2	ODRL:n kontekstimalli.....	55
5.3	Resurssien identifiointi ODRL:ssä.....	57
5.4	Passiivisista resursseista aktiivisiin.....	57
6	YHTEENVETO	59
	LÄHTEET	62

1 JOHDANTO

Huomattava osa kaikesta tiedosta tuotetaan nykyisin digitaalisessa muodossa tai muunnetaan jossain vaiheessa digitaalseksi. Digitaalisuus helpottaa teksti-, ääni- ja kuvaresurssien muokkaamista, jakelua ja säilytystä, mutta samalla se tekee niihin liittyvien oikeuksien hallinnasta haasteellista. Digitaalisesta materiaalista voi useissa tapauksissa ottaa erittäin helposti alkuperäisen veroisia kopioita, joita voi jakaa lähes rajattomasti tietoverkkojen välityksellä. On siis syntynyt selvä tarve kehittää standardeja ja järjestelmiä digitaaliseen käyttö- ja tekijänoikeuksien hallintaan.

1.1 Tutkimuksen lähtökohdat

Yksi digitaalisen oikeuksien hallinnan keskeisistä kysymyksistä on, miten resurssit identifioidaan eli miten voidaan yksiselitteisesti kertoa, mihin resurssiin viitataan. Viittaamisella voidaan tarkoittaa skeä viittaamista kirjallisessa materiaalissa että jossain tietojärjestelmässä. Perinteiset julkaisut kuten kirjat ja kirjan kaltaiset julkaisut on esimerkiksi voitu identifioida *ISBN-numerolla* (International Standard Book Number) (ISO 2108:1992), mutta sen käyttö digitaalisen aineiston identifioinnissa ei välttämättä ole paras ratkaisu. Internetistä saatavilla oleva aineisto taas olisi helppo identifioida käyttämällä tunnisteena resurssin sisältämän tiedoston *Uniform Resource Locator* -osoitetta (URL). URL ei kuitenkaan identifioi resurssia itseään vaan ainoastaan sijainnin, jossa resurssi on mahdollisesti jonain ajankohtana ollut saatavilla. Jos resurssi siirretään toiseen paikkaan, sen URL-osoite muuttuu.

Tämän tutkimuksen lähtökohtana on Jyväskylän yliopiston projekti, jossa kehitetään järjestelmää digitaalisen oppimateriaalin ja sen oikeuksien hallintaan. Sopimukset oppimateriaaliin liittyvistä käyttö- ja tekijänoikeuksista on perinteisesti tehty paperimuodossa ja oikeuksia on hallittu järjestelmillä, jotka eivät ole yhteydessä hallittavaan materiaaliin. Tämä on ongelmallista erityisesti kahdessa suhteessa. Ensinnäkin digitaalisessa muodossa olevaa materiaalia muokataan useammin kuin painettua, joten oppimateriaalin tuottajan ja oppimisympäristön ylläpitäjän välisiä sopimuksia joudutaan päivittämään usein. Suurin ongelma on kuitenkin materiaalin käytön valvonta ja erityisesti väärinkäytösten estäminen.

Oikeudet on päätetty kuvata toteutettavassa järjestelmässä *Open Digital Rights Language* -kielellä (ODRL) (Iannella 2002), mikä myös vaikuttaa tiettyihin rajauksiin ja valintoihin, joita olen tutkimuksessani tehnyt. Tutkielmani ensisijainen tutkimusongelma on, miten resurssit tulisi identifioida tässä järjestelmässä ja riittävätkö ODRL:n omat identifiointimekanismit sovellusalueelle. Tutkimuksen edetessä laajensin aiheen käsittämään myös resurssien paikantamisen havaittuani, että se liittyy olennaisesti identifiointiin. Tutkimukseni on käsiteteoreettinen ja se perustuu kirjallisuusanalyysiin. Lähdemateriaalina tutkimuksessani ovat olleet aiheesta kirjoitetut artikkelit sekä tekniset spesifikaatiot.

Tutkimuksen johdannossa käydään läpi aihealueeseen liittyvät keskeiset käsitteet. Joitain keskeisiä käsitteitä käsitellään seuraavissa luvuissa tarkemmin. Johdannossa myös käsitellään digitaalista oikeuksien hallintaa tutkimusalueena. Tässä yhteydessä kerrotaan muun muassa, että resurssien identifiointi on yksi digitaalisen oikeuksien hallinnan keskeisistä kysymyksistä.

Luku kaksi käsittelee resurssien identifioinnin lähtökohtia. Luvun alkuosassa pohditaan, milloin resurssille tulisi antaa omat tunnisteensa. Luvussa esitellään käsite *toiminnallinen granulariteetti*, joka tarkoittaa tarkkuutta, jota käytetään resurssien erottelussa. Luvun loppuosa käsittelee periaatteita, joiden perusteella eri identifiointijärjestelmiä voidaan arvioida ja jotka vaikuttavat käytettävän identifiointijärjestelmän valintaan. Tärkeimmät havainnot ovat, että tunnisteiden tulee olla pysyvä ja olisi hyvä välttää älykkäitä tunnisteita eli tunnisteita, joihin on koodattu merkityksellistä informaatiota.

Kolmannessa luvussa käsitellään muutamia käytössä olevia identifiointijärjestelmiä, joilla voidaan antaa tunniste digitaaliselle – ja joillakin myös fyysiselle – materiaalille.

Neljäs luku keskittyy resurssien paikantamiseen. Aiemmin todettiin, että tunnisteet eivät saisi olla älykkäitä. Niiden asemesta olisi hyvä kehittää jonkinlainen resoluutiopalvelu, jossa tunnisteiden perusteella resurssin ja sen metatietojen todellinen sijainti selvitetään erillisestä tietokannasta tai vastaavasta lähteestä. Paikantamisessa yleiskäyttöisin tekniikka on URL, joka on *URI:n (Uniform Resource Identifier)* osajoukko. Siihen

perustuu resurssien paikantaminen muun muassa World Wide Webissä. Resurssien paikantamisessa vaikein osa-alue on osoittaminen sellaisiin resursseihin, jotka ovat osa jotain suurempaa resurssia. Tämä tarkoittaa sitä, että resurssi käsittää esimerkiksi vain osan jotain tiedostoa. Tällaisten resurssien rajaamiseen esitetään URI:n katkelmatunnisteeseen perustuvia tai XML-dokumenttien käsittelyyn käytettäviä tekniikoita. Lisäksi kerrotaan dokumenttien hallintajärjestelmien käyttämistä oliotunnisteista.

Viidennessä luvussa käsitellään resurssien sijainnin ja tunnisteiden välistä yhteyttä oikeuksia hallitsevan järjestelmän kannalta. Ensimmäisenä esitellään uusi termi, *resoluutio*, eli resurssin sijainnin selvittäminen tunnisteiden perusteella. Tämän jälkeen siirrytään resurssien identifiointiin ODRL:ssä. Lopuksi käsitellään lyhyesti resursseja, jotka osallistuvat aktiivisesti käyttäjälle näytettävän sisällön muotoiluun.

Digitaalinen oikeuksien hallinta on tutkimuksessa mukana vain yhtenä lähtökohdista, eikä tutkimuksen aihetta käsitellä esimerkiksi tekijänoikeudellisesta näkökulmasta. Fyysisten resurssien samoin kuin audiovisuaalisen materiaalin ja äänitteiden hallinta rajataan tutkimusalueen ulkopuolelle, sillä se ei ole olennaista tutkimuksen jälkeen tehtävän jatkotutkimuksen kannalta.

1.2 Keskeiset käsitteet

Digitaalinen oikeuksien hallinta (Digital Rights Management) käsittää organisaation aineelliseen ja aineettomaan tietopääomaan kohdistuvien oikeuksien kuvauksen, kerrostamisen, analysoinnin, arvioinnin, kaupankäynnin sekä valvonnan (Iannella 2002, 1). Iannella (2001) korostaa, että kysymys on nimenomaan *digitaalisesta* oikeuksien hallinnasta eikä vain *digitaalisten* oikeuksien hallinnasta. Digitaalisen oikeuksien hallinnan avulla voidaan siis hallita sekä fyysisessä että digitaalisessa muodossa olevaa materiaalia.

Resurssi on tässä tutkielmassa yleisnimi kaikelle sille materiaalille, joka on syntynyt inhimillisen älyn tai mielikuvituksen tuloksena ja johon liittyy käyttö- tai tekijänoikeuksia. Valitsemani termi on vain yksi useista, joita lähdemateriaalissani

käytetään. Toinen usein käytetty termi on *aineeton omaisuus (intellectual property)* (esim. Iannella 2001), joka ei kuitenkaan suomennettuna ole tarpeeksi joustava. Resurssi on myös merkitykseltään laajempi ja sen voidaan käsittää tarkoittavan aineettoman omaisuuden lisäksi myös fyysisen materiaalin. Resurssi on myös Internet-standardeissa (esim. Sollins & Masinter 1994) yleisesti käytetty yleisnimi kaikelle elektroniselle materiaalille.

Entiteetti on jokin eristettävissä oleva ja jonkin järjestelmän tai mallin kannalta mielekäs itsenäinen kokonaisuus. Entiteetti voi siis asiayhteydestä riippuen tarkoittaa esimerkiksi jotain resurssia, sen käyttäjää tai näiden kahden välillä olevia oikeuksia.

Identifiointi tarkoittaa tässä tutkimuksessa ensisijaisesti resurssien tunnistamista eli sitä, miten jokin resurssi voidaan erottaa yksiselitteisesti muista. Se tarkoittaa sekä sitä prosessia, jolla jollekin resurssille annetaan tunniste, että tähän resurssiin viittaamista tunnisteen avulla. Kirja tai siihen rinnastettava julkaisu voidaan esimerkiksi identifioida ISBN-numeron (ISO 2108:1992) avulla. Identifioinnista voitaisiin käyttää myös suomalaisperäistä sanaa *tunnistaminen*. Kyseistä sanaa voidaan kuitenkin käyttää myös muissa merkityksissä, joten vierasperäisen ilmaisun käyttäminen vaikuttaa selkeämmältä. Toisaalta identifiointia voidaan kutsua myös *nimeämiseksi* (esim. Sollins & Masinter 1994), mutta termi on mielestäni merkitykseltään suppeampi kuin valitsemani. Lisäksi on selkeämpää, että tunnistetta ei rinnasteta nimeen, sillä nimi on usein attribuutti, jolta ei vaadita pysyvyyttä.

Tunniste on jokin merkkijono tai numerosarja, joka yksilöi resurssin vähintään yhden järjestelmän sisällä. Sitä voidaan kutsua myös *resurssin nimeksi*. Tunnisteet voivat olla maailmanlaajuisesti yksiselitteisiä, mikä tarkoittaa, että kahdella toisistaan poikkeavalla resurssilla ei voi olla samaa tunnistetta.

Resoluutiolla tarkoitetaan tässä yhteydessä prosessia, jossa resurssin tunnisteen perusteella käyttäjälle palautetaan joko itse resurssi tai sen metatietoja. Resurssin ja sen metatietojen sijainti selvitetään jonkinlaisen resoluutiopalvelun avulla. Palvelu voi toimia esimerkiksi sovelluksena käyttäjän koneella, se voi olla sisäänrakennettu resursseja hallitsevaan järjestelmään tai sijaita omalla palvelimellaan.

Tutkimuksen kannalta vähemmän olennainen, mutta usein mainittu *Internet-mediatyyppi* eli *MIME-tyyppi* (*Multipurpose Internet Mail Extensions*) (Freed & Borenstein 1996a ja 1996b) on tietyssä muodossa kerrottu tietokokonaisuuden kuvaus, josta selviää tiedon päätyyppi ja alityyppi. Mediatyyppien avulla ohjelmat kertovat toisilleen, millaista tietoa ne lähettävät, jotta vastaanottava sovellus osaa esittää tiedon oikealla tavalla.

Tutkimusraportissa viitataan joihinkin identifiointi- ja osoitusjärjestelmiin ennen niiden varsinaista käsittelyä luvussa kolme. Näitä ovat:

- *ISBN (International Standard Book Number)* (ISO 2108:1992), joka on laajasti käytössä oleva numerointijärjestelmä kirjoille.
- *ISSN (International Standard Serial Number)* (ISO 3297:1998), joka on ISBN-numeron vastine sarjajulkaisuille.
- *URI (Uniform Resource Identifier)* (esim. Berners-Lee et al. 1998), joka koostuu kahdesta osasta
 - o *URL (Uniform Resource Locator)* (esim. Berners-Lee et al. 1994), jota käytetään World Wide Webissä jonkin resurssin paikantamiseen.
 - o *URN (Uniform Resource Name)* (esim. Sollins & Masinter 1994), jonka on tarkoitus olla maailmanlaajuisesti yksilöivä ja pysyvä tunniste. URN:ää käytetään sekä resurssin identifiointiin että resoluutioon (Sollins & Masinter 1994, 2).
- *DOI (Digital Object Identifier)* (Paskin 2001), joka on URN:ää vastaava identifiointijärjestelmä. Se tarjoaa identifioinnin lisäksi myös reititysjärjestelmän, ja on näistä identifiointijärjestelmistä monimutkaisin.

1.3 Mitä on digitaalinen oikeuksien hallinta?

Tässä kohdassa käsitellään digitaalisen oikeuksien hallinnan tutkimuksen pääpiirteitä. Digitaalista oikeuksien hallintaa on tutkinut muun muassa Renato Iannella (esim. Iannella 2001 ja 2002). Iannella (2001) kertoo, että aiemmin digitaalinen oikeuksien hallinta oli pääsääntöisesti sitä, että dokumentit lukittiin ja niissä käytettiin salauksia. Nykyinen tutkimus tarkastelee aihetta kuitenkin monipuolisemmin ja ottaa huomioon sekä digitaalisen että fyysisen materiaalin ja myös oikeuksien haltijoiden väliset suhteet.

Iannella esittelee artikkelissaan (2001) kaksi arkkitehtuuria, jotka ovat kriittisiä digitaaliseen oikeuksien hallintaan suunniteltavissa järjestelmissä. Toinen on *toiminnallinen arkkitehtuuri (functional architecture)* ja toinen *tietoarkkitehtuuri (information architecture)*.

Toiminnallisessa arkkitehtuurissa digitaaliseen oikeuksien hallintaan suunniteltujen järjestelmien kehitys mallinnetaan kolmella alueella. Ensimmäinen näistä on resurssien luonti ja kokoaminen (Intellectual Property Asset Creation and Capture), jossa tutkitaan, miten voidaan hallita sisällön luomista siten, että sillä voidaan käydään helposti kauppaa. Toinen alue on näiden resurssien hallinta (Intellectual Property Asset Management) eli se, miten mahdollistetaan kaupankäynti tiedoilla. Kolmantena tutkitaan resurssien käytön hallintaa (Intellectual Property Asset Usage) sen jälkeen, kun omaisuudella on käyty kauppaa. (Iannella 2001)

Tietoarkkitehtuurin tasolla käsitellään sitä, miten järjestelmien käsittelemiä entiteettejä ja niiden välisiä suhteita mallinnetaan oikeuksien hallinnan kehyksessä. Myös tämä alue voidaan jakaa kolmeen osaan, jotka ovat *entiteettien mallintaminen*, niiden *identifiointi ja kuvaaminen* sekä *oikeusmäärittelysten muodostaminen*. (Iannella 2001)

Entiteettejä ja niiden välisiä suhteita on mallinnettu muun muassa indecs-projektissa (Rust & Bide 2000). Projektissa tunnistettiin kolme perusentiteettiä: *käyttäjät*, *oikeudet* ja *sisältö*. International Federation of Library Associations (IFLA) on puolestaan luonut mallin, jossa identifioitavat resurssit jaetaan neljälle abstraktiotasolle. Korkeimmalla tasolla on *teos (work)*, joka on vain ajatus jostain olemassa olevasta henkisen työn tuotteesta. Teoksesta on erilaisia *ilmauksia (expression)*, joista puolestaan on erilaisia *ilmentymiä (manifestation)*. Alimmalla tasolla IFLA:n mallissa on *teoskappale (item)*, joka tarkoittaa tietystä ilmentymästä otettua kopioita. (Plassard 1998, 12—13 ja 16—23) IFLA:n malli kuvataan tarkemmin seuraavassa luvussa.

Kaikki resurssit on sekä identifioitava että kuvattava, ja molempiin tehtäviin tulisi käyttää avoimia ja standardeja mekanismeja (Iannella 2001). Oikeuksien kuvaamiseen

kehitettyissä kielissä identifiointimekanismit voivat kuitenkin olla erilaisia. Sisällön kuvaus voidaan puolestaan tehdä jollain metadatastandardilla. Niitä ovat kuvanneet muun muassa Päivärinta et al. (2002).

Viimeinen osa tietoarkkitehtuurin mallintamista on oikeuksien kuvaaminen jonkin kuvauskielen syntaksin mukaisilla lauseilla. Nämä lauseet voivat olla hyvinkin monimutkaisia, ja siksi niihin on kehitetty myös tapoja kuvata eri lauseiden välisiä yhteyksiä. Eräs mahdollinen kuvauskieli on Open Digital Rights Language (ODRL) (Iannella 2002).

1.4 Miksi tätä kaikkea pitää tutkia?

Digitaaliseen oikeuksien hallintaan liittyvä tutkimus on osa muuta metatietojen kuvauksen tutkimista. Indecs-projektissa (Rust & Bide 2000) kehitettiin malli metadatan kuvaamista varten. Projektin alkuoletukset kuvaavat mielestäni hyvin myös tämän tutkimuksen lähtökohtia:

- 1) Metadata on kriittistä. Sähköinen kaupankäynti nojaa perinteistä kaupankäyntiä enemmän siihen, miten asiat (ihmiset, tavarat ja sopimukset) identifioidaan ja kuvataan. Jos jossain vaiheessa näiden kuvauksien välisissä suhteissa on aukkoja, ei välttämättä saavuteta haluttuja lopputuloksia.
- 2) Tavara (*stuff*) on monimutkaista. Esimerkiksi yksi audiovisuaalinen teos voi koostua sadoista tai tuhansista yksittäisistä resursseista. Lisäksi jonkin resurssin erilaisiin ilmentymiin voi kohdistua erilaisia oikeuksia.
- 3) Metadata on modulaarista. Koska tavara on monimutkaista, kaupankäynnin kohdetta kuvaava metadata koostuu useiden henkilöiden luomista palasista.
- 4) Transaktiot tarvitsevat automaatiota. Järjestelmistä tulee niin monimutkaisia, että niiden hallinnointi käy erittäin vaikeaksi ilman osittaista tai täydellistä automaatiota. (Rust & Bide 2000, 4–5)

Kaiken lähtökohta on kuitenkin se, että resurssit voidaan erottaa toisistaan. On oltava jotkin periaatteet siitä, milloin ja miten resurssit identifioidaan. Näitä periaatteita kuvataan seuraavassa luvussa.

2 RESURSSIEN IDENTIFIOINTI

Erilaisia resursseja voidaan tarkastella lukemattomilla eri tavoilla, ja eri tahojen vaatimukset identifioinnille eroavat toisistaan huomattavasti. Toisille riittää se, että voidaan erottaa jonkin resurssin abstraktio eli taustalla oleva ajatus jostain henkisen työn tuotteesta. Toiset taas voivat vaatia yksilöivää tunnistetta, jonkinlaista sarjanumeroa, jokaiselle kopiolle, joka jostain resurssista on tehty. Resurssin osat voivat myös vaatia identifiointia millä tahansa tasolla ja niitä voidaan pitää myös omina resursseinaan. Voidaan esimerkiksi haluta erottaa jokin luku kokonaisuudesta tai viitata tiettyyn kuvaan tekstin keskellä. On myös pohdittava, pitäisikö jonkin resurssin eri kehitysvaiheissa tuotetut versiot erottaa toisistaan.

2.1 Toiminnallinen granulariteetti

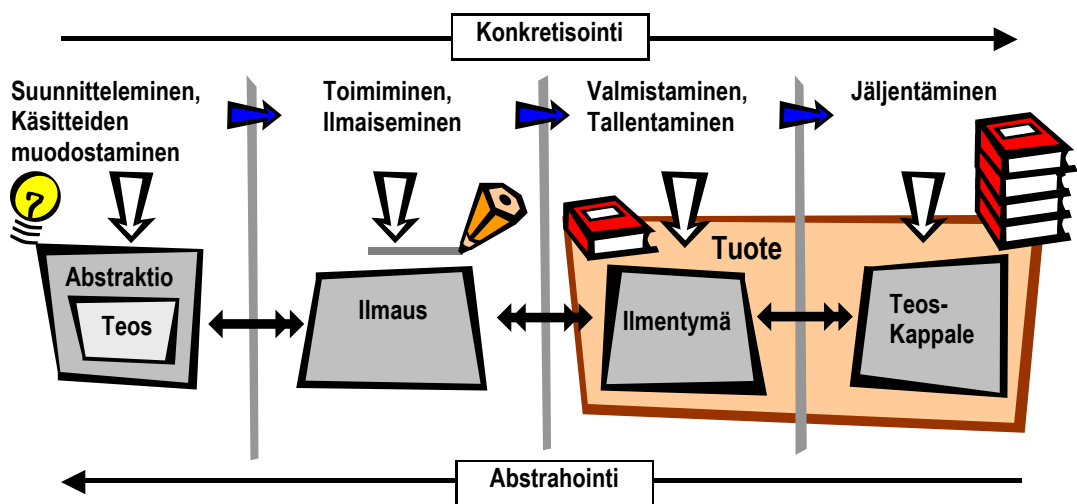
Kaikkea ei kannata identifioida, vaikka voitaisiinkin. Paskin (1997) on arvioinut identifioitavien entiteettien määrää maailmassa, jos resursseista eroteltaisiin myös niiden komponentit sekä eri versiot. Arviossa huomioitiin vain tieteelliset, tekniset ja lääketieteelliset dokumentit. Identifioitavien resurssien ja niiden komponenttien yhteismäärä nousisi silti noin sataan biljoonaan.

Termi, jota käytetään tämänkaltaisesta päätöksien teosta, on *toiminnallinen granulariteetti*. Se tarkoittaa sitä, kuinka tarkasti resursseja ja niiden osia ja versioita yksilöidään. Sen periaate on, että jokin entiteetti identifioidaan, silloin *kun siihen on tarvetta* (Rust & Bide 2000, 10). Entiteetti voi olla minkä tahansa laajuinen kokonaisuus, joka halutaan yksilöidä jossain yhteydessä. Tarkkuus, jolla resurssit yksilöidään riippuu valtavasta määrästä tekijöitä, joihin kuuluu Rust & Biden (2000, 10) mukaan ainakin resurssin tyyppi, sen sijainti ajassa ja paikassa, sen tarkka kokoonpano ja asema, käyttötarkoitukset, tiedon pysyvyys, luomisprosessi sekä sen osapuolen entiteetti, joka identifioi resurssin. Eri organisaatiot voivatkin erotella resursseja toisistaan hyvin erilaisilla periaatteilla. Esimerkiksi julkaisija voi valita, että se erottelee toisistaan kirjan painokset, joissa on joitain pieniä kirjoitus- ja painovirheitä, mutta kirjasto ei välttämättä halua tehdä samanlaista erottelua. Toisaalta kirjaston lainausjärjestelmä vaatii, että jokaiselle niteelle luodaan oma tunnisteensa.

2.1.1 Abstraktiosta teoskappaleeseen

International Federation of Library Associations (IFLA) on kehittänyt mallin bibliografisten tallenteiden luomista varten (Plassard 1998). Mallissa resurssit ja niiden tuottajat kuvataan entiteetteinä ja näiden entiteettien välisiä suhteita ja attribuutteja tarkastellaan monipuolisesti. Oman osansa entiteettien välisten suhteiden kuvaamiseen on tuonut Indecs-projekti (Rust & Bide 2000), joka on luonut kehyksen resurssien metatietojen kuvausta varten. Molemmissa kehyksissä kuvataan myös resurssien suhteita muihin entiteetteihin, kuten käyttäjiin ja sisällöntuottajiin, mutta tässä yhteydessä tarkastelemme vain resurssien eri abstraktiotasojen välisiä suhteita. Nämä suhteet esitetään kuviossa 1. Kuvio pohjautuu IFLA:n ja Indecs-projektin entiteettien välisiä suhteita mallintaviin kuvioihin (Plassard 1998, 13 ja Rust & Bide 2000, 27).

Kuvioon on piirretty neljä entiteettiä, jotka on järjestetty niiden abstraktiotason mukaisesti. Entiteettien yläpuolella esitetään, minkälaisen toiminnan tuloksena entiteetti on syntynyt. Entiteettien välissä olevat yksin- ja kaksinkertaiset nuolet puolestaan kuvaavat, kuinka moneen seuraavan tason entiteettiin yksi entiteetti voi olla yhteydessä.



KUVIO 1. Resurssien abstraktiotasot.

Ylimmällä tasolla olevaa entiteettiä kutsutaan Indecs-kehyksessä *abstraktioksi* (*abstraction*) (Rust & Bide 2000, 28). Abstraktioita ovat teokset, esitykset, äänitteet, radiolähetykset sekä kriittiset tai tieteelliset julkaisut (Rust & Bide 2000, 30). IFLA taas käyttää abstrakteimmasta entiteetistä nimitystä *teos* (*work*), joka on jokin eristettävissä oleva taiteellinen tai älyllinen tuotos (Plassard 1998, 16). Teos on abstrakti entiteetti, eikä ole olemassa yhtään fyysistä esinettä, jota voidaan kutsua teokseksi. Toimintana teoksen – tai jonkin muun abstraktion – luominen on vain jonkinlaista ideointia ja suunnittelua, joka ei ole vielä konkretisoitunut missään muodossa. IFLA määrittelee, että teos tunnustetaan sen ilmauksien kautta, mutta itse teos on vain se, mitä sen saamien olomuotojen välillä on yhteistä (Plassard 1998, 16). Kun puhumme esimerkiksi Aleksis Kiven kirjoittamasta Seitsemästä veljeksestä teoksena, emme tarkoita tiettyä fyysistä kirjaa kirjahyllyssämme, vaan sitä älyllistä työtä, joka on kaikkien teoksen ilmauksien takana.

IFLA:n käsityksen mukaan tietyn teoksen parannellut versiot eivät ole omia teoksiaan, vaan saman teoksen eri ilmauksia. Jos sen sijaan uuden version tuottaminen vaatii merkittävästi itsenäistä taiteellista ponnistusta, sitä pidetään uutena teoksena. (Plassard 1998, 16–17) Esimerkiksi Armas Launiin Seitsemästä veljeksestä kirjoittamaa ja säveltämää 4-näytöksistä koomillista oopperaa tai Jouko Turkan sovittamaa TV-sarjaa voidaan pitää omina teoksinaan.

Toinen IFLA:n ja Indecs-projektin tunnistama entiteetti on *ilmaus* (*expression*) (Plassard 1998, 18 ja Rust & Bide 2000, 27). Se on teoksen – eli Indecs-kehysten mukaisesti abstraktion – älyperäinen tai taiteellinen toteutuma esimerkiksi alfanumeerisena, musikaalisena tai koreografisena notaationa, äänenä, kuvana, esineenä tai edellä mainittujen muotojen yhdistelminä (Plassard 1998, 18). Kuviossa 1 abstraktion ja ilmauksen väliin piirretty nuoli osoittaa, että yhdestä abstraktiosta voi olla useita ilmauksia, mutta ilmaus on vain yhden abstraktion toteutuma (Plassard 1998, 13).

Ilmauksena pidetään esimerkiksi sitä, minkälaisia sanoja tai lauseita jonkin teoksen toteutuksessa on käytetty. Se ei ota huomioon fyysisiä asioita kuten kirjan sivujen taittoa tai käytettyjä kirjasinleikkauksia. (Plassard 1998, 18) Jos ilmauksen älyperäiseen tai taiteelliseen sisältöön tehdään muutoksia, pidetään tuloksena syntyvää entiteettiä

uutena ilmauksena (Plassard 1998, 19). Seitsemän veljeksien eri ilmauksia ovat esimerkiksi Aleksis Kiven alkuperäinen teksti vuodelta 1870 ja Alex Matsonin käännös *Seven Brothers* vuodelta 1952.

Kolmas abstraktiotaso on *ilmentymä (manifestation)*, joka on tietyn ilmauksen konkreettinen muoto (Plassard 1998, 20). Se syntyy jonkinlaisen tallentamisen tai valmistamisen tuloksena. Entiteettinä ilmentymä edustaa kaikkia fyysisiä olioita, joilla on samat ominaispiirteet niin sisällön kuin fyysisen muotonsa suhteen (Plassard 1998, 20). Ilmaus voi ruumiillistua ilmentymäksi eri tallennusvälineille kuten paperille, äänikasetille tai vaikkapa tiedostona CD-levylle. Joissain tapauksissa teoksen ilmauksesta on olemassa vain yksi kopio, esimerkiksi kirjailijan käsikirjoitus tai alkuperäinen maalaus. Toisissa tapauksissa ilmauksesta on tehty useita kopioita julkista levitystä varten. Seitsemästä veljeksestä on esimerkiksi otettu viimeisen reilun 130 vuoden aikana useita painoksia, joissa sisältö on sama, mutta ulkoasu on vaihdellut. Jokaista näistä painoksista pidetään omana ilmentymänään. Ilmauksen ja ilmentymän välillä vallitsee monesta moneen -suhde: Ilmauksesta voi olla monta ilmentymää, mutta toisaalta yksi ilmentymä voi sisältää useita ilmauksia (Plassard 1998, 13).

Teoskappale (item) on yhden ilmentymän konkreettinen malliesimerkki (Plassard 1998, 23). Monissa tapauksissa se on yksi fyysinen olio, kuten kirja tai levy, mutta joissain tapauksissa teoskappale koostuu useammasta osasta, kuten kirja, joka on jaettu useaksi niteeksi tai äänite, joka on jaettu kahdelle CD-levylle. Kun puhutaan sisällöstä ja muodosta, teoskappale on identtinen ilmentymän kanssa. Kuitenkin eri teoskappaleiden välillä voi olla eroja, jos esimerkiksi teoskappale on vahingoittunut sen jälkeen kun se on tuotettu. Esimerkiksi Seitsemästä veljeksestä on omassa kirjahyllyssäni kopio, jonka etukannen sisäpuolelle on liimattu *Ex Libris*. Kirjaston lainakappaleen kannet on puolestaan suojattu kontaktimuovilla. Kuvioon 1 ilmentymän ja teoskappaleen väliin piirretty nuoli osoittaa, että ilmentymä voi olla kopioitu useiksi teoskappaleiksi, mutta teoskappale on aina yhden ja vain yhden ilmentymän kopio. Indeks-kehys kutsuu lisäksi ilmentymiä ja teoskappaleita yhteisellä nimellä *tuote (artefact)* (Rust & Bide 2000, 27).

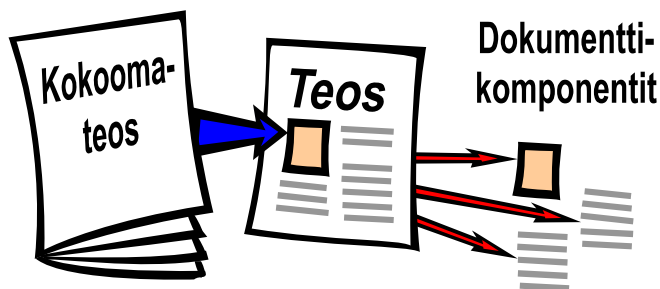
2.1.2 Kokoomateoksesta komponentteihin

IFLA:n kehyksen mukaan teos voi koostua myös toisista teoksista (Plassard 1998, 28). Teos voi siis edustaa jonkin suuremman teoksen osaa kuten jossain lehdessä olevaa yksittäistä artikkelia. Nämä *kokoomateokset* käyttäytyvät mallissa samoin kuin muut teokset, ja myös ne saavat erilaisia ilmauksia ja ilmentymiä.

Kuviossa 2 havainnollistetaan kolmea tasoa, joilla Paskinin (1997) mukaan jollekin resurssille voidaan antaa tunnisteita. Ylimmällä tasolla on kokoomateos, tai kuten Paskin (1997) sitä kutsuu, *dokumenttiryhmä (document cluster)*, jota edustaa jokin fyysinen ilmaus, tietty aikakausjulkaisu tai jokin muu kokoelma erillisiä resursseja. Sille voidaan antaa oma tunnisteensa sarjajulkaisujen ISSN-numeron tapaan. Tällaisina kokoomateoksina voidaan pitää jopa tietokantoja (Paskin 1997).

Kokoomateoksen sisällä on toinen taso, yksittäinen teos. Tästä entiteettiä Paskin (1997) kutsuu yksinkertaisesti dokumentiksi. Se voi olla esimerkiksi yksittäinen luku kirjassa tai artikkeli aikakauslehdessä.

Alimmalla tasolla ovat komponentit, joista identifioitavan resurssin ilmentymä koostuu: kuvioita, taulukoita, lukuja, kappaleita ja niin edelleen. Näistä Paskin (1997) käyttää nimitystä *dokumenttikomponentit*. Paskin (1997) epäilee, että näiden identifiointi erikseen on sekä mahdollista että käytännöllistä erityisesti kun siirrytään elektroniseen julkaisu-ympäristöön.



KUVIO 2. Kokoomateoksesta komponentteihin. Resurssi koostuu toisista resursseista.

2.2 Pysyvät tunnisteet

Resurssit voidaan identifioida monin tavoin. WWW:ssä yleinen tapa elektronisten dokumenttien identifioinnissa on käyttää siihen *Uniform Resource Locatoria* eli *URL:ia*. Tämä on kuitenkin virheellistä, sillä URL ei oikeastaan identifioi resurssia, vaan ainoastaan jonkin sijainnin, jossa resurssi on tietynä ajankohtana mahdollisesti ollut saatavilla. Useissa tapauksissa URL ei ole pysyvä osoite. Kun laitteistojen kokoonpanoja muutellaan, tiedostojärjestelmiä järjestellään uudelleen ja organisaatorakenteita muutetaan, Internet-resurssien osoitteet muuttuvat. URL:ia voidaan siten verrata vaikkapa jonkin kirjan sijaintiin kirjaston hyllyssä. Kirjaa ei identifioidakaan hyllysijainnin perusteella, vaan kirjaston ATK-järjestelmä on todennäköisesti antanut kirjalle jonkinlaisen kirjaston sisäisessä käytössä olevan tunnistenumeron. Tämän numeron avulla voidaan seurata kirjan liikkeitä lainaajien ja kirjaston välillä.

Uusia identifiointijärjestelmiä esitellään aika ajoin. Jotkin on tarkoitettu vain painotuotteiden identifiointiin, kun taas toiset yrittävät vastata myös elektronisen aineiston haasteisiin. Vanhoja järjestelmiä on myös liitetty uusien järjestelmien osiksi.

Tunnisteille asetetut vaatimukset ovat eri lähteissä yhteneviä. Esimerkiksi Rust & Bide (2000, 9) ovat listanneet identifiointijärjestelmien tärkeimmiksi ominaisuuksiksi 1) yksilöllisyyden vähintään yhdessä toimintaympäristössä 2) tunnisteiden pysyvyyden, 3) turvallisuuden ja 4) jonkin tunnistettavaa resurssia kuvaavan perusmetadatan julkisen saatavuuden.

Pysyvien tunnisteiden ongelma on tunnistettu monilla tahoilla ja sen parissa on työskennelty laajasti. Muun muassa Internet Engineering Task Force (IETF), Internetin kehitykselle standardeja asettava organisaatio, tunnisti URL-osoitteiden rajoitteet useita vuosia sitten, ja perusti työryhmän työstämään järjestelmää yksilöllisille tunnisteille. Lopputuloksena oli *Uniform Resource Name (URN)* (esim. Sollins & Masinter 1994). Corporation for National Research Initiatives (CNRI) ja R. R. Bowker puolestaan kehittivät Digital Object Identifier (DOI) -järjestelmää. Teknisesti DOI perustuu CNRI:n kehittämään kahvajärjestelmään (Handle System) (esim. Sun & Lannom 2002),

joka taas on yhteensopiva URN-tunnusten kanssa. Tunnisteita käsitellään tarkemmin seuraavassa luvussa.

Tunnisteen pysyvyyteen vaikuttaa se, miten paljon merkityksellistä informaatiota siihen on koodattu. *Tyhmillä tunnisteella* tarkoitetaan puhtaasti satunnaista merkkijonoa, joka voidaan tulkita vain viittaamalla keskitettyyn tietokantaan. Tunnisteen itsensä tutkiminen ei kerro mitään resurssista, jonka se identifioi. *Älykkästä tunnisteesta* voi sen sijaan palauttaa jotain tarkoituksellista tietoa kysymättä sitä toiselta järjestelmältä tai resurssilta itseltään. (Green & Bide 1997) Esimerkki älykkästä tunnisteesta on ISBN: sen ensimmäinen osa identifioi maan, kielen tai maantieteellisen alueen, jossa kirja on julkaistu. Toinen osa taas identifioi julkaisijan. (ISO 2108:1992, 1)

Green & Bide (1997) huomauttavat, että ISBN:ää ja vastaavia älykkäitä tunnisteita on kritisoitu digitaalisessa ympäristössä. Toiset sidosryhmät, erityisesti kirjailijoiden ryhmä, näkevät tunnisteiden oleva julkaisijakeskeisiä, ja niiden käytössä unohdetaan taustalla olevat aineistoon liittyvät oikeudet. Tämä voikin olla epäoleellista maailmassa, jossa käsitellään fyysisiä tavaroita ja jossa on tärkeintä vain tunnistaa näiden tavaroiden lähde. Aineistoon liittyvät oikeudet ovat selvästi tärkeämpiä digitaalisessa ympäristössä, jossa fyysisen tuotteen julkaisijalla ei ole välttämättä elektronisia oikeuksia aineistoon tai jossa oikeudet, jotka ovat alun perin olleet julkaisijalla, on myös voitu siirtää jollekin toiselle. (Green & Bide 1997)

On helppo kuvitella, että älykkäitä tunnisteita pyrittäisiin välttämään. Jos jokin tunniste koostuu enemmän tai vähemmän merkityksellisestä tiedosta, sitä joudutaan aika ajoin päivittämään. Green & Bide (1997) väittävätkin, että väittely tyhmien ja älykkäiden tunnisteiden käytöstä ratkeaa tyhmien tunnisteiden hyväksi. Ainoa älykkyys on silloin mekanismeissa, jotka takaavat numeroiden ainutkertaisuuden.

2.3 Tunnistejärjestelmien arviointi

Digitaalisten resurssien identifiointijärjestelmän valinta on monimutkainen tehtävä, ja väärä valinta on helppo tehdä. Julkaisijat taipuvat Green & Biden (1997) mukaan esimerkiksi ajattelemaan, että kirja ja teos ovat sama asia, joten teos voidaan

identifioida ISBN-numerolla. ISBN-numero kuitenkin identifioi ainoastaan kyseisen teoksen yhden ilmentymän, sitoo sen tiukasti julkaisijaan ja jättää tekijänoikeudellisen näkökulman huomiotta. IFLA:n mallissa teos ja sen ilmaukset ovat kuitenkin abstrakteja entiteettejä, joilla ei ole tunnistetta attribuuteissaan. Mallissa tunniste liitetään ainoastaan ilmentymään ja teoskappaleisiin (Plassard 1998, 44 ja 49). Indecs-kehys liittyy kuitenkin kaikkiin abstraktiotasoihin jonkinlaisen tunnisteiden (Rust & Bide 2000, 28—29).

Digitaalisessa ympäristössä tarvittava granulariteetin taso voi olla mielivaltaisen, ja voidaan joutua identifioimaan pieniäkin osia jostain isommasta resurssista. Näiden pienten osien tai dokumenttikomponenttien identifiointiin ei ole kehitetty yksiselitteisesti toimivia ratkaisuja. Toisaalta esimerkiksi DOI:lla kerrotaan voitavan identifioida mikä tahansa pienikin resurssi, mutta pelkkä tunnisteiden antaminen ei riitä. Komponenttiin pitää myös pystyä osoittamaan. Tähän ongelmaan palataan tarkemmin luvussa neljä.

Lynch (1997) on esittänyt viisi tärkeää kysymystä, jotka on otettava huomioon kun arvioidaan tunnistejärjestelmiä.

1. Mikä on tunnistejärjestelmän laajuus ja kenellä on lupa antaa tunnisteita?
2. Mitkä ovat säännöt, joilla uusia tunnisteita annetaan ja milloin kaksi jonkin työn ilmentymää katsotaan erillisiksi?
3. Miten teoksen tunniste luodaan, ja voidaanko se johtaa työstä itsestään?
4. Kuinka tästä tunnisteesta voidaan selvittää tunnistettavan resurssin sijainti?
5. Kuinka pysyvä tunniste on ajan kuluessa?

Green & Bide (1997) ovat vakuuttuneita siitä, että kysymykseen identifiointijärjestelmistä ei ole yhtä yksittäistä vastausta, ja sen löytyminen on yhtä todennäköistä kuin onnistuminen yksisarvisen etsinnässä. He myös palauttavat mieliin sen, että tällä hetkellä suuri osa siitä materiaalista, josta julkaisijoiden asiakkaat ovat kiinnostuneita, ei ole elektronisessa muodossa. Jos taas jostain resurssista on olemassa fyysinen ilmentymä, myös sen elektroniset ilmentymät identifioidaan todennäköisesti samalla tunnisteella.

3 IDENTIFIOINTIJÄRJESTELMÄT

Tässä luvussa käsitellään joitain kansainvälisesti vakiintuneita identifiointijärjestelmiä. Lista järjestelmistä ei ole tyhjentävä, vaan luvun tarkoitus on luoda kuvaa siitä, miten erilaisin perustein resursseja voidaan identifioida, sekä siitä, miten erilaisia muotoja tunnisteet voivat saada. Erilaisia tunnisteita myös käytetään eri yhteyksissä, ja pyrin tämän luvun tietojen perusteella muodostamaan käsityksen siitä, onko jokin tunnistejärjestelmä muita käyttökelpoisempi oppimisympäristön resurssien identifioinnissa.

3.1 Kansainväliset standardinumerot

Tässä kohdassa käsitellään kansainvälisiä standardinumeroita. Niillä tarkoitan *ISO:n* (*International Organization for Standardization*) standardoimia identifiointijärjestelmiä. Tunnisteista käytetään yleisesti nimitystä numero, vaikka tunnisteet voivat sisältää myös muita merkkejä.

Kansainvälisten standardinumeroiden kehitys alkoi ISBN-numeron (International Standard Book Number) (ISO 2108:1992) kehityksestä 1960-luvun puolivälissä. Kirjalalla oli havaittu tarve kansainväliselle ja tehokkaalle kirjojen numerointijärjestelmälle, kun kirjojen tilauksia ja varastokirjanpitoa alettiin hoitaa tietokoneilla. Numerointijärjestelmää kehitettiin 1960-luvun loppupuolella, ja ISBN-standardin ensimmäinen versio hyväksyttiin vuonna 1970. (International ISBN Agency 2001, 1)

Ajan myötä ISO on kehittänyt useita numerointijärjestelmiä eri tarkoituksiin. Niitä ovat:

- ISSN (International Standard Serial Number) sarjajulkaisujen identifiointiin (ISO 3297:1998 ja ISSN International Centre 2002)
- ISRC (International Standard Recording Code) äänitteiden identifiointiin (ISO 3901:2001 ja IFPI Secretariat 2002)
- ISMN (International Standard Music Number) painettujen musiikkijulkaisujen identifiointiin (ISO 10957:1993)
- ISWC (International Standard Musical Work Code) musiikkiteosten identifiointiin (ISO 15707:2001). Teoksella tarkoitetaan tässä IFLA:n mallin

(Plassard 1998) mukaista abstraktiota eikä mitään tiettyä ilmentymää. Teoksen ilmentymät voidaan puolestaan identifioida esimerkiksi ISRC:n ja ISMN:n avulla.

- ISRN (International Standard Technical Report Number) teknisten raporttien ja tutkimusten identifiointiin (ISO 10444:1994).

Lisäksi kehitteillä ovat ainakin audiovisuaalisen materiaalin identifiointiin tarkoitettu ISAN (International Standard Audiovisual Number) (esim. Finseth 2002) ja tekstimuotoisten teosten identifiointiin tarkoitettu ISTC (International Standard Text Code) (esim. ISTC Working Group 2002). ISAN on valmistuessaan ISO:n standardi numero 15706 ja ISTC numero 21047. Molempien standardien kehitystyö on pitkällä ja numerot otettaneen käyttöön lähitulevaisuudessa.

Numerojärjestelmät muistuttavat toisiaan, mutta niissä on yksi perusteellinen ero: Toiset numerot, kuten ISSN tai ISTC eivät kerro identifioitavasta kohteesta mitään (ISO 2108:1992, 2 ja ISTC Working Group 2002). Jotkin numerot, kuten ISBN, identifioivat myös julkaisijan ja voivat sisältää myös muuta tunnisteseen koodattua tietoa. Seuraavissa kohdissa kerron tarkemmin tekstimuotoisten julkaisujen standardinumeroista.

3.1.1 International Standard Book Number (ISBN)

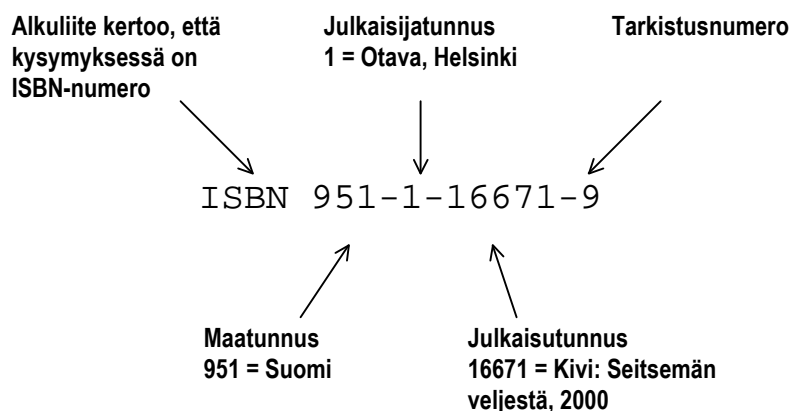
International Standard Book Number (ISBN) (ISO 2108:1992) on kehitetty alun perin kirjojen tunnistejärjestelmäksi, mutta nykyisin ISBN-numero voidaan antaa paitsi kirjoille myös kirjan kaltaisille julkaisuille. Vaikka standardia on muutettu, numeron rakenne ei kuitenkaan ole muuttunut. (International ISBN Agency 2001, 1) Kymmenestä merkistä koostuva tunniste on osoittautunut käytännölliseksi, ja numeron käyttö on suosittua useissa tarkoituksissa ja erityisesti ATK-pohjaisissa järjestelmissä. (International ISBN Agency 2001, 2)

Kuviossa 3 esitetään esimerkki ISBN-numerosta sekä selvitetään tunnisteen eri osat. Tunniste koostuu kymmenestä merkistä, joista yhdeksän ensimmäistä ovat numeroita nollasta yhdeksään ja kymmenes merkki joko numero tai kirjain X. Tunniste jakautuu

neljään osaan, jotka kirjoitettuna erotetaan toisistaan väliviivalla (-) tai välilyönnillä. Osien pituudet voivat vaihdella. Tunnisteen eteen lisätään myös kirjaimet ISBN, jotta se voidaan erottaa muista tunnisteista. (ISO 2108:1992, 1—2)

Ensimmäinen osa identifioi kansallisuuden tai kieliryhmän, ja sen määrittämisestä vastaa kansainvälinen ISBN-keskus (ISO 2108:1992, 2). Suomen tunnuksia ovat 951 ja 952. Toinen numerosarja identifioi yhden julkaisijan julkaisijaryhmän sisällä. Jos julkaisijan oletetaan tuottavan runsaasti julkaisuja, sille annetaan lyhyt julkaisijatunnus (ISO 2108:1992, 2). Merkkien lukumäärä ei saa ylittää kymmentä, joten lyhyt julkaisijatunnus jättää tilaa pidemmille julkaisutunnuksille. Esimerkissä käytetty julkaisijatunnus 1 tarkoittaa Helsingissä toimivaa Kustannusosakeyhtiö Otavaa. Kolmas osa identifioi julkaisun julkaisijan kaikkien nimekkeiden joukosta ja sen pituus määräytyy kahden edellisen osan perusteella. Otavan julkaisu 16671 tarkoittaa vuonna 2000 painettua Aleksis Kiven Seitsemää veljestä. Neljäs osa on tarkistusnumero, joka lasketaan muiden yhdeksän numeron perusteella. Tarkistusnumero voi olla luku nolasta yhdeksään tai kirjain X.

Kun ISBN myönnetään, sitä ei voida käyttää enää toista kertaa. Jokaiselle esitysmuodolle (esimerkiksi kova- ja pehmytkantiselle kirjalle ja sähköiselle kirjalle) on annettava oma ISBN-numeronsa. Numero on uusittava myös, jos julkaisusta tuotetaan korjattu versio.



KUVIO 3. ISBN-numeron rakenne.

3.1.2 *International Standard Serial Number (ISSN)*

International Standard Serial Number (ISSN) (ISO 3297:1998) voidaan antaa mille tahansa julkaisulle, joka ilmestyy sarjana ilman määrättyä loppua. Se voidaan antaa muun muassa lehdille, kausijulkaisuille, vuosikirjoille ja niin edelleen. ISSN-numeron voi liittää myös missä tahansa muodossa julkaistuun elektroniseen julkaisuun, jos se täyttää sarjalle asetetut kriteerit. Tämä tarkoittaa sitä, että julkaisusarjan osat julkaistaan avoimena dokumenttien sarjana saman otsikon alla, ja sarjan päättymisajanjaksoa ei ole rajoitettu etukäteen. (ISSN International Centre 2002) ISBN:stä poiketen ISSN-numero ei sisällä itsessään mitään tietoa julkaisun alkuperästä (ISO 3297:1998, 2).

ISSN-tunniste koostuu kahdesta neljän merkin sarjasta, jotka erotetaan toisistaan väliviivalla (-). Esimerkiksi ISSN 1041-4347 identifioi julkaisun IEEE Transactions on Knowledge and Data Engineering. Tunnisteen seitsemän ensimmäistä merkkiä ovat numeroita, ja ISBN-numeron tavoin sen viimeinen merkki on tarkistusnumero, joka voi olla numero nolasta yhdeksään tai kirjain X. (ISO 3297:1998, 1—2)

Julkaisulle on annettava uusi ISSN-numero, jos sen otsikko muuttuu. Muut mahdolliset muutokset, kuten julkaisijan tai julkaisupaikan vaihtuminen eivät vaikuta numeroon. Jos kuitenkin vaihdetaan julkaisuvälinettä esimerkiksi siirryttäessä painetusta julkaisusta elektroniseen, julkaisusarjalle on luotava uusi ISSN-numero. (ISSN International Centre 2002)

ISSN-numero identifioi tietyn julkaisusarjan, ei siis tiettyä julkaisua sarjassa (ISSN International Centre 2002). Jos halutaan viitata sarjan tiettyyn osaan, ISSN-numeron yhteyteen on liitettävä tieto esimerkiksi julkaisuvuodesta tai julkaisun järjestysnumerosta sarjan sisällä. Sarjajulkaisuissa voidaan käyttää ISBN-numeroa ISSN-numeron rinnalla, jolloin ISSN identifioi julkaisusarjan ja ISBN yksittäisen julkaisun. Julkaisusarjan sisällä olevaan artikkeliin voidaan viitata myös esimerkiksi SICI-koodilla (Standard Information and Contribution Identifier) (NISO Press 1996), joka käyttää ISSN-numeroa tunnisteen osana. SICI-järjestelmää käsitellään kohdassa 3.2.

3.1.3 International Standard Text Code (ISTC)

International Standard Text Code (ISTC) on numerointijärjestelmä, jota kehitetään tekstimuotoisten teosten identifiointiin. Tunnistejärjestelmän arvellaan olevan käytössä jo vuoden 2002 loppuun mennessä. (ISTC Working Group 2002)

Tekstimuotoisella teoksella tarkoitetaan teosta, joka koostuu pääsääntöisesti sanoista. Sellaisia ovat muun muassa esseet, romaanit, runot, käsikirjoitukset ja novellit. Teoksella tarkoitetaan juuri IFLA:n mallin (Plassard 1998) mukaista abstraktiota jonkin ilmauksen taustalla. ISTC-koodilla ei ole siis tarkoitus identifioida jonkin teoksen ilmentymiä. (ISTC Working Group 2002)

ISTC poikkeaa muista standardinnumeroista siten, että tunniste koostuu kuudestatoista heksadesimaaliluvusta, joita ovat numerot 0–9 sekä kirjaimet A:sta F:ään. ISTC:n neljä elementtiä käsittävät rekisteröintitoimiston tunniste, vuosiluvun neljällä numerolla ilmoitettuna, teos-elementin ja tarkistusnumeron. Tulostettuna elementit erotetaan toisistaan välilyönnillä tai väliviivalla (-). ISTC-koodi voi siten olla esimerkiksi ISTC 0A9-2002-12B4A105-6. (ISTC Working Group 2002)

ISTC ei sisällä lainkaan älykkyyttä, eli se ei kerro teoksesta mitään. Tunnisteesta ei voi esimerkiksi selvittää, kuka on teoksen tekijä tai omistaja. Sillä ei myöskään ole mitään tekemistä tekijänoikeuksien kanssa. (ISTC Working Group 2002)

Numeroiden antamiseen eri teoksille pätee samat säännöt kuin IFLA:n mallissa: jos uuden version tuottaminen vaatii merkittävästi itsenäistä taiteellista ponnistusta, sitä pidetään uutena teoksena. (Plassard 1998, 16–17) Esimerkiksi tietystä kirjasta tehty näytelmäsovitus on oma teoksensa, joka vaatii oman tunnisteensa. (ISTC Working Group 2002)

3.1.4 International Standard Technical Report Number (ISRN)

International Standard Technical Report Number (ISRN) (ISO 10444:1994) on korkeintaan 36:sta alfanumeerisesta merkistä koostuva tunniste, jolla identifioidaan

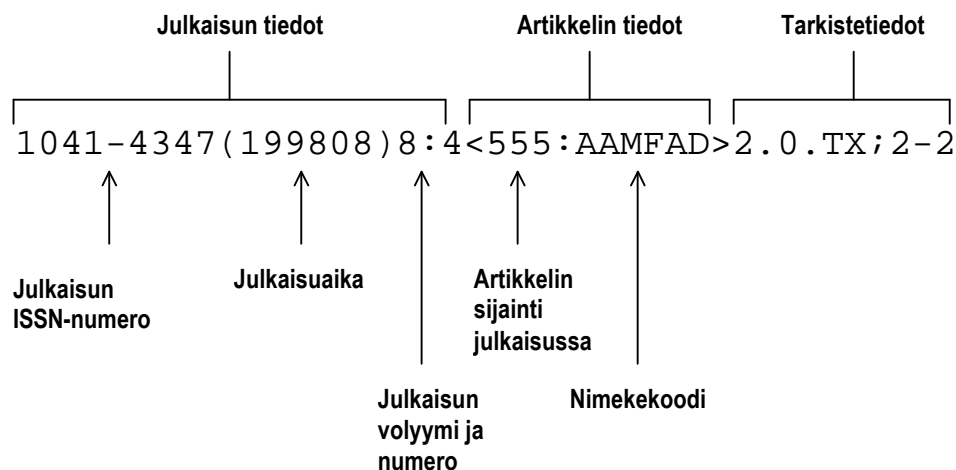
tekniisiä raportteja. Teknisellä raportilla tarkoitetaan yleensä jonkin tutkimuksen tuloksena tuotettavaa dokumenttia, joka ei tavallisesti ole luonteeltaan kaupallinen. Raportit voidaan julkaista jonkin sarjan osana tai yksittäisinä julkaisuina. (ISO 10444:1994, 1)

ISRN-koodi koostuu kolmesta osasta, joiden lisäksi tunnisteessa voi olla myös paikallinen loppupääte. Tämä ei kuitenkaan virallisesti ole osa ISRN-tunnistetta. (ISO 10444:1994, 1) Kuviossa 4 esitellään tunnisteiden osat. Esimerkin tunniste on kuvitteellinen eikä identifioi mitään todellista resurssia.

Tunnisteiden ensimmäinen osa käsittää korkeintaan 16 merkkiä, ja se päättyy ryhmäerottimeen, joka on kaksoisviiva (--). Tällä koodilla voidaan ilmaista esimerkiksi organisaatio tai sen osa, joka on vastuussa dokumentista, sekä sarja, johon dokumentti kuuluu. (ISO 10444:1994, 2) Jyväskylän yliopiston tietojenkäsittelytieteiden laitos voisi esimerkiksi haluta projektidokumentaatiolleen ISRN-tunnisteet, joiden alkuosa on JY/TKTL-PROJ. Toinen osa identifioi yhden julkaisun edellisen ryhmän sisältä. Sen pituus on korkeintaan 14 merkkiä, ja se koostuu kahdella numerolla ilmaistusta vuosiluvusta, järjestysnumerosta ja mahdollisesti versiotunnuksesta. (ISO 10444:1994, 3) Esimerkin dokumentin valmistumisvuosi on 2002 ja järjestysnumero 123. Lisäksi on haluttu erottaa dokumentin erikieliset versiot toisistaan, ja liitetty tunnisteeseen EN ilmoittamaan, että dokumentti on kirjoitettu englanniksi. Kolmas osa käsittää maatumuksen, mutta se ei ole pakollinen (ISO 10444:1994, 4). Osa erotetaan edellisestä osasta ryhmäerottimella (--).

Maatumuksen jälkeen on mahdollista liittää tunnisteeseen paikallinen osa, joka voi sisältää mitä tahansa tärkeäksi katsottua tietoa ja jonka pituutta ei ole rajoitettu. Se erotetaan muusta tunnisteesta plus-merkillä (+) ja voi sisältää kirjaimia, numeroita, pilkkuja, pisteitä ja kenoviivoja (/). (ISO 10444:1994, 4) Paikallisella päätteellä voidaan kertoa esimerkiksi tiedostomuoto kuten kuviossa 4 olevassa esimerkissä.

ja mahdollisesti myös julkaisukuukausi ja -päivä, vuodenaika tai vuosineljännes. (NISO Press 1996, 10) Esimerkin artikkeli on julkaistu elokuussa 1998. Vuosiluvun jälkeen identifioidaan julkaisun numero (NISO Press 1996, 11—13). Merkintä 8:4 tarkoittaa lehden volyymia 8 ja sen numeroa 4.



KUVIO 5. Serial Item and Contribution Identifier -koodin rakenne.

Artikkelia koskevat tiedot sijoitetaan kulmasulkeiden väliin. Sen ensimmäinen osa ilmoittaa artikkelin sijainnin julkaisussa. Elektronisessa muodossa julkaistussa artikkelissa sijaintitieto voi jäädä tyhjäksi. (NISO Press 1996, 13) Esimerkissä sijaintitieto 555 tarkoittaa sivun numeroa, jolta artikkeli alkaa. Artikkelin tietojen toinen osa on nimekekoodi, joka muodostetaan artikkelin otsikon kuuden ensimmäisen sanan alkukirjaimista. (NISO Press 1996, 13—14) Esimerkin nimekekoodi AAMFAD on muodostettu artikkelin otsikosta An Authorization Model for a Distributed Hypertext System.

Kolmas osa, tarkistetiedot kertoo, miten tunniste tulee tulkita. Sen ensimmäinen numero Code Structure Identifier (CSI) voi saada kolme arvoa, 1, 2 tai 3. Jos arvo on 1, tunniste identifioi kokonaisen julkaisun. Jos se on 2, tunnisteella viitataan artikkeliin julkaisun sisällä. Jos julkaisija haluaa käyttää esimerkiksi julkaisun editointivaiheessa tunnisteiden yhteydessä standardista poikkeavia sisäisiä numerointijärjestelmiä, CSI:lle annetaan

arvo 3. (NISO Press 1996, 8) Seuraava numero erotetaan CSI:stä pisteellä. Se on Derivative Part Identifier (DPI). Numerolla ilmaistaan onko kyse artikkelista (0), sisällysluettelosta (1), hakemistosta (2) vai abstraktista (3). (NISO Press 1996, 8) DPI:n jälkeen sijoitetaan Medium/Format Identifier (MFI). Se on pituudeltaan kahden kirjaimen mittainen ja ilmaisee dokumentin formaatin. Kirjaimet TX ilmoittavat, että artikkeli on ilmestynyt painettuna tekstinä. Toinen vaihtoehto voi olla esimerkiksi CO, jolla ilmaistaan, että artikkeli on julkaistu verkossa. (NISO Press 1996, 8–9). Puolipisteen jälkeen kerrotaan tunnisteiden versio. Vuoden 1996 standardia noudattavilla tunnisteilla versionumero on 2, ja sitä varhaisemmilla 1. (NISO Press 1996, 9) Version jälkeen sijoitetaan väliviiva (-) ja tarkistusnumero, joka lasketaan tietyllä algoritmilla (NISO Press 1996, 9).

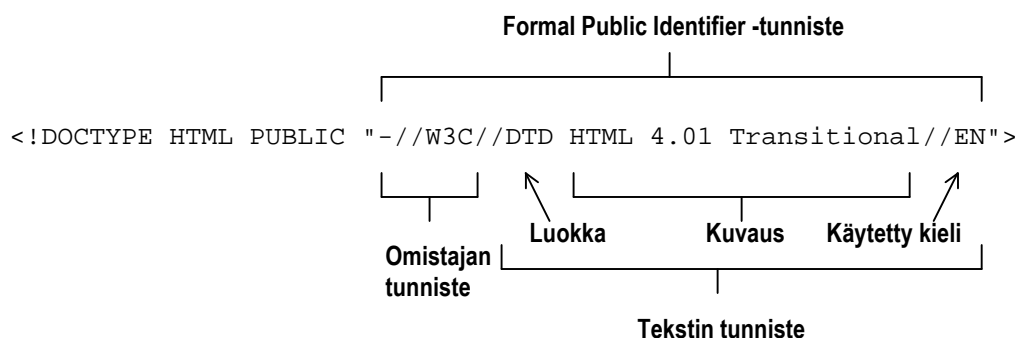
Hakala (2002) näkee SICI-tunnuksessa tällä hetkellä pahimpana putteena sen, että jonkin artikkelin eri muodossa olevia elektronisia versioita ei voi erottaa toisistaan. Siten esimerkiksi digitaalisessa arkistossa tunnisteiden käyttö artikkeleiden identifioinnissa on ongelmallista.

3.3 Formal Public Identifier (FPI)

Formal Public Identifier (FPI, muodollinen julkinen tunniste) on SGML-standardissa (Standard Generalized Markup Language) (ISO 8879:1986) määritelty tunnisteiden muoto. Se koostuu omistajan tunnisteesta ja tekstin eli resurssin tunnisteesta, jotka erotetaan toisistaan kahdella kenoviivalla (/). ISO:a lukuun ottamatta organisaatiot aloittavat oman tunnisteensa merkeillä +// tai -// riippuen siitä, onko organisaatio rekisteröitynyt tunnisteensa vai ei (Goldfarb 1990, 383–385). Formal Public Identifierin loppuosa on tekstin tunniste, joka identifioi resurssin omistajaorganisaation sisällä. Se sisältää ainakin tekstin luokan ja kuvauksen sekä mahdollisesti muita osia kuten käytetyn kielen. Luokka ja kuvaus erotetaan toisistaan välilyönnillä ja muut osat kahdella kenoviivalla. (Goldfarb 1990, 385–390)

Tunniste lienee tutuin SGML-pohjaisten dokumenttien alkuosasta, jossa määritellään dokumentin tyyppi (KUVIO 6). Muodollisen julkisen tunnisteiden osuus on kuvion esimerkissä sijoitettu lainausmerkkien väliin. Miinusmerkki ennen omistajan nimeä

kertoo, että tekstin omistaja, World Wide Web Consortium (W3C), ei ole rekisteröitynyt tunnistettaan. Tekstin luokka puolestaan on DTD eli dokumenttityyppimäärittely ja sen kuvaus on HTML 4.01 Transitional. Viimeinen osa ”EN” viittaa kaikkein ilmeisimmin siihen, että dokumentti on englanninkielinen.



KUVIO 6. Formal Public Identifier dokumenttityyppimäärittelyssä.

3.4 Digital Object Identifier (DOI)

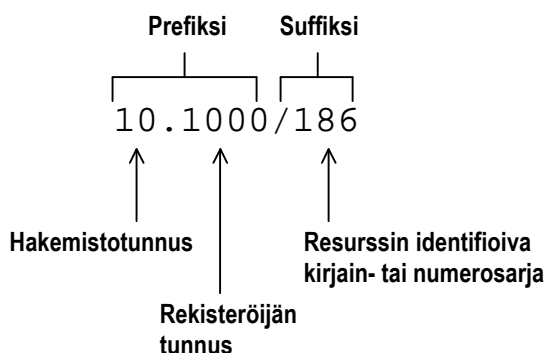
Digital Object Identifier (DOI) on eräs ratkaisu resurssien pysyväksi tunnisteeksi. Sillä voidaan identifioida mitä tahansa digitaalisia tai fyysisiä resursseja millä tahansa abstraktiotasolla. Tunniste pysyy samana vaikka resurssin omistajuussuhteet tai fyysinen sijainti muuttuisivat. (Paskin 2001, 15)

DOI on periaatteessa puhtaasti tyhjä tunniste, joka ei tarjoa mitään tietoja identifioitavasta resurssista (Paskin 2001, 17). Vanhoja käytössä olevia tunnisteita, kuten ISBN:ää, voidaan kuitenkin käyttää DOI:n osana, mikä helpottaa käyttäjiä liittämään uuden tunnisteeseen olemassa oleviin resursseihin. DOI:lla voidaan identifioida myös osia suuremmista resursseista sekä tietotyyppejä, joita varten ei ole aiemmin ollut olemassa mitään tunnustuskeemaa. (Paskin 2001, 18)

DOI tarjoaa käyttäjilleen kuitenkin enemmän kuin pelkän yksilöllisen tunnisteeseen, joka on vain yksi osa järjestelmää (Paskin 2001, 15). Sen lisäksi järjestelmään kuuluu

resurssin kuvaaminen metadatatalla, resurssien resoluutio eli resurssin tai sen metatietojen haku tunnisteiden perusteella sekä säännöt, jotka hallitsevat järjestelmän toimintaa (Paskin 2001, 17).

Kuvio 7 esittää esimerkin DOI-tunnisteesta. Merkkijono koostuu kahdesta osasta, prefiksistä ja suffiksista. Prefiksi puolestaan koostuu kahdesta osasta, hakemiston tunnuksesta (Directory Code), joka on toistaiseksi aina numero 10 sekä rekisteröijän tunnuksesta (Registrant Code). Tunnukset erotetaan toisistaan pisteellä. Prefiksin ja suffiksin välillä on kenoviiva (/). Suffiksi on resurssin tunnus, ja se voi olla mikä tahansa yksilöivä numero- tai kirjainsarja organisaation sisällä. Se voi olla esimerkiksi juokseva numero, mutta voi usein olla helpompaa käyttää siinä jo jotain olemassa olevaa tunnistetta, kuten julkaisun ISBN-numeroa. (Paskin 2001, 23-24) DOI:n syntaksi on myös standardoitu Yhdysvalloissa ANSI/NISO-standardiksi Z39.84-2000 (NISO Press 2000).



KUVIO 7. Digital Object Identifier -tunnisteen rakenne.

Teknisesti DOI perustuu CNRI:n (Corporation for National Research Initiatives) kehittämään *kahvajärjestelmään (Handle System)* (esim. Sun & Lannom 2002), joka on yhdenmukainen URN-kehyksen kanssa (Payette 1998). Kahva on merkkijono, joka identifioi digitaalisen kohteen jossain järjestelmässä riippumatta kohteen sijainnista. (Grass & Arms 1994) DOI koostuu kahvan tavoin kahdesta osasta: nimen antavan tahon

tunnuksesta (prefiksi) sekä resurssille annetusta nimestä (suffiksi). Kahvassa osat myös erotetaan toisistaan kenoviivalla (/) (Sun & Lannom 2002).

Hakalan (2002) mukaan järjestelmässä on kuitenkin puutteita, joiden perusteella DOI:n käyttö kirjastojärjestelmissä on kyseenalaista. Hän mainitsee ongelmiksi muun muassa riippuvuuden HTTP-protokollasta, kaupallisuuden sekä riippuvuuden yhdestä organisaatiosta, International DOI Foundationista. Lisäksi kahvajärjestelmän standardointi ei ole onnistunut aiotulla tavalla. Tähän Hakala epäilee syyksi, että Internetin standardeja kehittävät organisaatiot eivät halua tukea järjestelmää, joka olisi niiden oman URN-järjestelmän kilpailija.

Pahimpana ongelmana Hakala (2002) kuitenkin pitää sitä, että DOI:n sisällä voi käyttää mitä tahansa identifiointitunnusta. DOI ei määrittele syntaksissaan, millä tavalla eri nimeämisskeemoista johdetut tunnisteet tulisi merkitä. Esimerkiksi ISBN-numero voi toimia DOI-suffiksina sellaisenaan tai sen edessä voi olla kirjaimet ISBN. DOI:n syntaksin standardoinnin arvo onkin siten vähäinen. (Hakala 2002)

DOI-tunnisteiden resoluutio tapahtuu nykyisellään keskitetysti yhden palvelimen kautta. Käytännössä jokaisen DOI-tunnisteen eteen lisätään toistaiseksi <http://dx.doi.org>, jolloin kaikki resursseihin kohdistuvat pyynnöt osoitetaan tälle DOI-resoluutiopalvelimelle. Resoluutiopalvelin puolestaan ohjaa resurssipyynnön sen todelliseen määränpäähän. Hakala (2002) pohtiikin, miten toimiva yhteen ainoaan palvelimeen perustuva maailmanlaajuinen järjestelmä on.

3.5 Universal Unique Identifier (UUID)

UUID (Universal Unique Identifier) on 128-bittinen numero, jolla voidaan identifioida mikä tahansa entiteetti Internetissä. Se voidaan luoda millä tahansa algoritmilla, mutta jollekin entiteetille annetun UUID:n pitäisi olla suurella todennäköisyydellä erilainen kuin millään muulla entiteetillä. Tämä taataan jakamalla tunniste komponentteihin, joiden yhdistelmä on suurella todennäköisyydellä ainutlaatuinen. Esimerkiksi The Open Group (1997) käyttää UUID:n osien luomisessa käytetyn tietokoneen verkko-osoitetta, aikaleimaa ja satunnaislukuja. UUID koostuu viidestä kentästä, jotka koostuvat

heksadesimaaliokteteista ja erotetaan toisistaan viivalla (-). Oktetit puolestaan muodostuvat kahdesta heksadesimaalinumerosta, joita ovat luvut 0:sta 9:ään ja kirjaimet A:sta F:ään. Neljä ensimmäistä kenttää muodostetaan aikaleiman pohjalta ja viides kuuden oktetin kenttä verkko-osoitteesta. UUID voi siten olla esimerkiksi 2fac1234-31f8-11b4-a222-08002b34c003.

3.6 Kansallisbibliografian ID-tunnus (NBN)

Kansallisbibliografian ID-tunnus eli NBN-tunnus (National Bibliographic Number) on kansainvälinen kansalliskirjastojen käyttämä identifiointijärjestelmä. Eri maissa järjestelmä on kuitenkin toteutettu eri tavoin ja takuuta tunnisteiden ainutkertaisuudelle ei ole (Hakala 2002).

Suomessa annetut tunnukset ovat perinteisesti muodostuneet kirjaimesta f, vuosiluvusta ja juoksevasta numerosta. Internet-käytössä tunnukset annetaan kuitenkin hieman erilaisin periaattein. Ensinnäkin elektronisten julkaisujen tunnisteisiin lisätään alkuun Suomen maatunnus fi, jotta taataan tunnusten maailmanlaajuinen ainutkertaisuus. Toiseksi perinteisen f-kirjaimen perään lisätään vielä e-kirjain kertomaan, että kyseessä oleva aineisto on elektronisessa muodossa. (Hakala 2002) Elektronisen julkaisun tunnistete voi siten olla esimerkiksi fi-fe20021301.

Suomen verkkojulkaisujen kansallinen arkisto on automatisoinut NBN-tunnuksien antamisen. Tunniesteiden laskemiseen käytetään MD5-algoritmia (Rivest 1992), joka tuottaa resurssille sen sisällön perusteella 128-bittisen tarkistussumman eli 32-merkkisen heksadesimaaliluvun. Tätä tarkistusnumeroa käytetään resurssin tunniesteena lisäämällä sen eteen alkuliite fea. On vain häviävän pieni todennäköisyys sille, että kaksi resurssia tuottaisi saman tarkistussumman, joten numeroa voidaan käyttää tunniesteena. (Hakala 2002)

NBN on tärkeä tunnistete juuri verkossa julkaistun materiaalin identifioinnissa. Hakala (2002) huomauttaa, että useille verkkoresursseille ei voi eikä edes kannata antaa perinteisiä identifiointitunnuksia.

3.7 Uniform Resource Name (URN)

Uniform Resource Name (URN) (Moats 1997) on Uniform Resource Identifierin (URI) (Berners-Lee et al. 1998) osa, joka on tarkoitettu resurssien pysyvien tunnisteen luomiseen. URI on tiivis merkkijono, jolla identifioidaan tai paikannetaan abstrakti tai fyysinen resurssi (Berners-Lee et al. 1998, 2). URI:n toinen osa, URL (Uniform Resource Locator) (Berners-Lee et al. 1994) puolestaan on World Wide Webissä laajassa käytössä oleva tekniikka resurssien paikantamiseen. URL-viitteitä käsitellään tarkemmin seuraavassa luvussa.

URN eroaa URL:sta siinä, että sen ensisijainen tarkoitus on olla pysyvä tunniste jollekin resurssille, kun taas URL-osoitteiden pysyvyydestä ei ole mitään takeita. URN:ää kehitettäessä on siis haluttu vetää selkeä raja tunnisteen ja resurssin todellisen sijainnin välille. URN:n avulla on kuitenkin voitava selvittää resurssin todellinen sijainti.

URN:ää voidaan pitää yleisnimenä joukolle toisiinsa verrattuna hyvinkin erilaisia tunnisteita. Ne voidaan muodostaa käyttäen lähes mitä tahansa nimiavaruutta, ja jokainen nimiavaruus määrää tavan, jolla tunniste annetaan. Yhteistä tunnisteille on, että ne alkavat urn-alkuliitteellä, jonka jälkeen kerrotaan käytetty nimiavaruus sekä nimiavaruuden mukaisesti annettu tunniste. Osat erotetaan toisistaan kaksoispisteillä (:)
(Moats 1997, 1).

Yksilöllisyyden ja pysyvyyden lisäksi URN-tunnusten toiminnallisiin vaatimuksiin kuuluvat muun muassa mahdollisuus liittää mikä tahansa olemassa oleva tai tuleva identifiointijärjestelmä osaksi URN:ää, jos se muuten täyttää tunnuksen syntaktiset vaatimukset tyydyttävästi. (Sollins & Masinter 1994, 3)

Monet identifiointijärjestelmät sopivatkin URN-skeemaan sellaisenaan, ja joistain on tehty omat URN:n mukaiset spesifikaationsa (esim. Hakala & Walravens 2001). Esimerkiksi edellisessä kohdassa esimerkkinä NBN-tunnus voidaan helposti muuntaa URN:n syntaksin mukaiseksi. Tunnus olisi silloin URN:NBN:fi-fe20021301. Alku kertoo kyseessä olevan URN-tunnisteen ja seuraava osa taas identifioi käytettävän nimiavaruuden, NBN.

ISBN- tai ISSN-numerosta muodostettu URN-tunniste ei vaadi mitään muutoksia tunnisteiden numero-osaan (Lynch et al. 1998, 4 ja 5). Mistä tahansa ISBN-numerosta voidaan siten muodostaa URN-tunniste lisäämällä numerosarjan eteen URN-skeeman sekä ISBN-nimiavaruuden määrittelyt. Lopullinen tunnus voi olla esimerkiksi URN:ISBN:951-1-16671-9.

SICI-koodi sisältää kuitenkin merkkejä, jotka eivät kuulu URN:n merkistöön, ja ne on muunnettava merkkien %-alkuisiksi vastineiksi (Lynch et al. 1998, 7). Aiemmin esimerkkinä käytetty SICI-koodi 1041-4347(199808)8:4<555:AAMFAD>2.0.TX;2-2 sisältää tunnisteiden sallittuihin merkkeihin kuulumattomat merkit < ja >, jotka on korvattava merkkijyhdistelmillä %3C ja %3E. Siten tunnisteesta muodostettu URN-tunniste olisi URN:SICI:1041-4347(199808)8:4%3C555:AAMFAD%3E2.0.TX;2-2.

URN-nimiavaruudet on kuitenkin rekisteröitävä Daigle et al.:in (1999) spesifioiman prosessin mukaisesti. Tällä hetkellä laillisen URN-tunnisteiden voi luoda ISBN-numeron, ISSN-numeron ja kansallisbibliografian ID-numeron perusteella niistä laadittujen ohjeiden (Hakala & Walravens 2001, Rozenfeld 2001 ja Hakala 2001) mukaan. Nimiavaruuksien rekisteröinti on kuitenkin aktiivisesti käynnissä. Hakalan (2002) mukaan seuraavaksi työn alla ovat SICI- ja ISTC-nimiavaruuksien rekisteröinnit.

Hakala (2002) kertoo, että URN-jakelun käynnistyttyä kuka tahansa on voinut Suomessa identifioida oman julkaisunsa. Helsingin yliopiston kirjaston WWW-sivuilla onkin jonkin aikaa ollut vapaasti käytettävissä ohjelma, joka luo yksilöllisen URN-tunnisteiden kansallisbibliografian ID-tunnuksen (NBN) perusteella.

3.8 Järjestelmien sisäisistä tunnisteista

Tässä kohdassa käsitellään tapoja, joilla tietokantajärjestelmät identifioivat sisältämiään kohteita. Kerron lähinnä siitä, miten oliopohjaisissa tietokantajärjestelmissä voidaan muodostaa tunnisteita sekä miten niitä voidaan käyttää näiden järjestelmien ulkopuolella.

Olen rajannut relaatiokannat tämän tutkimuksen ulkopuolelle. Relaatiomallissa entiteetin identifiointi perustuu sen saamiin arvoihin, ja siihen voidaan viitata vain perusavaimella (primary key) (Cattell 1991, 85). Mielestäni dokumenttien tai muiden vastaavien monimutkaisten entiteettien hallinnassa oliopohjaisuus on kuitenkin käyttökelpoisempaa. Oliopohjaiset järjestelmät nimittäin pohjautuvat arvojen asemesta olioiden identiteetteihin, jolloin kohteisiin voidaan viitata yksilöllisellä sisäisesti luodulla numerolla, eli oliotunnisteella (object identifier, OID) (Cattell 1991, 85).

Khoshafian & Copelandin (1986, 406) mukaan monet ohjelmointikielet ja tietokantajärjestelmät suunnitellaan usein ottamatta huomioon tiedon pysyvyyttä, joten ne tukevat identifiointia huonosti. Ne saattavat käyttää muuttujien nimiä erottelemaan tilapäisiä olioita sekoittaen identiteetin siihen, miten oloon osoitetaan. Joskus taas olion identiteetti sekoitetaan sen attribuuttien saamiin arvoihin. Attribuuttien käyttö tunnisteena ei kuitenkaan ole hyvä ratkaisu. (Khoshafian & Copeland 1986, 406—409)

Attribuuttien käytön asemesta olioille luodaan identiteetti yksilöllisen oliotunnisteen avulla. Oliotunniste voidaan muodostaa monella tavalla. Khoshafian & Copeland (1986) ovat tarkastelleet joitain vaihtoehtoja. Tunnisteet vaihtelevat täysin fyysisestä puhtaasti loogiseen, ja näiden välimuotojakin on useita. Fyysiset tunnisteet sisältävät olion todellisen osoitteen eli sijainnin esimerkiksi tietokoneen kovalevyllä, kun taas loogiset tunnisteet on yhdistettävä niiden todelliseen sijaintiin jonkinlaisen indeksin avulla. (Khoshafian & Copeland 1986, 412—414)

Fyysisiä tunnisteita käytetään harvoin tietokantajärjestelmissä, sillä entiteetin siirtyessä paikasta toiseen myös tunniste muuttuu ja kaikki viittaukset entiteettiin on päivitettävä. Suositumpaa onkin käyttää rakenteista osoitetta, jossa on sekä fyysinen että looginen osa. (Cattell 1991, 151) Tunnisteen fyysinen osa sisältää tyypillisesti tiedostosegmentin ja sivun numeron, joiden perusteella löydetään oikea kohta levyltä nopeasti. Tässä yhteydessä *sivu* liittyy tiedostojärjestelmän toimintaan, ja tarkoittaa vakiomittaista levyhaun perusyksikköä. Tunnisteen fyysinen osa voi ilmoittaa myös levyn tai palvelimen, jolta tieto löytyy (Khoshafian & Copeland 1986, 413). Looginen osa taas sisältää paikan (slot) numeron, jonka perusteella voidaan päätellä olion todellinen sijainti sivulla. (Cattell 1991, 151)

Khoshafian & Copelandin (1986, 413) mukaan kaikkein tehokkain tekniikka identiteetin takaamiseen on puhtaasti loogisen sijaistunnisteen (surrogate) käyttö. Se voidaan luoda millä tahansa algoritmilla, joka takaa tunnisteiden yksilöllisyyden (Cattell 1991, 151). Sijaistunnisteet on yhdistettävä todellisiin osoitteisiin jonkinlaisen indeksin avulla. Sijaistunnisteista on käytössä myös niin sanottu tyypitetty sijaistunniste, jossa tunniste sisältää sekä olion tyyppin (tai luokan) tunnisteeseen että olion yksilöllisen tunnisteeseen. Tästä etuna on se, että olion tyyppi voidaan päätellä tunnisteesta noutamatta itse oliota. (Cattell 1991, 151)

Jotkin oliotietokannanhallintajärjestelmät mahdollistavat sen, että olioilla voi olla sekä ihmisen ymmärrettävissä oleva nimi että oliotunniste. Esimerkiksi dokumenttitietokannassa voidaan käyttää otsikoita viitattaessa dokumentteihin ja lukuihin. Nämä nimet ovat verrattavissa relaatiomallin perusavaimiin. Cattell (1991, 86) kutsuu näitä olioavaimiksi. Ne ovat kuitenkin käytössä vain mukavuussyistä.

Suurin näkyvä ero eri valmistajien tietokantajärjestelmien välillä on, että toisissa tunnisteet ovat piilossa, kun taas toisissa käyttäjällä on suora pääsy niihin. Tunnisteet voidaan esimerkiksi tulostaa, ja niitä voidaan käyttää myöhemmin löytämään sama olio. Toisissa järjestelmissä taas tunnisteita voidaan käyttää vain erityisissä kahvoissa, eikä niistä ole hyötyä tietokantajärjestelmän ulkopuolella. (Cattell 1991, 86) Hyvä esimerkki ensimmäisen kaltaisesta järjestelmästä on Lotus Notes/Domino. Se antaa jokaiselle tietokantaan tallennetulle dokumentille tunnisteeseen, jonka käyttäjä voi halutessaan nähdä ja jota voidaan käyttää esimerkiksi dokumentin noutamiseen järjestelmästä.

Lotus Notes/Domino on oliopohjainen tietokantajärjestelmä, jota voidaan käyttää muun muassa dokumenttienhallintajärjestelmien kehittämiseen. Lotus Notes on asiakasohjelmisto, joka käyttää Lotus Domino -palvelimelle sijoitettuja tietokantoja. Dominon hallinnoima perusentiteetti on nimeltään dokumentti, ja järjestelmän kaikki tiedot haetaan ja tallennetaan jonkin dokumentin kautta. Kaikki tietokannassa olevat dokumentit identifioidaan useilla tunnisteilla, joista käyttäjän tai sovelluskehittäjän kannalta olennaisin on Universal Note ID. Kyseinen tunniste on maailmanlaajuisesti yksilöllinen sillä rajoituksella, että jos tietokannasta on kahdella tai useammalla palvelimella toistensa kanssa synkronoituja eli niin sanottuja replikakopioita, kaikilla

tietyn dokumentin kopioilla on sama Universal Note ID (Lotus Development Corporation 2000). Tunniste on olennaisin siksi, että se on pysyvä, ja dokumentti voidaan hakea tietokannasta tämän tunnisteiden perusteella.

3.9 Identifiointijärjestelmän valinta

Käsittelin tässä luvussa useita identifiointijärjestelmiä ja tapoja muodostaa resursseille tunnisteita niiden perusteella. En kuitenkaan pysty tämän tutkimuksen perusteella sanomaan, mikä järjestelmästä olisi oppimisympäristön kannalta paras valinta tai olisiko se edes mikään edellä käsittelemistäni. Toisaalta on vaikuttanut, että kehitettävän järjestelmän kannalta on jopa lähes yhdentekevää, millä algoritmilla resurssin tunniste muodostetaan, jos vain lopputuloksena syntyvä merkkijono on ainutkertainen. Toisten identifiointijärjestelmien käyttöä kuitenkin rajoittaa haluttu granulariteetin taso. Jos esimerkiksi on tarvetta identifioida osia jostain suuresta kokonaisuudesta, pelkällä ISBN-numerolla ei päästä haluttuun lopputulokseen.

Yksinkertaisissa järjestelmissä resurssien identifiointi järjestelmän sisällä juoksevilla numerolla on varmasti toimiva ratkaisu, mutta uskon, että pidemmän päälle on kuitenkin kannattavaa luoda resursseille yksilölliset tunnisteet, jotka säilyvät ainutkertaisina, vaikka resurssi siirtyy järjestelmästä toiseen.

Järjestelmää toteutettaessa on kuitenkin mietittävä myös sitä, mikä on järjestelmän kannalta mielekkäin ratkaisu. ISBN:n tai jonkin muun standardinumeron hankkiminen jokaiselle pienellekin resurssille ei edes tulisi kysymykseen. Tällöin esimerkiksi Lotus Notesin kaltaiseen dokumenttien identifiointiin perustuva ratkaisu voi olla toimivin: resursseja hallitseva järjestelmä luo jokaiselle resurssille joka tapauksessa sisäisen tunnisteiden, jota voitaisiin käyttää myös esimerkiksi oikeuksien määrittämisissä.

Pelkkä tunnisteiden antaminen resurssille ei kuitenkaan riitä, vaan jonkin järjestelmän hallitsemille resursseille on luotava myös yksiselitteiset osoitteet. Seuraavassa luvussa käsitellään tätä ongelmaa.

4 RESURSSIEN PAIKANTAMINEN

Kun rakennetaan järjestelmää resurssien digitaaliseen oikeuksien hallintaan, ei riitä, että resursseille luodaan tunniste jollain identifiointimekanismilla. Niihin on voitava myös osoittaa. Yksinkertaisimmillaan osoittamiseen riittää, että tiedetään resurssin sisältävän tiedoston URL-osoite. Resurssin paikantaminen on kuitenkin haasteellisempaa, jos se käsittää vain osan jonkin tiedoston sisällöstä. Nykyiset tekniikat eivät tarjoa selkeää ratkaisua osan erottamiseen esimerkiksi HTML-tiedostosta.

Tässä luvussa käsitellään resurssien paikantamista ja erityisesti niiden rajaamista. Ensin esitellään johdantona aiheeseen WWW:ssä käytetty osoitusmekanismi, Uniform Resource Locator (URL), joka on Uniform Resource Identifierin (URI) osajoukko. Loppuosa luvusta käsittelee tilannetta, jossa jokin resurssi on osa jotain suurempaa kokonaisuutta, kuten HTML-tiedostoa, joka koostuu useista resursseista. Ratkaisua resurssin rajaamiseen etsitään muun muassa URI:n katkelmatunnisteiden (fragment identifier) ja XML-pohjaisten tekniikoiden avulla.

4.1 Uniform Resource Locator (URL)

Ennen kuin siirryn käsittelemään resurssien rajaamista, esittelen Uniform Resource Identifierin toisen osajoukon, Uniform Resource Locatorin (URL). URL identifioi resurssin sen sijainnin mukaan. URL ei kuitenkaan toimi pysyvänä tunnisteena, sillä resurssin sijainti voi muuttua.

URL noudattaa URI:n yleistä syntaksia. Sen mukaan tunniste alkaa käytettävän skeeman määrittelyllä. Skeemaa seuraa kaksoispiste (:) sekä skeeman määrittelemän syntaksin mukaan muodostettu skeemaspesifi osa. (Berners-Lee et al. 1998, 11) URL:lle tyypillisiä skeemoja ovat muun muassa Internet-käytöstä tutut `http` sekä `ftp`. URN-tunnuksien yhteydessä skeema on aina `urn`. Http-skeeman mukaiset URI:t sisältävät tavallisesti isäntäkoneen nimen tai Internet Protocol -osoitteen sekä resurssin hakemistopolun. Mitään yleistä rakennetta eri skeemojen skeemaspesifeille osille ei kuitenkaan ole. (Berners-Lee et al. 1998, 11)

Vaikka monet URL-skeemat on nimetty tiettyjen protokollien mukaan, tämä ei tarkoita, että URL:n ilmoittamassa sijainnissa olevaan resurssiin pääsee käsiksi vain näiden protokollien avulla. Yhdyskäytävät, välityspalvelimet, välimuistit ja nimiresoluutiopalvelut voivat olla apuna pääsyssä riippumatta alkuperäisestä protokollasta. Jonkin URL:n resoluutio voi myös vaatia useamman kuin yhden protokollan käyttöä. On esimerkiksi tyypillistä, että http-alkuisten URL:ien resoluutioon käytetään DNS-nimijärjestelmää (Domain Name System) ja Hypertext Transfer Protocol:ia (HTTP), jos resurssia ei löydy paikallisesta välimuistista. (Berners-Lee et al. 1998, 3)

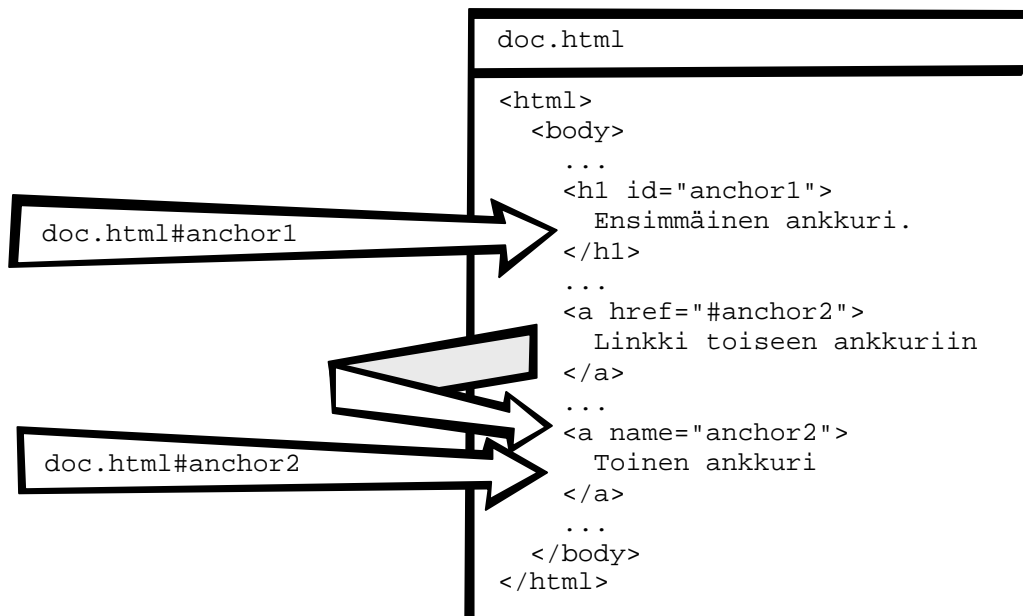
4.2 HTML: Katkelmatunnisteet ja URI

HTML (Hypertext Markup Language) (esim. Raggett et al. 1999) on tärkein sisällöntuotannon kieli World Wide Webissä. Sen käyttö on kuitenkin haasteellista oikeuksien hallinnan kannalta, jos yhden HTML-tiedoston sisältä halutaan identifioida pienempiä osia. Oikeuksien määrittelyn kannalta ihanteellisin tilanne olisi se, että identifioitava resurssi olisi yksi tiedosto, johon voidaan osoittaa yksinkertaisella tiedoston nimeen päättyvällä URL-osoitteella. Dokumentti voidaan silloin näyttää käyttäjälle sellaisenaan, jos oikeusmääritykset sen sallivat.

Kaikki resurssit ja niitä koskevat oikeusmääritykset eivät välttämättä kuitenkaan ole niin yksinkertaisia. Jostain resurssista voidaan haluta erilaisia näkymiä eri käyttäjille, joilla on resurssiin erilaiset käyttöoikeudet. Toisaalta voi olla, että vaikka käyttöä ei halutakaan rajoittaa, sitä halutaan seurata. Jokin resurssi voi esimerkiksi koostua pienemmistä osista, jotka ovat eri tekijöiden tuottamia. Näin ollen näiden pienempien resurssien tekijänoikeudet myös poikkeavat toisistaan. Oikeuksienhallintajärjestelmän kannalta nämä pienemmät osat ovat omia resurssejaan, jotka on voitava rajata suuremman resurssin sisältä. HTML-dokumenteilla tämä ei ole mahdollista, ellei dokumentille lisätä ylimääräisiä attribuutteja ja elementtejä.

Tällaisia attribuutteja ovat *name* ja *id*. Molemmat attribuutit identifioivat elementin siten, että sille voidaan luoda oma URL-osoitteensa. Osoitteeksi annetaan tiedoston osoite, jonka perään lisätään *katkelmatunniste (fragment identifier)*.

Katkelmatunnisteeksi annetaan sama arvo, joka on sijoitettu elementin id-tribuutin arvoksi, ja se erotetaan muusta URI:sta numeromerkillä (#). (Berners-Lee et al. 1998, 14—15) Kuviossa 8 esitetään esimerkki kahdesta URI:sta, jotka osoittavat kahteen ankkuriin HTML-dokumentin sisällä. Ankkurit identifioidaan tässä tapauksessa tunnisteilla `anchor1` ja `anchor2`, joista ensimmäinen identifioi erään ankkurielementin (`<a>`) ja toinen erään otsikkoelementin (`<h1>`). Ankkureita ja katkelmatunnisteita voidaan käyttää myös viittauksiin dokumentin sisällä ankkurista toiseen.



KUVIO 8. Ankkurit ja katkelmatunnisteet.

Katkelmatunnisteet eivät kuitenkaan periaatteessa rajaa mitään aluetta dokumentista, vaan ne vain osoittavat tiettyyn kohtaan. Katkelmatunnisteet myös tulkitaan vasta selaimessa. Yleinen tapa katkelmatunnisteiden tulkitsemiseen on se, että HTML-dokumentti ladataan selaimen kokonaisuudessaan, haluttu ankkuri etsitään dokumentista ja dokumenttia vieritetään ruudulla kyseiseen kohtaan asti.

Oikeuksien hallinnassa tämän tapainen dokumenttien käsittely on kuitenkin riittämätöntä. Sivut olisi tulkittava ja dokumenttien rajaukset tehtävä jo palvelimella. Tätä varten olisi kehitettävä oma ohjelmistonsa, joka lukee halutun HTML-dokumentin,

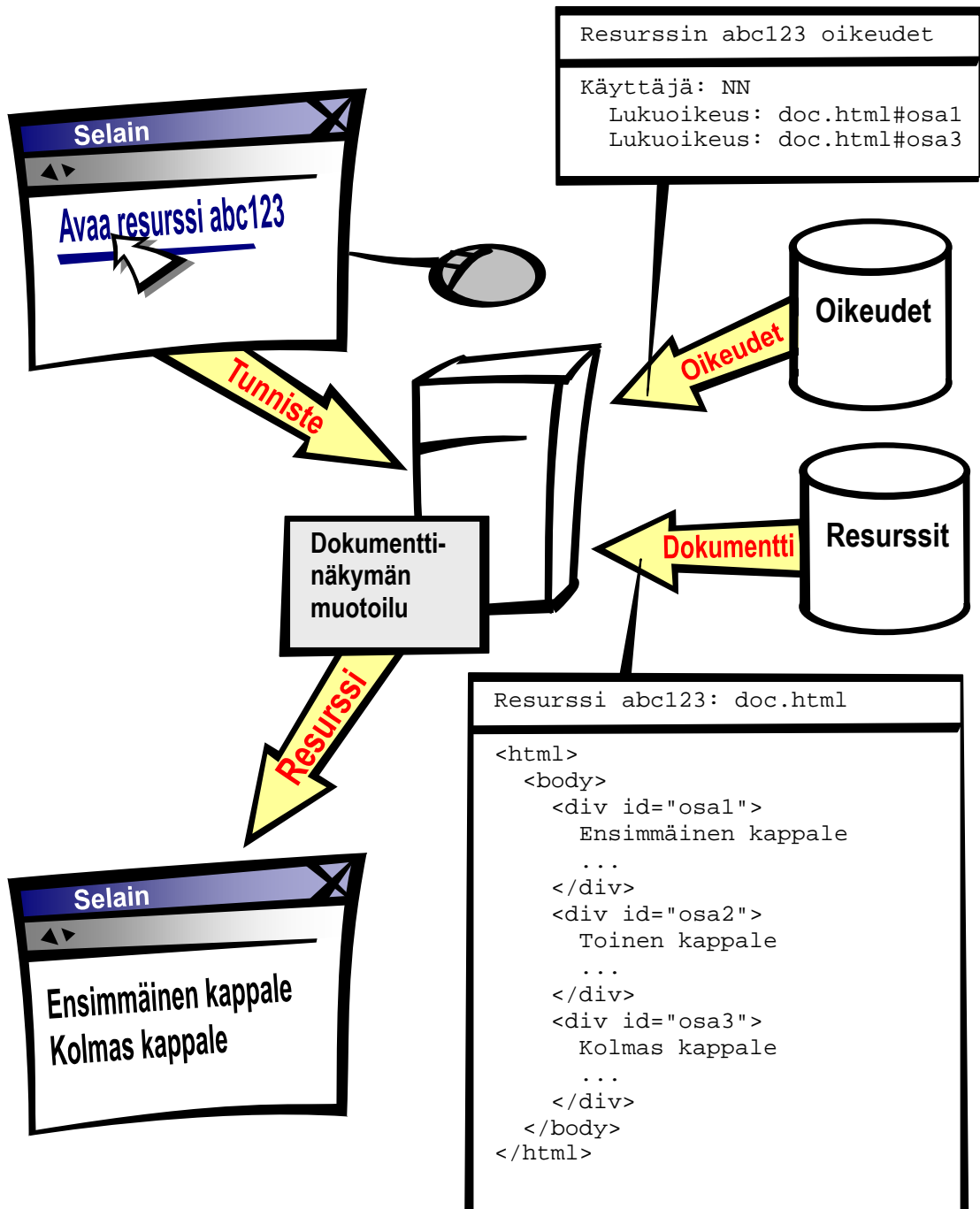
tarkastaa oikeudet sekä rajaa dokumentista näytettäväksi vain ne osat, joihin käyttäjän oikeudet riittävät. Eräs tapa tämän toteuttamiseen voisi mielestäni olla se, että HTML-dokumenteille lisättäisiin ylimääräisiä elementtejä, joilla sisältö rakenteistettaisiin osittain. Tähän soveltuvia elementtejä ovat esimerkiksi `div` ja `span`, joilla voi erottaa HTML-dokumentista osia. Jos dokumentin tietosisältö sijoitetaan näiden elementtien sisään, ja elementit identifioidaan id-tribuuteilla, tiettyyn osakokonaisuuteen voidaan osoittaa katkelmatunnisteella varustetulla URI:illa.

Kun käyttäjä haluaa avata tietyn HTML-dokumentin, oikeuksia hallitsevan järjestelmän tulisi noutaa haluttu dokumentti, tarkastaa käyttäjän oikeudet ja palauttaa tämän selaimen ne osat, joiden lukemiseen hänen oikeutensa riittävät. Kuviossa 9 esitetään esimerkki tästä periaatteesta. Esimerkissä käyttäjä lähettää selaimestaan oikeuksia hallitsevalle järjestelmälle pyynnön avata resurssi, jonka tunniste on abc123. Järjestelmä havaitsee, että resurssi sijaitsee HTML-dokumentilla doc.html, jonka käyttöoikeudet noudetaan oikeudet sisältävästä tietokannasta. Dokumentti doc.html on jaettu kolmeen osaan div-elementeillä. Elementeille on annettu tunnisteet osa1, osa2 ja osa3. Käyttäjälle on annettu lukuoikeudet dokumentin elementteihin, jotka on identifioitu tunnisteilla osa1 ja osa3. Oikeuksia hallitseva järjestelmä muotoilee dokumentista käyttäjälle näkymän dokumenttiin, josta jätetään pois tunnisteella osa2 identifioitu elementti. Tämä näkymä palautetaan käyttäjän selaimen.

4.3 Extensible Markup Language ja resurssien paikantaminen

HTML:n rajoitteet on tiedostettu ja on kehitetty muita merkkaukieliä, joilla tiedon rakennetta voidaan kuvata paremmin ja jotka eivät välttämättä edes ota kantaa esitettävän tiedon ulkoasuun. Näistä tunnetuin ja käytetyin lienee *XML* eli *eXtensible Markup Language* (esim. Bray et al. 2000). Kieltä on kehitetty vuodesta 1996 lähtien, ja se perustuu HTML:n tavoin SGML:ään (Standard Generalized Markup Language).

XML-dokumentit ovat rakenteeltaan puumaisia, ja kielen mukaan muodostettujen dokumenttien käsittelyssä on mahdollista käyttää tähän hierarkkiseen rakenteeseen perustuvia polkulauseita ja kyselykieliä. Seuraavissa kohdissa tarkastellaan XML:n perusolemuksen lisäksi näistä kielistä XPathia sekä XPointeria.



KUVIO 9. Dokumenttinäkymän luominen HTML-muotoiseen resurssiin.

XML kuvailee luokan tietokohteita, joita kutsutaan XML-dokumenteiksi ja osittain kuvailee niitä käsittelevien tietokoneohjelmien käyttäytymisen (Bray et al. 2000). XML-dokumentit koostuvat yksiköistä, joita kutsutaan entiteeteiksi. Entiteetit voivat sisältää joko *jäsennettyä (parsed)* tai *jäsentämätöntä* dataa. Jäsennetty data koostuu merkeistä, joista jotkut muodostavat *merkkietoa (character data)* ja jotkin *merkkausta (markup)*. Merkkkaus koostuu muun muassa elementtien *alku-* ja *lopputunnisteista (tag)* sekä *entiteettiviitteistä*, joilla kuvataan dokumentin looginen rakenne. (Bray et al. 2000) XML keskittyy perustaltaan vain tiedon rakenteeseen, ja ulkoasun määrittelyyn on kehitetty omia kieliään kuten eXtensible Stylesheet Language (XSL) (Adler et al. 2001).

Siinä missä HTML-kielessä käytössä olevat elementit on ennalta määrätty, XML-koodauksessa voidaan periaatteessa käyttää minkä tahansa nimisiä elementtejä. XML-dokumenttien on oltava vähintään *hyvin muodostettuja (well-formed)*, mutta ne voivat olla myös *valideja*. Hyvin muodostetut dokumentit noudattavat XML:n syntaksia, ja validit ovat jonkin *DTD:n (Document Type Definition)* mukaisia. (Bray et al. 2000)

XML-dokumenttien yhteydessä pienten resurssien irrottaminen kokonaisuudesta on HTML-dokumentteihin verrattuna helpompaa. XML-dokumentista voidaan valita tiettyjä elementtejä ja jopa osia elementtien sisältä. Tätä varten kehitettyjä kieliä ovat muun muassa XPath (XML Path Language) (Clark & DeRose 1999) ja XPointer (XML Pointer Language) (mm. Grosso et al. 2002b ja DeRose et al. 2002b), joita käsitellään seuraavissa kohdissa.

4.3.1 XML Path Language (XPath)

XML Path Language (XPath) on kieli, jolla voidaan luoda osoite XML-dokumentin osille. Se tarjoaa myös mahdollisuuden merkkijonojen, numeroiden sekä Boolean-arvojen käsittelyyn. XPath-kielen syntaksi ei ole XML:n mukainen, joten sitä voidaan käyttää URI:ien ja XML-attribuuttien arvoina. XPath saa nimensä sen URI:n kaltaisesta polkunotaatiosta, jossa tiettyyn elementtiin navigoidaan XML-dokumentin hierarkkisen rakenteen läpi. XPath käsittelee XML-dokumenttia puun kaltaisena mallina, joka koostuu solmuista. Puussa on erityyppisiä solmuja, joihin kuuluvat elementtisolmut, attribuuttisolmut ja tekstisolmut. (Clark & DeRose 1999)

Polkulauseke on sarja elementtien nimiä tai ennalta määrättyjä funktioita, jotka erotetaan toisistaan kenoviivalla (/) ja muilla operaattoreilla. Alla olevat esimerkit havainnollistavat operaattoreiden käyttöä lausekkeissa (TAULUKKO 1). XPath-lauseen tulkintaan vaikuttaa se, mitä elementtiä dokumenttia käsittelevä ohjelma on parhaillaan käsittelemässä. Tätä elementtiä kutsutaan *kontekstielementiksi*. Kenoviiva viittaa kaikkiin niihin kontekstielementin lapsielementteihin, jotka ovat kenoviivan oikealla puolella. Kaksoiskenoviiva (//) taas tarkoittaa kaikkia niitä elementtejä, jotka ovat kontekstielementin jälkeläisiä millä tahansa tasolla. Piste viittaa nykyiseen elementtiin ja kaksi pistettä (..) nykyisen elementin vanhempaan. Taksamerkillä (@) puolestaan viitataan siihen kontekstielementin attribuuttiin, jonka nimi on merkin oikealla puolella. Asteriski (*) valitsee kaikki elementin lapsielementit, ja jos se esimerkiksi yhdistetään taksamerkkiin, saadaan valittua kaikki elementin attribuutit. Operaattorit voidaan kirjoittaa kahdessa muodossa, joista pidempää en käsittele tässä raportissa. XPath-esimerkkini onkin kirjoitettu XPathin lyhyessä muodossa.

TAULUKKO 1. Esimerkkejä XPath-lausekkeista ja kielen operaattoreista.

Operaattori	Esimerkki	Selitys
/	/kirja/luku/kappale	Kappale-elementit, jotka ovat luku-elementin lapsia ja kirja-elementin lapsenlapsia.
.	./kappale	Kappale-elementit, jotka ovat nykyisen solmun lapsia.
..	../luku/kappale	Kappale-elementit, joiden vanhempi on luku-elementti, joka puolestaan on nykyisen elementin vanhemman lapsielementti.
//	/kirja//kuva	Kaikki kuva-elementit, jotka ovat kirja-elementin jälkeläisiä millä tahansa tasolla. Kuva-elementin ei siis tarvitse olla kirja-elementin lapsi.
@	/kirja/@nimi	Kirja-elementin nimi-attribuutti.
*	/kirja/@*	Kaikki kirja-elementin attribuutit.

Oikeuksien hallinnan kannalta hyödyllisiä ovat erityisesti XPathin ehtolauseet, joilla voidaan valita tietyn ehdon täyttäviä elementtejä. Ehtojen muodostamisessa voi käyttää niin elementtien kuin niiden attribuuttienkin sisältöä. Ehdot erotetaan muusta polusta hakasuluilla ([ja]). Ehdoista voi myös tehdä Boolean-lauseita yhdistelemällä niitä and- ja or-operaattoreilla. Ehtolauseissa voi käyttää myös funktioita last() ja position(), joilla voi valita kontekstisolmun lapsista sen, joka on halutussa kohdassa. Last()-funktiota käyttämällä voidaan valita solmun viimeinen lapsielementti, ja position()-funktiolla voidaan irrottaa mikä tahansa elementti. Position-funktio voidaan korvata myös sijoittamalla hakasulkeiden sisään pelkkä elementin järjestysnumero. Ehtolauseet voivat sijaita myös polun keskellä, ja rajauksen tuloksista voidaan siten vielä valita vain tietyt elementit. XPath sisältää myös funktioita merkkijonojen ja numeroiden käsittelyyn, mutta en käsittele niitä tässä yhteydessä. Taulukko 2 esittää esimerkkejä ehtolauseiden muodostamisesta ja funktioiden käytöstä.

TAULUKKO 2. Esimerkkejä XPath-kielen ehtolauseista ja funktioista.

Esimerkki	Selitys
/kirja/luku[./nimi="Johdanto"]	Kirja-elementin lapsena oleva luku-elementti, jolla on lapsielementti nimi, jonka arvo on "Johdanto".
/kirja/luku[./@numero="2"]//kuva	Kaikki kuva-elementit, joiden esi-isä on luku-elementti, jonka numero-attribuutin arvo on 2.
/kirja/luku[last()]	Kirja-elementin lapsista viimeinen luku-elementti
/kirja/luku[position()=3]	Kirja-elementin lapsista kolmas luku-elementti.
/kirja/luku[3]/kappale[2]	Kirjan kolmannen luvun toinen kappale.

4.3.2 XML Pointer Language (XPointer)

XML Pointer Language (XPointer) (Grosso et al. 2002b) on kehitteillä oleva kehys, joka toimii perustana XML-resursseihin osoittavien URI-viitteiden katkelmatunnisteille. Sen avulla voidaan luoda URI-viite mille tahansa osalle XML-dokumenttia. Sillä voidaan osoittaa tiettyihin pisteisiin ja alueisiin siinä missä myös kokonaisiin solmuihin.

XPointerilla on kaksi mahdollista muotoa, lyhyt muoto sekä skeemaan perustuva pitkä muoto. Lyhyessä muodossa kirjoitettu XPointer sisältää ainoastaan identifioitavan elementin tunnisteiden erotettuna muusta URI:sta numeromerkillä (#). Pitkässä muodossa kirjoitettu XPointer puolestaan sisältää käytettävän skeeman nimen sekä skeemaspesifin osan. (Grosso et al. 2002b) Mahdollisia skeemoja ovat XPath-kieleen perustuva xpointer-skeema (DeRose et al. 2002b) sekä elementtien tunnistisiin ja järjestysnumeroihin perustuva element-skeema (Grosso et al. 2002a). Lisäksi käytettäviä nimiavaruuksia voidaan määritellä xmlns-skeemalla (DeRose et al. 2002a), mutta tämän skeeman jätän käsittelemättä.

Xpointer-skeema perustuu XPath-kieleen, mutta sitä on laajennettu useilla funktioilla, tyypeillä ja käsitteillä. Sillä voidaan osoittaa tiettyyn elementtiin mutta myös rajata dokumentista osia, jotka koostuvat useista elementeistä. Nämä alueet voivat alkaa mistä tahansa ja loppua mihin tahansa, jopa siten, että alku- ja loppukohdat ovat eri elementtien sisällä. Alue edustaa silloin kaikkia XML-rakenteita ja sisältöä alku- ja loppukohdan välillä. Se voi myös käsittää vain pienen osan jonkin elementin sisällöstä. (DeRose et al. 2002b)

Alue määritellään jollain siihen tarkoitetulla funktiolla. Tämän tutkimuksen aihealueen kannalta pidän niistä käytännöllisimpänä funktiota `range-to()`. Se määrittelee alueen, joka alkaa kontekstisolmusta ja päättyy funktion argumenttina olevaan sijaintiin (DeRose et al. 2002b). Esimerkiksi alla oleva XPointer-lause viittaa alueeseen, joka alkaa luku-elementistä, jonka num-attribuutin arvo on 1 ja päättyy luku-elementtiin, jonka num-attribuutin arvo on 3.

```
#xpointer(//luku[@num="1"]/range-to(//luku[@num="3"]))
```

Element-skeemassa elementtiin viitataan joko sen yksilöllisellä tunnisteella tai navigoimalla elementtien järjestysnumeroiden avulla dokumentin puurakenteessa. Numerot erotetaan toisistaan kenoviivalla (/), ja jokainen numero paikantaa aiemmin paikannetun elementin yhtä monennen lapsielementin. Tapoja voi myös yhdistellä siten, että sarjan ensimmäinen elementti paikannetaan tunnisteiden perusteella. (Grosso et al. 2002a) Alla olevat viitteet voivat siten identifioida saman elementin, jos tunnisteella ”johdanto” identifioitu elementti on juurielementin toinen lapsi.

```
#element(johdanto/3/1)
```

```
#element(/1/2/3/1)
```

XPointerin lyhyttä muotoa voidaan käyttää, jos halutaan viitata yksittäiseen elementtiin, jolla on yksilöllinen tunniste. Silloin tämä tunniste toimii yksin XPointer-lauseena. (Grosso et al. 2002b) Tämä vastaa xpointer-skeeman id-funktion tai element-skeeman käyttöä, joten alla olevat osoitteet ovat toistensa vastineita.

```
http://www.esim.org/doc.xml#xpointer(id("johdanto"))
```

```
http://www.esim.org/doc.xml#element(johdanto)
```

```
http://www.esim.org/doc.xml#johdanto.
```

On kuitenkin huomattava, että edellä kuvatut XPointer-lauseet viittaavat dokumentissa tiettyyn elementtiin vain, mikäli dokumenttia käsittelevä ohjelma tulkitsee jotkin dokumentin elementit tai attribuutit yksilöllisiksi tunnisteiksi ja viitteessä mainittu tunniste on sijoitettu jonkin tällaisen elementin tai attribuutin sisällöksi. Attribuutit ja elementit määritellään tunnisteiksi XML-skeema- tai dokumenttityypimäärittelyissä. Sovelluksilla voi olla myös omia XPointer-kehiksen ulkopuolisia määrittelyjä, joiden mukaan tietyt attribuutit tai elementit tulkitaan aina tunnisteiksi. (Grosso et al. 2002b) Jos mitään näistä ei ole saatavilla, lauseet eivät identifioi yhtään elementtiä. Siinä tapauksessa elementtiin voidaan viitata esimerkiksi XPath-lauseella, jossa valintaehtona on elementin attribuutin arvo. Jos attribuutti on nimeltään esimerkiksi `identifier`, edellisen esimerkin XPointer-lause voidaan kirjoittaa muotoon `xpointer(//*[@identifier="johdanto"])`.

Voisi kuvitella, että XPointer-kielen käyttäminen ratkaisisi myös HTML-dokumenttien osiin viittaamisen ongelman. Kielen käyttö ei kuitenkaan ole sallittu HTML-dokumenttien käsittelyssä. Ensinnäkään HTML-dokumenttien puurakenne ei ole aina yhtä selkeä kuin XML-dokumenttien, sillä HTML-kielessä kaikille elementeille ei vaadita lopputunnistetta (esim. `<p>` ja ``) (Raggett et al. 1999). XPointer-lauseen sisään muotoillut polkulauseet eivät siis välttämättä olisikaan aivan yksiselitteisiä. HTML-dokumenttien rakenteeseen voi myös jäädä runsaastikin virheitä, sillä useat selaimet osaavat näyttää dokumentin halutunlaisena niistä huolimatta. Suurempi rajoite on kuitenkin se, että URI:n katkelmatunnisteen merkitys tulee päätellä dokumentin Internet-mediatyyppin perusteella (Berners-Lee et al. 1998, 15). Jos dokumentin tyyppi on `text/html`, katkelmatunnisteen on sanottu yksiselitteisesti osoittavan dokumentissa elementtiin, joka on identifioitu vastaavalla tunnisteella (Connolly & Masinter 2000, 4). Tämän mukaan siis HTML-muotoisen resurssin URI:n perään liitettyä XPointer-lauseetta ei pitäisi lainkaan tulkita XPointeriksi. Oletan, että jotkin ohjelmistot tekevät tästä kuitenkin poikkeuksen, mutta yleisesti ottaen en voi suositella XPointerin käyttöä HTML-dokumenttien yhteydessä. Siinä riittääköön elementtien identifiointi id-attribuuteilla.

4.4 Extensible Hypertext Markup Language (XHTML)

Edellisissä kohdissa on kerrottu vaikeuksista osoittaa HTML-dokumenttien osiin ja toisaalta siitä, että XML-dokumenteilla tämä on helpompaa. Seuraavissa kohdissa pyrin löytämään vastauksen siihen, kannattaisiko sisällöntuotannossa siirtyäkin HTML:stä XHTML:ään, joka noudattaa XML:n syntaksia

HTML on ollut käyttökelpoinen kieli sisällön tuottamiseen, mutta sen käyttäminen on ollut vaativaa sekä sisällöntuottajille että ohjelmistojen kehittäjille. HTML:ssä on esimerkiksi mahdollista jättää tietyiltä elementeilta lopputunnisteet pois ja myös minimoida määritteitä (Raggett et al. 1999). Lisäksi HTML 4:ään on ajan myötä lisätty runsaasti uusia elementtejä, mikä on johtanut ongelmiin dokumenttien yhteensopivuudessa eri ympäristöjen välillä (Pemberton et al. 2000).

XHTML (eXtensible HyperText Markup Language) on dokumenttityyppien ja moduulien perhe, jossa HTML 4:n dokumenttityypit on muotoiltu uudelleen XML-sovelluksiksi. Perheen ensimmäinen dokumenttityyppi on XHTML 1.0, joka on lähes identtinen HTML 4.01:een verrattuna. Koska dokumentit ovat kuitenkin XML:n syntaksin mukaisia, niitä voidaan käsitellä XML-työkaluilla.

4.4.1 XHTML:n syntaksi

XHTML ei eroa syntaksiltaan paljoakaan HTML:stä. XHTML on kuitenkin XML-sovellus, joten dokumenttien on oltava XML-määritysten mukaisesti hyvin muodostettuja. Tämä tarkoittaa esimerkiksi sitä, että XHTML:ssä on oltava lopputunniste myös sellaisilla elementeillä, joilta sen on saanut HTML:ssä jättää pois. Jokaisen elementin lopputunnisteen on myös oltava saman elementin sisällä kuin sen alkutunnisteen. Lisäksi elementtien ja attribuuttien nimet on sovittu kirjoitettavan pieni kirjaimin. (Pemberton et al. 2000)

XHTML-dokumentilla on Baker & Starkin (2002) mukaan kolme mahdollista tiedostopäätettä: `.xht`, `.xhtml` ja `.html`. Vaikka XHTML-dokumentti onkin XML:n syntaksin mukainen, Baker & Starkin (2002) eivät suosittele tiedostopäätteen `.xml` käyttöä. WWW-palvelimet voivat käsitellä kyseisenlaista dokumenttia siten kuin se olisi mediatyypiltään `text/xml` tai `application/xml`. Tämä taas voi johtaa siihen, että selaimet, jotka eivät osaa käsitellä XML-dokumentteja, käsittelevät sisältöä kuin se olisi tyyppiltään `text/plain` ja näyttävät sisällön merkkauksineen pelkkänä tekstinä (esim. Murata et al. 2001, 5-6).

4.4.2 Katkelmatunnisteet XHTML:ssä

Tarkoitukseni oli selvittää, helpottaisiko XHTML:n käyttäminen sisällön tuottamisessa resurssien rajaamista dokumenteilta. Voisi kuvitella, että koska kieli on XML-sovellus, sen kanssa voisi käyttää esimerkiksi XPointeria. Havaitsin kuitenkin, että näin ei ole. Vaikka XHTML onkin XML-sovellus, sen mediatyyppi ei ole mikään niistä, joiden yhteydessä katkelmatunniste tulkitaan XPointeriksi. Näitä ovat `text/xml`, `application/xml`, `text/xml-external-parsed-entity` tai

`application/xml-external-parsed-entity` (Grosso et al. 2002b). XHTML:n suositeltava mediatyyppi taas on `application/xhtml+xml` (Baker & Stark 2002). Sen yhteydessä voidaan käyttää myös tyyppiä `text/html` (Connolly & Masinter 2000), `application/xml` (Murata et al. 2001, 9–11) ja `text/xml` (Murata et al. 2001, 7–9), mutta se, miten dokumenttia käsitellään riippuu suuresti käyttäjäagentista (Ishikawa 2002).

XHTML on kielenä ja dokumenttityyppinä hyvin nuori, mutta dokumenttien käsittely standardisoituu varmasti ajan myötä. Tällä hetkellä XML-dokumenttityyppien määritykset (Murata et al. 2001) eivät ota huomioon mediatyyppiä `application/xhtml+xml` eivätkä määrittele XML-dokumentteihin osoittavien katkelmatunnisteiden muotoa. Baker & Stark (2002, 4) toteavatkin, että kunnes mediatyyppiä päivitetään, katkelmatunnisteet XHTML-dokumentteihin osoittavissa URI-viitteissä toimivat kuten HTML-dokumenttienkin yhteydessä eli määrittelevät elementin, jolla on vastaava ID-attribuutin arvo.

4.5 Havaintoja

Tässä luvussa tehtyjen havaintojen perusteella en voi suositella, että resursseja rajataan toisten resurssien sisältä. Esimerkiksi edellä käsittelemäni HTML-dokumentin rakenteistaminen `div`- tai `span`-elementeillä ja näihin elementteihin osoittaminen katkelmatunnisteella vaikuttaa mielestäni enemmänkin pakon edessä tehdyttä kuin kestäväältä ratkaisulta. Suosittelenkin, että yksi tiedosto sisältäisi vain yhden resurssin materiaalia ja kaikkea yhden tiedoston sisältämää materiaalia koskisi samat oikeudet.

XML:n käyttö sisällön tuotannossa kuulostaa houkuttevalta ja tuntuu, että se ratkaisisi kaikki ongelmat. En kuitenkaan uskalla suositella siirtymistä sisällöntuotannossa HTML:stä XML:ään. Oletan, että sisällön tuotanto vaikeutuisi ja siihen tarvittaisiin uusia työkaluja sekä uudenlaista osaamista. XML:n käyttö voisi kieltämättä olla käytännöllistä, joten sitä ei kannata hylätä ajatuksena täysin. Luulen kuitenkin, että oikeuksien hallintaa kehitettäessä ei useinkaan ole mahdollista vaikuttaa sisällön tuotannon tapoihin. Ratkaisu olisi siis löydettävä mekanismeista, jotka toimivat, vaikka resursseihin ei voisi lisätä edes yhtä uutta elementtiä.

5 TUNNISTEEN RESOLUUTIO

Tässä luvussa käsittelen sitä, minkälainen yhteys resurssin ja sen tunnisteiden välillä on resursseja hallitsevan järjestelmän kannalta. Aiheen käsittely alkaa *resoluutio*-termin esittelyllä. Resoluution käsittely sisältää myös keskustelua siitä, mitä käyttäjän ruudulle palautetaan tunnisteiden perusteella. Tämän jälkeen siirrytään käsittelemään resurssien identifiointia Open Digital Rights Language -kielessä. Lopuksi käsittelen erilaisia resurssityyppejä, jotka jaan passiivisiin ja aktiivisiin resursseihin.

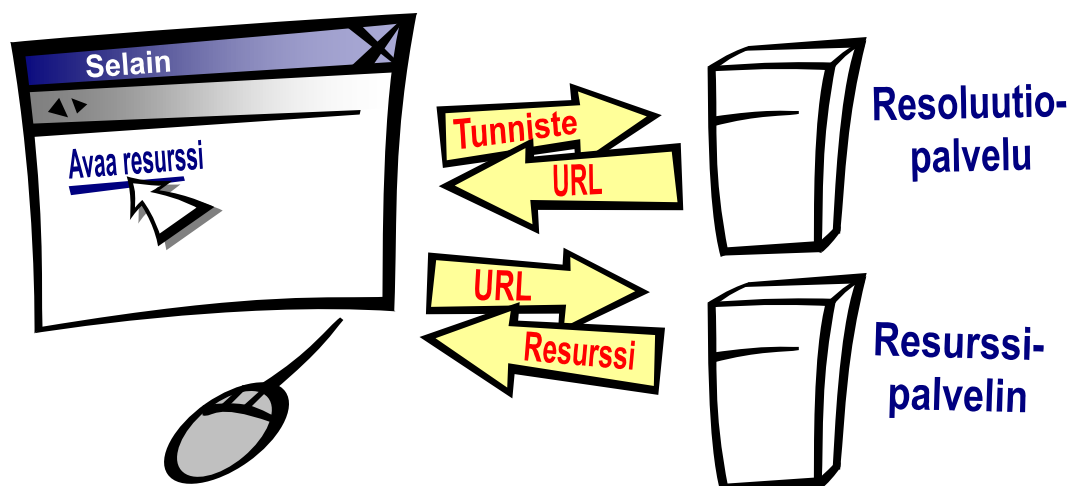
5.1 Resoluutiopalvelut

Resoluutiolla tarkoitetaan tässä yhteydessä prosessia, jossa resurssin tunnisteiden perusteella käyttäjälle palautetaan joko itse resurssi tai joitain sen metatietoja. Resurssia ei ole kaikissa tapauksissa järkevää palauttaa käyttäjälle välittömästi. Esimerkiksi jos resurssin käyttöoikeuksia on rajoitettu, resoluutioprosessi voi palauttaa käyttäjälle resurssin metatietoja sekä kehotuksen kirjautua sisään palveluun omalla käyttäjätunnuksellaan. Tunnisteiden identifioima resurssi voi olla myös erittäin laaja, jolloin resoluution tuloksena voidaan palauttaa jonkinlainen valikko, josta käyttäjä voi valita, haluaako hän nähdä koko resurssin vai jonkin tietyn osan siitä.

Resoluutioprosessia voidaan verrata Internetin nimijärjestelmään (Domain Name System, DNS) (esim. Mockapetris 1987). Siinä palvelimien Internet Protocol -osoitteet (IP-osoitteet) haetaan DNS-palvelimelta palvelimen aliasnimen perusteella. WWW-palvelin etsitään verkosta todellisuudessa IP-osoitteen perusteella. Tunnisteiden resoluutiossa on kysymys täysin vastaavasta prosessista; jonkin resurssin tunnisteen vain vastaa aliasnimeä ja sijainti IP-osoitetta. Sun & Lannom (2002) huomauttavat, että juuri DNS-järjestelmää on ehdotettu laajennettavaksi ratkaisemaan myös resurssien identifiointiongelman. DNS-järjestelmä on kuitenkin Internetin reitityksen perusta, joten sen toimivuutta ei haluta vaarantaa ylimääräisillä laajennuksilla.

Kuvio 10 esittää tunnisteiden resoluutiota yksinkertaisimmillaan. Resoluutioprosessi alkaa siitä, kun halutaan avata jokin resurssi selaimen. WWW-sivulla oleva hyperlinkki ei sisälläkään suoraa linkkiä resurssiin, vaan resurssin tunnisteiden jossain

muodossa. Tulevaisuudessa voisi olla mahdollista, että linkki sisältää halutun resurssin tunniste sellaisenaan, ja selaimeen rakennettu lisäosa pystyy paikantamaan resurssin sen todellisesta sijainnista. Nykyisin todennäköisempää on, että linkki osoittaa esimerkiksi resoluutiopalvelimelle, ja halutun resurssin tunniste vietään palvelimelle URL:n osana. Tähän perustuu esimerkiksi DOI-järjestelmän nykyinen toteutus. Tässä esimerkissä resoluutiopalvelu palauttaa selaimelle resurssin todellisen URL-osoitteen, ja sen jälkeen resurssi noudetaan kyseisestä osoitteesta. Yhtä hyvin URL:n palauttaminen selaimelle voitaisiin jättää välistä ja toimittaa resurssi suoraan käyttäjälle tai toisaalta näyttää esimerkiksi edellä lista resurssin metatiedoista.



KUVIO 10. Yksinkertainen resoluutio.

Verkkoympäristössä voi olla samasta resurssista monta identtistä kopiota. On myös täysin mahdollista, että eri paikoissa sijaitsevilla kopioilla on erilaiset käyttö- ja tekijänoikeudet. Resoluutiopalvelun olisi tarjottava välineet myös tällaisten tilanteiden selvittämiseen. Onkin tärkeää erottaa jonkin tunnisteiden identifioima resurssi niistä tuloksista, jotka jokin resoluutiopalvelu palauttaa. Esimerkiksi ISSN identifioi mahdollisesti satoja tai jopa tuhansia numeroita tietystä lehdestä. Tässä tapauksessa tunnisteiden resoluutio ei voi palauttaa käyttäjälle kaikkia sen identifioimia kohteita, vaan

sen sijaan esimerkiksi jonkinlaisen HTML-pohjaisen lomakkeen, jolla käyttäjä voi navigoida haluamiinsa julkaisuihin tai valita mielenkiintoisia artikkeleita (Lynch et al. 1998, 3). Jopa jokin yksittäinen resurssi, kuten kirja, voi olla niin laaja, että resoluutio suoraan resurssiin ei ole paras vaihtoehto (Lynch et al. 1998, 3).

5.2 ODRL digitaalisessa oikeuksien hallinnassa

ODRL (Open Digital Rights Language) (Iannella 2002) on yksi kielistä, joilla voidaan kuvata organisaation käyttämän materiaalin käyttöoikeuksia digitaalisessa muodossa. Kieli on XML-pohjainen ja avoin, ja se on suunniteltu laajennettavaksi. Syksyllä 2002 ODRL sai W3C:n hyväksynnän, mikä kasvattanee kielen käyttäjäkuntaa.

Kielellä voidaan kuvata monipuolisesti oikeuksien haltijoiden ja hallittavien resurssien suhteita. Kuvaukset eivät rajoitu vain digitaaliseen aineistoon, vaan sillä voidaan kuvata myös fyysiseen materiaaliin kohdistuvia oikeuksia. On kuitenkin selvää, että ODRL:n hyödyt korostuvat erityisesti kun resurssit ovat digitaalisia. (Iannella 2002, 1–2) Kielessä on otettu huomioon IFLA:n mallin (Plassard 1998) abstraktiotasot ja oikeudet voidaan määrittellä esimerkiksi tietylle resurssin ilmentymälle tai niitä voidaan soveltaa kaikkiin resurssin ilmentymiin (Iannella 2002, 2).

ODRL on riippuvainen resurssien ja osapuolten yksilöllisestä tunnistamisesta. Yleinen tunnistus on kuitenkin erittäin vaikea ongelma, sillä se vaatii osapuolten välisiä sopimuksia. Siksi identifiointimekanismit ja -käytännöt on jätetty ODRL:n ulkopuolelle. (Iannella 2002, 2)

5.2.1 ODRL:n oikeuksien hallinnan malli

ODRL:n oikeuksien hallinnan malli koostuu kolmesta perusentiteetistä: *resursseista* (*assets*), *oikeuksista* ja *osapuolista* (*parties*). Resurssit voivat olla missä tahansa muodossa ja koostua useistakin osista. Oikeudet pitävät sisällään lupia (permissions), jotka voivat sisältää rajoitteita (constraints), vaatimuksia (requirements) ja ehtoja (conditions). Luvat ovat johonkin resurssiin kohdistuvaa käyttöä, joka käyttäjälle on sallittu, esimerkiksi *toista video*. Rajoitteet asettavat toiminnalle rajoja, esimerkiksi

toista video enintään viisi kertaa. Vaatimukset ovat velvoitteita, joita tarvitaan luvan harjoittamiseen, esimerkiksi maksa 5 € joka kerta kun toistat videon. Ehtolauseet taas määrittelevät poikkeukset, joiden toteutuessa luvat raukeavat. (Iannella 2002, 4–5)

Osapuoliin kuuluvat loppukäyttäjät ja oikeuksien haltijat. Ne voivat olla henkilöitä, organisaatioita tai tiettyjä rooleja. Loppukäyttäjät ovat yleensä resurssien kuluttajia. Oikeuksien haltijat ovat yleensä osapuolia, joilla on ollut jokin rooli resurssin luomisessa, tuotannossa tai jakelussa ja joilla voi olla jonkinlainen omistajuussuhde resurssiin tai sen lupiin. (Iannella 2002, 5)

Näiden kolmen perusentiteetin avulla voidaan ilmaista tarjouksia (offer) ja sopimuksia (agreement). Tarjoukset ovat oikeuksien haltijoiden ehdotuksia tietyille oikeuksille resurssiin nähden. Sopimus syntyy, kun osapuolet hyväksyvät tarjouksen. (Iannella 2002, 5)

5.2.2 ODRL:n kontekstimalli

Resurssit identifioidaan ODRL-kielessä käyttäen erityistä konteksti-elementtiä (context). Kyseistä elementtiä voidaan käyttää useassa eri tarkoituksessa, ja se voi liittyä mihin tahansa toiseen elementtiin. Jos kontekstielementti esimerkiksi liitetään resurssiin (asset-elementti), sillä voidaan ilmaista resurssin yksilöllinen tunniste. Osapuoliin (party-elementti) liittyvä kontekstielementti puolestaan ilmaisee esimerkiksi osapuolen roolin tai nimen. Kontekstielementti voi liittyä myös kokonaiseen oikeuslauseeseen, kuten tarjoukseen, jolloin se ilmaisee esimerkiksi tarjouksen yksilöllisen tunnisteen sekä sen, milloin tarjous on tehty. (Iannella 2002, 19)

ODRL:n kontekstimalli koostuu kokonaisuudessaan kymmenestä elementistä (Iannella 2002, 19), mutta resurssien identifioinnin ja resoluution kannalta niistä olennaisimmat ovat resurssin yksilöllinen tunniste (uid-elementti) ja digitaalinen sijainti (dLocation-elementti). ODRL:n uusimman version (Iannella 2002) mukaan molempien elementtien sisältönä on käytettävä URI:n syntaksin mukaista merkkijonoa. ODRL:n aiemmissa versioissa uid-elementti sai kuitenkin sisältää myös muiden identifiointijärjestelmien mukaan muodostettuja tunnisteita, joten yhteensopivuuden vuoksi ne sallitaan edelleen.

Kontekstielementti voi sisältää myös useita uid-elementtejä, jos resurssi koostuu useista osista ja jokaiseen halutaan viitata omalla tunnisteellaan. Näin kokoelmaa voidaan käsitellä yhtenä resurssina. (Iannella 2002, 20)

Kuviossa 11 on esimerkki ODRL:n mukaisesta sopimuksesta (agreement-elementti). Sopimuksen ensimmäinen elementti on context, jolla määritellään tässä tapauksessa sopimuksen tunniste. Sopimuksen toinen elementti, asset, määrittelee sopimuksen kohteena olevan resurssin. Resurssin tunniste ja sijainti määritellään edellä kuvatun kaltaisesti uid- ja dLocation-elementeillä. Esimerkin sopimuksessa resurssin määrittelyn jälkeen kuvataan siihen liittyvät luvat (permission-elementti) ja osapuolet (party-elementti), joiden sisältöä en käy tässä tarkemmin läpi.

```
<agreement>
  <context>
    <uid>doi:10.9876/123</uid>
  </context>
  <asset>
    <context>
      <uid>urn:isbn:951-1-16671-9</uid>
      <dLocation>http://esim.org/951-1-16671-9</dLocation>
    </context>
  </asset>
  <permission>
    ...
  </permission>
  <party>
    <context>
      ...
    </context>
  </party>
</agreement>
```

KUVIO 11. ODRL:n kontekstimalli.

ODRL:ää käyttävä oikeuksienhallintajärjestelmä voi myös toimia itsessään resoluutiopalveluna. Oikeusmäärityksethän voivat sisältää sekä resurssin tunnisteen että sen digitaalisen sijainnin, joten tarvetta ulkopuolisen resoluutiopalvelun käyttöön ei välttämättä ole.

5.3 Resurssien identifiointi ODRL:ssä

Tämän tutkimuksen tarkoituksena oli selvittää, miten oikeuksien hallinnan kohteet tulisi identifioida ODRL-kielessä. Lisäksi tehtäväni oli selvittää, tarvitseeko kieltä laajentaa hyväksymään URI:n lisäksi myös muita identifiointimekanismeja.

Edellisissä luvuissa olen käsitellyt eri identifiointimekanismeja sekä resurssien paikantamista. Identifiointijärjestelmiä on olemassa useita, ja kaikkien mahdollisten järjestelmien käsittely on lähes mahdotonta. En myöskään pidä sitä tarpeellisena. Tutkimuksen edetessä vakuutuin siitä, että URI on identifiointiskeemoista yleiskäyttöisin. Monet uudet tunnistejärjestelmät suunnitellaan alusta asti URI-yhteensopiviksi, ja vanhoille tunnistejärjestelmille rekisteröidään URN-nimiavaruuksia, jolloin nekin voidaan kirjoittaa URI:n syntaksin mukaisesti. Myös resurssien paikantaminen on tehtävä URI:n avulla. Riippumatta siitä, käytetäänkö pelkkää URL-osoitetta vai osoitetaanko XML-resurssiin XPointerilla, resurssin osoite on URI:n syntaksin mukainen.

Näin ollen voin sanoa, että havaintojeni mukaan ODRL:n identifiointimalli on tällaisenaan riittävä, eikä sitä tarvitse laajentaa muilla skeemoilla. Se, miten identifiointiin ja paikantamiseen käytettävät URI:t muodostetaan käytännössä, on kysymys, johon tämä tutkimus ei kuitenkaan anna vastausta. Vastauksen antamiseen tarvittaisiin tietoja muusta käytettävästä teknologiasta.

5.4 Passiivisista resursseista aktiivisiin

Resurssit voivat olla järjestelmissä myös paljon monimutkaisempia kokonaisuuksia kuin vain staattisia tiedostoja, joiden sisältö luetaan ja näytetään käyttäjälle sellaisenaan. Tällaisista *passiivisista resursseista* poiketen monet järjestelmät tuottavat sisältöä dynaamisesti erilaisten palvelimella toimivien ohjelmien avulla. Kutsun tällaisia resursseja *aktiivisiksi*. Aktiivisia resursseja ovat esimerkiksi Nelsonin esittelemät kirjastojärjestelmien tarpeisiin kehitetyt bucket-oliot (Nelson 2000). Sana 'bucket' merkitsee sankoa ja viittaa siihen, että olio on kuin säilytysastia, johon on kerätty kaikki

tieto, jota tarvitaan sen itsenäiseen toimintaan ja siihen, että se osaa näyttää oikeille henkilöille oikean sisällön.

Nelson on havainnut, että digitaalisia kirjastoja kehitettäessä unohdetaan usein, että tietosisällön tulisi olla tärkeämpi kuin sitä käsittelevien järjestelmien. Keskusteluissa keskitytään toteutuksen yksityiskohtiin, kuten tiettyjen tietokantaratkaisujen toimintaan tai siihen, mitä järjestelmiä käytetään. Tämä johtuu siitä, että kun valitaan tietty tapa toteuttaa esimerkiksi juuri digitaalinen kirjasto, sen tarjoamat palvelut on usein kytketty tiukasti sisältöön, jotta sisällön oikeuksia voidaan hallita. Tämä kuitenkin rajoittaa tulevaisuuden kirjastojärjestelmiin siirtymistä, eikä mahdollista sitä, että useat järjestelmät voivat käyttää samoja tietoresursseja. (Nelson 2000, 1)

Nelsonin aktiiviset resurssit ovat älykkäitä ja oliopohjaisia rakenteita, joita voidaan käyttää tiedon julkaisuun digitaalisissa kirjastoissa. Olioiden sisältöihin on mahdollista päästä käsiksi useista järjestelmistä, ja niitä voidaan myös siirtää helposti toisiin järjestelmiin. Resurssit voidaan toteuttaa monin tavoin. Nelson esimerkiksi käytti omassa toteutuksessaan CGI-ohjelmia tukevaa http-palvelinta, ja ohjelmointikielenä Perliä. (Nelson 2000, 4)

Nelsonin mukaan resurssien tulisi olla itsenäisiä ja aktiivisia, ja niiden tulisi kyetä vastaamaan toimintoihin itsenäisesti. Tämä tarkoittaa sitä, että resurssi itsessään sisältäisi kaiken tarvittavan tiedon sopivan dokumenttinäkymän muodostamiseen, kun tietty käyttäjä haluaa nähdä resurssin. (Nelson 2000, 16—17) Tällöin käyttäjän tai oikeuksienhallintajärjestelmän ei tarvitse tietää resurssista muuta kuin URL-osoite, jossa resurssi on saatavilla. Kaikki käytössä olevat menetit resurssin tietojen käsittelyyn on automaattisesti sisällytetty olion tuottamaan HTML-tulosteeseen. (Nelson 2000, 27)

Resoluution käsittelyn yhteydessä esitin, että järjestelmä ei aina veisikään käyttäjää annetun tunnisteiden perusteella suoraan resurssiin. Sen sijaan hänet ohjattaisiin johonkin navigointirakenteeseen tai resurssin metatietoihin, joista on mahdollista edetä todellisen sisällön tarkasteluun. Aktiivisen resurssin yhteydessä tämänkaltainen toiminta on mahdollista toteuttaa resurssiin itseensä, jolloin oikeuksia hallitsevan järjestelmän osaksi jää mahdollisesti vain resoluutio oikeaan URL-osoitteeseen.

6 YHTEENVETO

Tässä työssä tutkittiin resurssien identifioinnin tärkeimpiä kysymyksiä ja pohdittiin, mitkä asiat vaikuttavat käytettävän identifiointijärjestelmän valintaan. Tutkimuksen lähtökohta oli digitaalinen oikeuksien hallinta, jonka yksi keskeisistä osa-alue on resurssien identifiointi.

Tutkimuksessa todettiin, että resursseja voidaan identifioida erilaisilla tasoilla ja erilaisella tarkkuudella. Tästä käytetään nimitystä toiminnallinen granulariteetti. Identifioinnin tarkkuudessa on kaksi ulottuvuutta. Toisaalta voidaan erottaa resurssien eri abstraktiotasot ja päättää, halutaanko nämä tasot identifioida omilla tunnisteillaan. Toisaalta taas voidaan erotella resursseista pienempiä tunnistettavia osia kokoomateoksesta aina pieniin dokumenttikomponentteihin saakka.

Tunnisteiden antamisen ja erityisesti identifiointijärjestelmän valinnan todettiin olevan haastava tehtävä. Tunnisteen tärkeimmän ominaisuuden katsottiin olevan pysyvyys, jonka todettiin onnistuvan parhaiten, jos tunnisteessa ei ole minkäänlaista älykkyyttä eli tunnistetta tarkastelemalla ei saada tietoa tunnistettavasta resurssista, sen sijainnista eikä omistussuhteista. Tällaisten niin sanottujen tyhmiä tunnisteiden käyttö kuitenkin vaatii jonkinlaista resoluutiopalvelua, jossa tunnisteen perusteella haetaan muun muassa tunnistettavan resurssin tai sen metatietojen sijaintitiedot, joiden perusteella resurssi voidaan noutaa.

Käsittelin tutkimuksessani useita identifiointijärjestelmiä ja etsin niistä sopivinta, jota voisi käyttää oikeuksien hallintajärjestelmässä resurssien identifiointiin. Yksikään identifiointijärjestelmä ei noussut ylitse muiden, mutta käyttökelpoisimmilta vaikuttivat identifiointijärjestelmät, joiden tuottama tunniste voidaan kirjoittaa myös URI:n syntaksin mukaiseksi.

Resurssien identifioinnin lisäksi käsittelin sitä, miten resurssin sisältä voidaan rajata pienempiä resursseja sekä miten näille resurssin osille voidaan luoda oma URI-viitteensä. Totesin, että XML-dokumenteilla osien irrottaminen kokonaisuudesta on yksinkertaista esimerkiksi URI:n perään liitettävän XPointer-lauseen avulla. HTML-

dokumenttien yhteydessä pidin mahdollisena ratkaisuna osittaista rakenteistamista ylimääräisten div- tai span-elementtien avulla ja näihin elementteihin viittaamista URI:n katkelmatunnisteella. Suosittelin kuitenkin sitä, että yhtä HTML-tiedostoa pidettäisiin yhtenä resurssina, jolloin koko tiedostoa koskisivat samat oikeudet. Tutkin myös, olisiko resurssien rajaamisessa hyötyä siitä, että HTML:n asemesta sisällön tuotannossa käytettäisiin XHTML:ää. Havaitsin kuitenkin, että nykyisten spesifikaatioiden mukaan XHTML:ää käsitellään pääosin kuin HTML:ääkin, joten esimerkiksi XPointerin käyttö XHTML-dokumenttien yhteydessä ei ole sallittua.

Viimeisessä luvussa ennen yhteenvetoa käsittelin tunnisteiden resoluutiota sekä resurssien identifiointia Open Digital Rights Language -kielessä. Totesin, että URI on riittävä sekä kohteiden identifiointiin että niiden paikantamiseen, joten kieltä ei tarvitse laajentaa näiden asioiden suhteen. Huomautin myös, että ODRL:ää käyttävä oikeuksienhallintajärjestelmä voi toimia itsessään resoluutiopalveluna, joten ulkopuolista resoluutiopalvelua ei tarvita. Lisäksi esitin vaihtoehdoisen resurssien toteutustavan. Tässä tavassa resurssit ovat itsenäisesti toimivia ja aktiivisia olioita, jotka sisältävät kaiken tarvittavan koodin dokumenttinäkymien muodostamiseen. Tällaisessa järjestelmässä oikeuksien hallintajärjestelmä siirtäisi käyttäjän suoraan resurssiin ja resurssi pitäisi itse huolen siitä, että käyttäjä näkee oikeat tiedot.

Tutkimus herätti vähintään yhtä paljon kysymyksiä kuin mihin se antoi vastauksia. Jatkossa olisikin hyvä tutkia esimerkiksi sitä, voisiko sisällön tuotannossa käyttää XML:ää HTML:n asemesta ja mitä vaatimuksia se asettaisi niin järjestelmälle kuin sisältöä tuottaville henkilöille. Jos taas halutaan käyttää HTML:ää ja silti rajata osia resurssien sisältä, pitää selvittää, toimisiko esittelemäni katkelmatunnisteisiin perustuva ratkaisu käytännössä tai olisiko ongelmaan olemassa jokin parempi ratkaisu.

Myös käytettävän identifiointijärjestelmän valinta jäi tässä tutkimuksessa ratkaisematta. Jatkotutkimuksissa olisi tiedettävä toteutettavasta järjestelmästä, käytettävästä teknologiasta sekä identifioitavista resursseista ja niiden granulariteetista enemmän, jotta kysymykseen voitaisiin löytää vastaus.

Suurin kysymys on tietysti se, miten koko oikeuksien hallintajärjestelmä tulisi toteuttaa. Kuinka aktiivinen toimija oikeuksia hallitseva järjestelmä on ja hallitseeko se oikeuksien lisäksi myös itse resursseja? Entä olisiko Nelsonin sankojen kaltaisista aktiivisista resursseista käytännön hyötyä vai tekisikö niiden käyttö järjestelmästä entistä monimutkaisemman?

Tutkimuksen aihealue on suhteellisen nuori ja osa tutkimistani standardeista ja tekniikoista hyvin tuoreita. Toiset niistä saattavat vanhentua nopeasti kun taas toiset vakiinnuttavat asemansa ja kestävät käytössä pitkään. Jotkin omasta mielestäni ongelmalliset asiat saattavat ratketa hyvinkin nopeasti, kun uusia tekniikoita kehitetään. Toisten kanssa luultavasti painitaan vielä hyvinkin pitkään.

LÄHTEET

Adler S., Berglund A., Caruso J., Deach S., Graham T., Grosso P., Gutentag E., Milowski A., Parnell S., Richman J. & Zilles S., 2001. Extensible Stylesheet Language (XSL) Version 1.0 [online]. W3C Recommendation 15 October 2001 [viitattu 7.6.2002]. Saatavilla WWW-muodossa <URL: <http://www.w3.org/TR/2001/REC-xsl-20011015/>>.

Baker M. & Stark P., 2002. The 'application/xhtml+xml' Media Type [online]. Internet Engineering Task Force, Network Working Group, January 2002 [viitattu 12.6.2002]. Saatavilla WWW-muodossa <URL: <http://www.ietf.org/rfc/rfc3236.txt>>.

Berners-Lee T., Fielding R., Irvine U. C. & Masinter L., 1998. Uniform Resource Identifiers (URI): Generic Syntax [online]. Network Working Group, Request for Comments: 2396, August 1998 [viitattu 18.2.2002]. Saatavilla tekstimuodossa <URL: <http://www.ietf.org/rfc/rfc2396.txt>>.

Berners-Lee T., Masinter L. & McCahill M., 1994. Uniform Resource Locators (URL) [online]. Network Working Group, Request for Comments: 1738, December 1994 [viitattu 15.7.2002]. Saatavilla WWW-muodossa <URL: <http://www.ietf.org/rfc/rfc1738.txt>>.

Bray T., Paoli J., Sperberg-McQueen C. M. & Maler E. 2000. Extensible Markup Language (XML) 1.0 (Second Edition) [online]. W3C Recommendation 6 October 2000 [Viitattu 6.6.2002]. Saatavilla WWW-muodossa <URL: <http://www.w3.org/TR/2000/REC-xml-20001006/>>.

Cattell R.G.G., 1991. Object data management: object-oriented and extended relational database systems. Reading, Massachusetts: Addison-Wesley. ISBN 0-201-53092-9.

Clark J. & DeRose S., 1999. XML Path Language (XPath) Version 1.0 [online]. W3C Recommendation 16 November 1999 [Viitattu 5.6.2002]. Saatavilla WWW-muodossa <URL: <http://www.w3.org/TR/1999/REC-xpath-19991116/>>

Connolly D. & Masinter L., 2000. The 'text/html' Media Type [online]. Network Working Group, Request for Comments: 2854, June 2000 [viitattu 5.6.2002]. Saatavilla WWW-muodossa <URL: <http://www.ietf.org/rfc/rfc2854.txt>>.

Daigle L., van Gulik D., Iannella R. & Faltstrom P, 1999. URN Namespace Definition Mechanisms [online]. Network Working Group, Request for Comments: 2611, June 1999 [viitattu 16.7.2002]. Saatavilla WWW-muodossa <URL: <http://www.ietf.org/rfc/rfc2611.txt>>.

DeRose S., Daniel R. Jr., Maler E. & Marsh J., 2002a. XPointer xmlns() Scheme [online]. W3C Proposed Recommendation 13 November 2002 [viitattu 17.12.2002]. Saatavilla WWW-muodossa <URL: <http://www.w3.org/TR/2002/PR-xptr-xmlns-20021113/>>.

DeRose S., Maler E. & Daniel R. Jr., 2002b. XPointer xpointer() Scheme [online]. W3C Working Draft 10 July 2002 [viitattu 17.12.2002]. Saatavilla WWW-muodossa <URL: <http://www.w3.org/TR/2002/WD-xptr-xpointer-20020710/>>.

Finseth C. A., 2002. The ISAN / V-ISAN Handbook, Draft [online]. ISO/TC 46/SC 9/WG 1 N 196. Date: 2002-09-09, [viitattu 15.9.2002]. St Paul, Minnesota: Firwood Consulting, Inc. Saatavilla Adobe Acrobat -muodossa <URL: <http://www.nlc-bnc.ca/iso/tc46sc9/wg1/wg1n196.pdf>>.

Freed N. & Borenstein N., 1996a. Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies [online]. Network Working Group. Request for Comments: 2045. November 1996 [viitattu 10.6.2002]. Saatavilla WWW-muodossa <URL: <http://www.ietf.org/rfc/rfc2045.txt>>.

Freed N. & Borenstein N., 1996b. Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types [online]. Network Working Group. Request for Comments: 2046. November 1996 [viitattu 10.6.2002]. Saatavilla WWW-muodossa <URL: <http://www.ietf.org/rfc/rfc2046.txt>>.

Goldfarb C. F., 1990. The SGML Handbook. Edited and with a foreword by Yuri Rubinsky. Oxford: Oxford University Press. ISBN 0-19-853737-9.

Grass J. E. & Arms W. Y., 1994. A Syntax for Distributed Object Handles [online]. August 3 1994 [viitattu 5.4.2002]. Saatavilla WWW-muodossa <URL: http://www.ifla.org/documents/libraries/cataloging/metadata/hdl_form.txt>.

Green B. & Bide M., 1997. Unique Identifiers: a brief introduction [online]. Book Industry Communication. [viitattu 4.4.2002]. Saatavilla WWW-muodossa <URL: <http://www.bic.org.uk/uniqid>>. ISBN 1-873671-18-0.

Grosso P., Maler., E., Marsh J. & Walsh N., 2002a. XPointer element() Scheme [online]. W3C Proposed Recommendation 13 November 2002 [viitattu 17.12.2002]. Saatavilla WWW-muodossa <URL: <http://www.w3.org/TR/2002/PR-xptr-element-20021113/>>.

Grosso P., Maler E., Marsh J. & Walsh N., 2002b. XPointer Framework [online]. W3C Proposed Recommendation 13 November 2002 [Viitattu 17.12.2002]. Saatavilla WWW-muodossa <URL: <http://www.w3.org/TR/2002/PR-xptr-framework-20021113/>>.

Hakala, J. & Walravens, H., 2001. Using International Standard Book Numbers as Uniform Resource Names [online]. Network Working Group, Request for Comments: 3187, October 2001 [viitattu 16.7.2002]. Saatavilla WWW-muodossa <URL: <http://www.ietf.org/rfc/rfc3187.txt>>.

Hakala, J., 2001. Using National Bibliography Numbers as Uniform Resource Names [online]. Network Working Group, Request for Comments: 3188, October 2001 [viitattu 16.7.2002]. Saatavilla WWW-muodossa <URL: <http://www.ietf.org/rfc/rfc3188.txt>>.

Hakala, J., 2002. Elektronisten julkaisujen identifointi [online]. Versio 3.0, 12.6.2002 [viitattu 15.7.2002]. Saatavilla WWW-muodossa <URL: <http://www.lib.helsinki.fi/meta/id.html>>. URN:NBN:fi-fe20021259.

Iannella R., 2001. Digital Rights management (DRM) Architectures [online]. Corporation for National Research Initiatives, June 2001 [viitattu 18.2.2002]. D-Lib Magazine June 2001, Vol. 7 No. 6. Saatavilla WWW-muodossa <URL: <http://www.dlib.org/dlib/june01/iannella/06iannella.html>>. ISSN 1082-9873.

Iannella R., 2002. Open Digital Rights Language (ODRL) Version 1.1 [online]. IPR Systems Pty Ltd, 8.8.2002 [viitattu 14.9.2002]. Saatavilla Adobe Acrobat -muodossa <URL: <http://odrl.net/1.1/ODRL-11.pdf>>.

IFPI Secretariat, 2002. International Recording Code (ISRC) Handbook: Incorporating the ISRC Practical Guide. 1st Edition, January 2002. Lontoo: International ISRC Agency (IFPI Secretariat). Saatavilla Adobe Acrobat -muodossa <URL: http://www.ifpi.org/site-content/library/isrc_handbook_2002.pdf>.

International ISBN Agency, 2001. The International Standard Book Number System: ISBN Users' Manual: International edition. Fourth, revised edition. Berliini: International ISBN Agency. ISBN 3-88053-075-0. Saatavilla Adobe Acrobat -muodossa <URL: <http://www.isbn-international.org/download/ISBNmanual.pdf>>.

Ishikawa M., 2002. XHTML Media Types [online]. W3C Note 30 April 2002 [Viitattu 5.6.2002]. Saatavilla WWW-muodossa <URL: <http://www.w3.org/TR/2002/NOTE-xhtml-media-types-20020430/xhtml-media-types.html>>.

ISO 2108:1992. Information and documentation – International standard book numbering (ISBN). Third Edition, 1992-05-15. Geneve: International Organization for Standardization.

ISO 3297:1998. Information and documentation – International standard serial number (ISSN). Third Edition, 1998-11-01. Geneve: International Organization for Standardization.

ISO 3901:2001. Information and documentation – International Standard Recording Code (ISRC). Second Edition, 2001-10-01. Geneve: International Organization for Standardization.

ISO 10444:1994. Information and documentation – International standard technical report number (ISRN). First Edition, 1994-12-15. Geneve: International Organization for Standardization.

ISO 10957:1993. Information and documentation – International standard music number (ISMN). First Edition, 1993-12-01. Geneve: International Organization for Standardization.

ISO 15707:2001. Information and documentation – International Standard Musical Work Code (ISWC). First Edition, 2001-11-15. Geneve: International Organization for Standardization.

ISSN International Centre, 2002. The ISSN [online]. Pariisi: ISSN International Centre [viitattu 10.7.2002]. Saatavilla WWW-muodossa <URL: <http://www.issn.org:8080/English/pub/faqs/issn>>

ISTC Working Group, 2002. International Standard Text Code (ISTC): Frequently Asked Questions [online]. International Organization for Standardization [viitattu 10.7.2002]. Saatavilla WWW-muodossa <URL: <http://www.nlc-bnc.ca/iso/tc46sc9/istc.htm>>.

Khoshafian S., & Copeland G. P., 1986. Object Identity. Teoksessa N. Meyrowitz (toim.) OOPSLA '86: Object-Oriented Programming Systems, Languages and Applications, conference proceedings, September 29 – October 2, 1986, Portland, Oregon. New York: ACM Press, 406–416.

Lotus Development Corporation, 2000. Lotus C API 5.0.3 User Guide [online-tietokanta]. Lotus Development Corporation, päivitetty 9.6.2000 [viitattu 26.6.2002].

Chapter 14-1, Anatomy of a Note ID. Saatavilla WWW-muodossa <URL: <http://doc.notes.net/tools/c/5.0.3/api503ug.nsf>>.

Lynch C., 1997. Identifiers and Their Role In Networked Information Applications [online]. ARL: A Biomonthly Newsletter of Research Library Issues and Actions 194. October 1997 [viitattu 4.4.2002]. Washington, DC: Association of Research Libraries. Saatavilla WWW-muodossa <URL: <http://www.arl.org/newsltr/194/identifier.html>>.

Lynch C., Preston C. & Daniel R., 1998. Using Existing Bibliographic Identifiers as Uniform Resource Names [online]. Network Working Group, Request for Comments: 2288, February 1998 [viitattu 15.7.2002]. Saatavilla WWW-muodossa <URL: <http://www.ietf.org/rfc/rfc2288.txt>>.

Moats R., 1997. URN Syntax [online]. Network Working Group, Request for Comments: 2141, May 1997 [viitattu 14.6.2002]. Saatavilla WWW-muodossa <URL: <http://www.ietf.org/rfc/rfc2141.txt>>.

Mockapetris P., 1987. Domain Names – Concepts And Facilities [online]. Request for Comments 1034. November 1987 [viitattu 21.4.2002]. Saatavilla WWW-muodossa <URL: <http://www.ietf.org/rfc/rfc1034.txt>>.

Murata M., St.Laurent S. & Kohn D., 2001. XML Media Types [online]. Internet Engineering Task Force, Network Working Group, Request for Comments: 3023, January 2001 [viitattu 12.6.2002]. Saatavilla WWW-muodossa <URL: <http://www.ietf.org/rfc/rfc3023.txt>>

Nelson M. L., 2000. Buckets: Smart Objects for Digital Libraries. Norfolk, Virginia: Old Dominion University, väitöskirja. Saatavilla WWW:ssä osoitteessa <URL: <http://mln.larc.nasa.gov/~mln/phd/>>

NISO Press, 1996. Serial Item and Contribution Identifier (SICI). ANSI/NISO Z39.56-1996 (Version 2). Bethesda, Maryland: National Information Standards Organization. Saatavilla Adobe Acrobat -muodossa <URL: <http://sunsite.berkeley.edu/SICI/sici.pdf>>.

NISO Press, 2000. Syntax for the Digital Object Identifier. ANSI/NISO Z39.84-2000. Bethesda, Maryland: National Information Standards Organization. Saatavilla Adobe Acrobat -muodossa <URL: <http://www.niso.org/standards/resources/Z39-84-2000.pdf>>.

Paskin N., 1997. Information Identifiers [online]. Originally published in Learned Publishing, Vol 10 No. 2, pp 135-136 (April 1997) [viitattu 4.4.2002]. Saatavilla WWW-muodossa <URL: <http://www.elsevier.nl/inca/homepage/about/infoident/>>.

Paskin N., 2001. The DOI Handbook [online]. International DOI Foundation, February 2001 [viitattu 18.2.2002]. Saatavilla Adobe Acrobat -muodossa: <URL: <http://dx.doi.org/10.1000/186>>. DOI: 10.1000/186.

Payette S., 1998. Persistent Identifiers on the Digital Terrain [online]. RLG DigiNews, April 15, 1998, Volume 2, Number 2 [viitattu 15.7.2002]. Saatavilla WWW-muodossa <URL: <http://www.rlg.org/preserv/diginews/diginews22.html>>. ISSN 1093-5371.

Pemberton S., Altheim M., Austin D., Boumphrey F. et al., 2000. XHTML™ 1.0: The Extensible HyperText Markup Language. A Reformulation of HTML 4 in XML 1.0 [online]. W3C Recommendation 26 January 2000 [viitattu 6.6.2002]. Saatavilla WWW-muodossa <URL: <http://www.w3.org/TR/2000/REC-xhtml1-20000126/>>

Plassard M.-F., 1998. Functional Requirements for Bibliographic Records Final Report. IFLA Study Group on the Functional Requirements for Bibliographic Records. München: Saur. UBCIM publications New Series Vol. 19. Saatavilla myös elektronisessa muodossa: <URL: <http://www.ifla.org/VII/s13/frbr/frbr.pdf>> [viitattu 18.2.2002]. ISBN 3-598-11382-X.

Päivärinta T., Tyrväinen P. & Ylimäki T., 2002. Defining Organizational Document Metadata: A case beyond standards. To appear in Proceedings of the Xth European Conference on Information Systems (ECIS), Gdansk, Poland, June 6.-8. 2002.

Raggett D., Le Hors A. & Jacobs I., 1999. HTML 4.01 Specification [online]. W3C Recommendation 24 December 1999 [Viitattu 5.6.2002]. Saatavilla WWW-muodossa <URL: <http://www.w3.org/TR/1999/REC-html401-19991224/>>

Rivest R., 1992. The MD5 Message-Digest Algorithm [online]. Network Working Group, Request for Comments: 1321, April 1992. Saatavilla WWW-muodossa <URL: <http://www.ietf.org/rfc/rfc1321.txt>>.

Rozenfeld S., 2001. Using The ISSN (International Serial Standard Number) as URN (Uniform Resource Names) within an ISSN-URN Namespace [online]. Network Working Group, Request for Comments: 3044, January 2001 [viitattu 15.7.2002]. Saatavilla WWW-muodossa <URL: <http://www.ietf.org/rfc/rfc3044.txt>>.

Rust G. & Bide R., 2000. <indecs> metadata framework: principles, model and dictionary [online]. Indecs Framework Ltd, June 2000 [viitattu 18.2.2002]. Saatavilla Adobe Acrobat -muodossa: <URL: <http://www.indecs.org/pdf/framework.pdf>>.

Sollins K. & Masinter L., 1994. Functional Requirements for Uniform Resource Names [online]. Network Working Group, Request for Comments: 1737. December 1994 [viitattu 5.4.2002]. Saatavilla WWW-muodossa <URL: <http://www.ifla.org/documents/libraries/cataloging/metadata/rfc1737.txt>>.

Sun S. X. & Lannom L., 2002. Handle System Overview [online]. February 2002, Updated 4 Mar 02 [viitattu 4.4.2002]. Saatavilla WWW-muodossa <URL: <http://www.handle.net/overview-current.html>>.

The Open Group, 1997. Universal Unique Identifier [online]. The Open Group CAE Specification, The Open Group DCE 1.1: Remote Procedure Call, Document number: C706 [viitattu 27.6.2002]. Saatavilla WWW-muodossa <URL: <http://www.opengroup.org/onlinepubs/9629399/apdxa.htm>>.