Hajautettu dokumenttien hallinta

Johdatus tekstin ja dokumenttien käsittelyyn tietoverkoissa

Timo Kuronen*

15.11.1997

* Email: Timo.Kuronen@gsf.fi

ISBN 951-42-4759-0

Oulun yliopiston kirjasto
Oulu 1997

Alkusanat

Tietoverkkojen kehittyminen on lyhyessä ajassa suuresti muuttanut vakiintuneita atk:n käyttötapoja. Päätekäyttöiset keskuskoneet ovat jääneet lähes kokonaan pois käytöstä. Tilalle ovat tulleet mikrot, työasemat, palvelimet ja tietoliikenneverkot.

Tietoverkot ovat moninaisuudessaan vaikeita ja hämmentäviä sekä maallikoille että ammattilaisille. Monet verkkojen käyttöön liittyvistä käsitteistä ovat uusia ja hämäriä. Niiden takana saattaa olla kokonaan uudenlainen tapa ajatella tietojenkäsittelyn perusteita. Osittain niiden takana on joukko kansainvälisiä standardeja, jotka paperilta luettuina ovat hyvin vaikeatajuisia. Useimmiten esiintyviä virallisten tai tosiasiallisten standardien koodeja tai akronyymejä ovat TCP/IP, Z39.50 ja SGML.

Tietoverkkojen peruskäsitteitä ja standardien perusteita käsitellään kaikilla tietoliikennekursseilla ja useilla atk:n hyväksikäytön kursseilla. Monet tavallisista käyttäjistä ovat käyneet näitä kursseja ja kurssien jälkeen olleet entistäkin hämmentyneempiä. Erilaiset kaavakuvat ja käsitteiden luettelot eivät ole tehneet asioita yhtään helpommiksi ymmärtää.

Olen viime vuosien aikana kirjoittanut muutamia (enimmäkseen julkaisemattomia) artikkeleita tästä aihepiiristä ja muutaman kerran opettanut tavallisille mikrojen käyttäjille tietoverkkojen peruskäsitteitä. Olen yrittänyt kehitellä tapoja keskeisimpien käsitteiden havainnollistamiseksi niin, että kuka tahansa voisi ne ymmärtää. Tavallisen käyttäjän kannalta on nimittäin jokseenkin yhdentekevää kuinka monta bittiä jossakin kentässä on tai missä järjestyksessä kenttien tulee olla. Useimmille riittää selkeä mielikuva siitä, miten kaikki oikeastaan toimii.

Selostan tässä julkaisussa yleistajuisesti mitä tarkoitetaan asiakas-palvelin (*client-server*) -mallilla ja mitä tarkoittavat *käytännössä* lyhenteiden TCP/IP, Z39.50 ja SGML taustalla ovat standardit. Nämä asiat ovat todella tärkeitä. Niiden kanssa joutuvat useimmat kirjastoalan ammattilaiset jatkuvasti tekemisiin tulevien vuosien aikana.

Teksti on kirjoitettu aihepiiristä, joka elää ja muuttuu koko ajan. Sanomattakin on selvää, että kyseessä on laaja ja vaikeasti hallittava kokonaisuus. Sen vaikeudesta osaltaan kertoo englanninkielisen, opetuskäyttöön soveltuvan materiaalin vähäisyys. Tämän tekstin tulevasta kehityksestä on mahdotonta esittää mitään lupauksia. Otan kuitenkin mielelläni vastaan mielipiteitä, korjauksia ja ehdotuksia tekstin muuttamisesta tai laajentamisesta.

* * * * *

Käsillä olevaa tekstiä on keskeneräisessä muodossa käytetty oppimateriaalina vuosien 1996–1997 aikana Tampereen yliopiston täydennyskoulutuskeskuksen Tietoverkkoasiantuntija-kursseilla. JUHA HAKALA, KALERVO JÄRVELIN, SINIKKA KANGAS ja REIJO SAVOLAINEN ovat esittäneet arvokkaita korjauksia ja parannusehdotuksia tekstiin. JANNE HIMANKA on kirjoittanut dokumenttien nimeämistä koskevan osuuden 6.2 ja muutenkin kommentoinut tekstiä. Erityiset kiitokset edellä nimeltä mainituille henkilöille, kiitokset myös kaikille muille tekstin syntymiseen ja muotoutumiseen vaikuttaneille henkilöille.

Espoossa 15.11.1997

Timo Kuronen
Email: `timo.kuronen@gsf.fi`

Sisältö

Alkusanat **3**

Johdanto **7**

1 Tieto ja informaatio **11**

2 Navigointi tietoverkoissa **14**
2.1 Navigointistrategiat 15
2.2 Tiedostohierarkia 16
2.3 Hakusanojen käyttö 17
2.4 Mielleyhtymät 18
2.5 Muut navigointistrategiat 19
2.6 Asiakas-palvelin -malli 19

3 Internetin yhteyskäytännöt — TCP/IP **21**
3.1 Toimintaperiaate 21
3.1.1 Toimintaperiaate ja protokolla 23
3.2 Historia ja tausta 24
3.3 Sovellukset 25
3.4 TCP/IP:n “virallinen asema” 26

4 Z39.50 — Hajautetut kirjastojärjestelmät **29**
4.1 Johdanto 30
4.2 Z39.50:n toimintaperiaate 32
4.3 Z39.50:n historia ja nykytilanne 33
4.4 Z39.50:n palvelut ja osapalvelut 36
4.4.1 Yhteyden muodostaminen 36
4.4.2 Tietokantakysely 38
4.4.3 Tietueiden haku 42

4.4.4 Tietokantojen esittely 44
4.4.5 Laajennettu palvelu 45
4.4.6 Yhteyden sulkeminen 45
4.4.7 Muut toiminnot 46
4.5 Z39.50:n jatkokehitys 47

5 SGML ja rakenteiset dokumentit 48
5.1 Pari sanaa tekstinkäsittelystä 49
5.2 SGML:n historia 51
5.3 SGML:n rakenne ja käyttö 52
5.3.1 Esimerkki 52
5.3.2 SGML-esittely 53
5.3.3 Dokumentin tyyppimäärittely 54
5.3.4 SGML-koodattu dokumentti 56
5.4 SGML:n hyötykäyttö 58
5.5 SGML:n “suuret” sovellukset 61
5.6 HTML, XML ja World Wide Web 62
5.6.1 Laajennettava merkintäkieli 65

6 Nimeäminen ja paikantaminen 69
6.1 Tietojen paikantaminen 70
6.1.1 GILS:n palvelut 71
6.1.2 GILS:n seuranta 72
6.2 Dokumenttien nimeäminen 73
6.3 Dokumenttien kuvailu 75
6.3.1 Dublin Core 76
6.4 Resurssien kuvailun puitteet 78

7 Laitteiden kehitys 80
7.1 Verkkotietokone 81
7.2 Verkkotietokoneen ohjelmisto 82

Kirjallisuutta 85

Johdanto

Tieteellisten kirjastojen ja osittain yleistenkin kirjastojen tulevaisuus kytkeytyy tiiviisti tietokantojen ja tietoliikenneverkkojen nykyistä monipuolisempaan hyödyntämiseen [15]. Tämä muodostaa suuren haasteen kirjastoväelle, jolle sopeutuminen tietotekniikan nopeisiin ja joskus yllättäviinkin muutoksiin voi olla vaikeaa. Haasteeseen vastaaminen on kuitenkin välttämätöntä ja jotkut tutkijat pitävät mahdollista epäonnistumista kohtalokkaana koko kirjastolaitoksen tulevaisuudelle. Usein nimittäin kysytään, tarvitaanko tulevaisuudessa kirjastoja enää lainkaan. Jotkut kyselevät sitäkin, onko kirjoilla lainkaan tulevaisuutta.

Tietoliikenteen uudet palvelut ovat joka tapauksessa joko tulleet tai tulossa sekä kirjastoihin että kouluihin. Ne ovat osa *tiedon valtaväyläksi* kutsuttua tietoyhteiskunnan kehittämisideaa. Eri tahot sisällyttävät tähän ideaan hyvinkin erilaisia tavoitteita. Valtiovallan, elektroniikkateollisuuden ja viihdeteollisuuden tavoitteena näyttäisi olevan tilaajavideoiden (*video-on-demand*) levittäminen ainakin asutustaajamiin. Samalla maksullisten TV-kanavien määrä pyritään Suomessakin nostamaan vähintäänkin sataan. Tietoliikenneverkkojen nopeuttaminen ja viihdetarjonnan lisääminen kytkeytyvät Euroopan Unionin suunnitelmissa tiiviisti yhteen.

Valtaosa käynnissä olevista hankkeista tietoyhteiskunnan toteuttamiseksi on vahvasti teknologiapainotteisia. Tietoyhteiskunta näyttää toteutuvan mikrotietokoneita ja verkkoyhteyksiä hankkimalla. Sisällölliset kysymykset ovat jääneet sivummalle ja toiminnan tavoitteet näyttävät jotakuinkin hämäriltä. Käynnissä olevien muutosten jäsentämiseksi tarvitaan välttämättä sellainen yleinen viitekehys, jonka puitteissa teknologian ja toiminnan muutokset ja näiden muutosten avulla tavoitellut päämäärät pystytään johdonmukaisella tavalla hahmottamaan. Kirjastojen toiminnalle tällaisen viitekehyksen tarjoaa sopivalla tavalla määritelty virtuaalikirjaston idea.

Virtuaalikirjaston idea

Käsite *virtuaalikirjasto (virtual library)* on ollut käytössä muutaman vuoden ajan. Virtuaalikirjastolla tarkoitetaan hyvin usein näyttöluetteloiden (OPAC, *Online Public Access Catalog*) ja muiden kirjastopalveluiden käyttöä tietoliikenneverkkojen välityksellä [6]. Käsite voidaan määritellä myös laajemmin, jolloin mukaan tulevat kansalaisten yleistä tiedonsaantia ja kansalaiskeskustelun mahdollisuuksia koskevat näkökohdat [14]. Virtuaalikirjasto tarkoittaa jotakuinkin samaa asiaa kuin digitaalinen kirjasto, elektroninen kirjasto tai verkotettu kirjasto. Näistä digitaalisen kirjaston käsitettä käytetään hyvin yleisesti. Digitaalisuus viittaa kuitenkin tarpeettoman voimakkaasti aineiston tallennusmuotoon (digitaalinen analogisen tai fyysisen vastakohtana). Elektroninen kirjasto on nimityksenä hiukan vanhahtava (vrt. elektroninen tietojenkäsittely). Verkotettu kirjasto viittaa sekin tarpeettoman yksioikoisesti teknologiaan: tietoliikenneverkkojen hyväksikäyttöön.

Sanalla *virtuaalinen* tarkoitetaan suomenkielessä oletettua, näennäistä tai mahdollista. Fysiikassa virtuaalinen kuva on valekuva. Oxfordin englanninkielen sanakirjan mukaan virtuaalinen on *being or acting as what is described, but not accepted as such in name or officially*. The American Heritage Dictionary määrittelee puolestaan sanan virtuaalinen seuraavasti: *Existing or resulting in essence or effect though not in actual fact, form, or name*. Virtuaalinen antaa siten toden tuntuisen vaikutelman tai lopputuloksen.

Tietokoneiden yhteydessä virtuaalimuistilla tarkoitetaan käyttöjärjestelmän, keskusyksikön ja kovalevyjen avulla toteutettua keskusmuistin laajennusta. Se antaa suorituksessa olevalle tietokoneohjelmalle vaikutelman aivan kuin muistia olisi lähes rajattomasti, vaikka todellisuudessa tietokoneen levyiltä luetaan muistiin kulloinkin tarvittavat muistisivut. Fyysistä, todellista keskusmuistia saattaa olla käytettävissä huomattavasti vähemmän kuin ohjelman koko sinällään edellyttäisi. Tietokoneohjelman kannalta vaikutelma on silti hyvin todellinen. Ohjelman kannalta virtuaalimuistina toteutettu muisti on yhtä käyttökelpoista ja todellista kuin vastaavan kokoinen fyysinen muisti. Ainoa ero virtuaalimuistin ja fyysisen muistin käytön välillä ilmenee suoritusnopeudessa. Fyysisen muistin käyttö on nopeampaa — suorituksessa oleva tietokoneohjelma ei nopeuseroa kuitenkaan pysty ‘havaitsemaan’.

Sanan *virtuaali*kirjasto virtuaalisuus voidaan ymmärtää samassa periaatteellisessa merkityksessä kuin edellä tietokoneiden muistien virtuaalisuus. Virtuaalikirjasto antaa yhtä todellisen tuntuisia *informaatio*palveluja kuin

todellinen, fyysinen kirjasto. Palvelujen antamisen keinot ovat kuitenkin hyvin erilaiset. Virtuaalikirjasto voi tarjota vain informaatiota tai kopion digitaalisessa muodossa olevasta dokumentista tai kirjasta — ei fyysisiä kirjoja. Saadun informaation perusteella fyysinen kirja tai muu tiedonlähde voidaan haluttaessa paikallistaa.

Virtuaalikirjasto on käsitteenä uusi eikä siihen päällisin puolin tarkasteltuna näyttäisi liittyvän mitään harhaanjohtavia sivumerkityksiä. Sen voi katsoa sisältävän perinteisten kirjastojen palvelujen ohella elektronisuuden, digitaalisuuden ja verkottuneisuuden tarjoamat mahdollisuudet. Sana virtuaalinen viittaa mielekkäällä tavalla tietotekniikan sukulaiskäsitteisiin. Lisäksi virtuaalikirjasto käsitteenä säilyttää oikealla tavalla sanan virtuaalinen alkuperäisen, englanninkielisen merkityksen sitomatta käsitettä mihinkään yksittäiseen teknologiaan tai toteutustapaan.

Lyhyesti määriteltynä: *Virtuaalikirjasto on tietokoneiden ja tietoliikenteen avulla toteutettu informaatiojärjestelmä, joka antaa samantapaisia informaatiopalveluja kuin perinteinen kirjasto. Virtuaalikirjaston perustehtävät kirjastonhoitajan näkökulmasta katsottuna ovat samat kuin perinteisessä kirjastossa: kokoelmien organisointi ja käyttöönotto käyttäjäkuntaa mahdollisimman hyvin palvelevalla tavalla. Virtuaalikirjasto on kuitenkin samalla myös käyttäjän kirjasto. Sen kokoelmat karttuvat käyttäjän määrittelemän profiilin mukaisesti ja ylittävät yksittäisen kirjaston kokoelmat. Käyttäjä voi halutessaan liittää virtuaalikirjastoon omia tekstejään ja osallistua tällä tavoin omilla mielipiteillään tietoverkoissa käytäviin kansalaiskeskusteluihin.*

Virtuaalikirjasto ja yhteistyö

Virtuaalikirjastoa voidaan tarkastella sekä kirjastojen yhteistyön tavoitteena että sen muotona. Yhteistyön *tavoitteena* virtuaalikirjasto tulee olemaan laaja infrastruktuuriratkaisu, joka on kattava sekä alueellisesti että sisällöllisesti. Virtuaalikirjasto koostuu esittelyteksteistä ja ohjeista, valikoista, luetteloista, viite- ja tekstitietokannoista sekä synteettisistä dokumenteista. Erityisesti tieteellisillä kirjastoilla on tärkeä rooli virtuaalikirjaston kehittämisessä. Ne kukin panevat tarjolle kyseisen kehysorganisaation mielenkiintoisimmat ja tärkeimmät tietovarannot. Tieteelliset kirjastot voivat yhteistyössä alakohtaisten asiantuntijoiden kanssa kehittää synteettisiä dokumentteja, jotka toimivat samalla kertaa sekä ohjeina että käyttöliittyminä varsinaisiin tietolähteisiin.

Yhteistyön *muotona* virtuaalikirjasto on hyvin vaativa. Se edellyttää ensinnäkin, että kirjastonhoitajat pystyvät miettimään työnsä sisällön, menetelmät ja tavoitteet pitkällä aikavälillä uudelleen. Kyseessä on merkittävä muutos työorientaatiossa, joka tulee vaatimaan paljon aikaa. Virtuaalikirjaston toteuttaminen tulee vaatimaan suuria ponnistuksia aineistojen pitkän aikavälin saatavuuden, tavoitettavuuden ja käytettävyyden varmistamiseksi. Erityisen tärkeitä ovat järkevin perustein valitut viralliset ja tosiasialliset standardit. Aineistot on tallennettava sellaisessa muodossa, että ne ovat mahdollisimman immuuneja laite- ja ohjelmistoteknologian muutoksille. Aineistot on kuvailtava ja järjestettävä siten, että ne löytyvät monien erilaisten hakustrategioiden avulla. Aineistojen on oltava sellaisessa muodossa, että ne ovat helposti otettavissa käyttöön lukemattomissa, jatkuvasti muuttuvissa käyttötarkoituksissa.

Luku 1

Tieto ja informaatio

Tieto on klassisen filosofisen määritelmän mukaan *hyvin perusteltu tosi uskomus* [20]. Määritelmä on hiukan hämärän tuntuinen mutta sen keskeisin sanoma on yksinkertaisella tavalla selvennettävissä. Voidaan nimittäin asettaa kysymys, onko tieteellisessä lehdessä olevalla artikkelilla tai mikrotietokoneen kovalevylle tallennetulla tiedostolla uskomuksia. Vastaus on tietenkin *ei*! Uskomuksia ja erityisesti sellaisia uskomuksia, jotka ovat tosia tai vähintäänkin todenkaltaisia ja jotka lisäksi ovat hyvin perusteltavissa, on *vain* ihmisellä. Tieto on siten ihmisen ominaisuus. Paperilla tai tietokoneen muistissa ei koskaan — ei edes periaatteessa — voi olla tietoa.

Paperilla ja tietokoneen muistissa voi sen sijaan olla dataa ja informaatiota. Raja datan ja informaation välillä on epätarkka ja häilyvä. Ihmisen mielessä myös raja informaation ja tiedon välillä on häilyvä. Ei ole mitenkään itsestään selvää, missä vaiheessa informaatio muuttuu tiedoksi. Jonkinlainen yleiseen ajattelutapaan pohjautuva hahmotelma näiden kolmen käsitteen eroista on silti esitettävissä.

Data (sana on monikkomuoto sanasta datum) on jotain sellaista, johon ei välttämättä liity mitään merkitystä; joka ei ole ehkä millään tavalla tulkittavissa informaatioksi ja omaksuttavissa tiedoksi (esim. kohina TV:n kuvaruudulla ohjelmien päätyttyä). *Informaatio* on sellaista dataa, johon on liitetty tai johon on liitettävissä jokin merkitys tai tulkinta. Esimerkiksi laitteiden sisäiset signaalit ovat tässä mielessä informaatiota. Pieni osa kaikesta informaatiosta on luonteeltaan sellaista, että se on oppimisen ja omaksumisen avulla muunnettavissa *tiedoksi*. Muuntumisen mahdollisuus riippuu sekä informaation luonteesta että oppijan edellytyksistä (aistien toiminta, kielitaito, aikaisemmin omaksutut tiedot).

"Tietämisen portaat" nimetään usein seuraavasti: data, informaatio, tieto, ymmärrys ja viisaus. Kaksi ensiksi mainittua ovat ihmisen tajunnan kannalta pääasiassa ulkoisia, kolme jälkimmäistä kuuluvat yksinomaan mielen ja tajunnan piiriin. Ymmärrystä sen enempää kuin viisauttakaan ei pystytä määrittelemään täsmällisellä tavalla. Tavanomaisin ajattelutapa lienee sellainen, että tietojen käyttäminen ihmisen aktiivisessa toiminnassa johtaa vähitellen laajempien kokonaisuuksien hallintaan eli ymmärrykseen. Ymmärryksestä taas elämänkokemus vähitellen jalostaa ripauksen viisautta, puhutaanhan esimerkiksi iän mukanaan tuomasta viisaudesta. Tämän ajattelutavan mukaan jokainen ihminen pääsisi vähitellen nauttimaan sekä ymmärryksestä että viisaudesta. Käsitteet voidaan rajata kuitenkin myös toisella tavalla.

Tähän toisenlaiseen ajatteluun päästään luontevasti kiinni Majid Tehranianin oppimista koskevien pohdiskelujen johdattelemana [27]. Tehranian ei ensinnäkään usko Comten esittämään jaotteluun, jonka mukaan ihmisen oppiminen on kehittynyt historiallisesti mytologisesta teologisen kautta tieteelliseen. Tehranianin mielestä ihminen tarvitsee jatkuvasti näitä kaikkia. Myyttien moniselitteisyys ei ole mitenkään ristiriidassa tieteellisen tiedon kanssa, sillä myytit täydentävät tiedettä. Myytit, uskonnot ja tiede tyydyttävät ihmisten erilaisia tietämisen tarpeita. Myytit liittyvät inhimillisen solidaarisuuden tarpeisiin. Uskontojen avulla on perusteltavissa sellaiset eettiset normit, joille ei löydy tieteellistä selitystä. Tieteellinen tieto kertoo ihmisestä luonnon osana — myös ihmisestä itsestään.

Tehranian itse puhuu kasautuvasta, uudistavasta ja muuntavasta oppimisesta (additive, regenerative, transformative) [27, s. 43]. Kasautuva tieto ja siihen liittyvä oppiminen (tieteellinen ja teknologinen informaatio) muodostavat hierarkian alimman ja samalla yksinkertaisimman tason. Kasautuva tieto on kumulatiivista ja logiikan lakien säätelemää. Uudistava tieto ja siihen liittyvä oppiminen ovat oleellisesti monimutkaisempia. Siihen hierarkian tasoon liittyvät oleellisena osana psykososiaaliset voimat. Jokainen sukupolvi joutuu omien kokemustensa ja kärsimystensä kautta uudelleen oppimaan ja uudelleen luomaan edeltävien sukupolvien moraalisen tietämisen tason. Muuntava tieto ja oppiminen ovat vieläkin vaikeampia. Historian kulun henkiset ja moraaliset hyppäykset saattavat vaatia jopa geneettisiä muutoksia.[1] Hammurabin lait, kymmenen käskyä, Buddhan neljä tietä, Jeesuk-

[1]Tehranian ei tarkemmin selitä, mitä hän tarkoittaa tässä yhteydessä geneettisillä muutoksilla. Ehkä hän tällä tavoin haluaa korostaa, että kyseessä on kumouksellinen muutos ihmisten ajattelussa.

sen vuorisaarna ja YK:n ihmisoikeuksien julistus ovat sellaisia hyppäyksiä, jotka jättävät varjoonsa kaiken kasautuvan ja uudistavan tiedon.

Tämän ajatusmallin pohjalta voidaan uudelleen miettiä tiedon, ymmärryksen ja viisauden välistä suhdetta. Kuka tahansa voi oppimisen ja kokemusten avulla hankkia ja saada tietoja. Tietojen jalostuminen ymmärrykseksi edellyttää uudistavaa oppimista, joka kytkeytyy ihmisen koko elämän ajan kestävään kehitykseen. Ymmärryksen kehittyminen vie normaalissa tapauksessa vuosikymmeniä. Järkyttävät, syvällisesti vaikuttavat kokemukset, kuten sodat ja vakavat sairaudet, saattavat nopeuttaa oleellisesti tätä kehitystä.

Syvällisin viisaus sen sijaan kehittyy hyvin hitaasti ja sen kasvu tapahtuu historiallisessa mielessä hyppäyksinä. Tehranianin ajatukset tulevat tässä hyvin lähelle Hegelin ajatuksia. Hegel puhuu historian viekkaudesta, joka aika ajoin oman välttämättömyytensä voimasta synnyttää poikkeuksellisia yksilöitä. Viisaus tiivistyy näiden poikkeusyksilöiden syvällisen ajattelun tuloksena. Hegelin tässä tarkoituksessa useimmin mainitsemat nimet ovat Sokrates ja Jeesus.

Portaat tiedosta viisauteen voidaan edellä esitetyn pohjalta tiivistää seuraavasti. Vanhoihin ja uudempiin teksteihin kirjatun viisauden omaksuminen tiedoksi on kasvatuksen ja koulutuksen ongelma. Tiedon jalostuminen ymmärrykseksi on henkilökohtainen koko elämän ajan jatkuva oppimisen ja kasvun prosessi. Ymmärryksen puhkeaminen viisaudeksi saattaa olla verraten harvinaista kiireisten länsimaisten ihmisten keskuudessa. Se on ehkä jotain sellaista, josta useimpien meistä on turhaa edes haaveilla.

Luku 2

Navigointi tietoverkoissa

Virtuaalikirjasto rakentuu Internet-verkon palvelujen varaan. Ilman riittävän laajassa käytössä olevaa tietoliikenneverkkoa virtuaalikirjastoa ei voisi rakentaa. Internet on maailmanlaajuinen tietokoneiden ja tietoliikenneverkkojen verkosto. Se on laajin kaikista käytössä olevista tietoliikenneverkoista ja se kasvaa nopeammin kuin mikään muu verkko [1]. Se on syntynyt aikanaan Yhdysvaltain puolustushallinnon ja suurten yliopistojen tarpeisiin. Sen jälkeen se on levinnyt läntisen maailman korkeakouluihin ja tutkimuslaitoksiin. Nykyisin se kattaa käytännöllisesti katsoen koko maapallon.

Viime vuosien aikana Internetin laajeneminen on kytkeytynyt erityisesti informaatiopalvelujen kehittymiseen. Sellaiset ohjelmat kuten Gopher, WAIS, Mosaic[1] ja erityisesti Netscape ovat ainakin niminä tulleet suurellekin yleisölle tutuiksi. Näitä ohjelmia käytetään informaation levittämiseen ja hakemiseen. Internet-verkosta löytyy informaatiota mistä tahansa kuviteltavissa olevasta aihepiiristä. Vielä muutama vuosi sitten informaation levittäminen ja hakeminen edellyttivät erityistä perehtyneisyyttä atk-teknisiin kysymyksiin. Internetin uudet, edellä mainitut ohjelmat ovat avanneet nämä palvelut myös tavallisille kansalaisille.

Internetin suosio on lisääntynyt myös Suomessa. Suomalaisissa päivälehdissä on muutaman vuoden ajan säännöllisesti julkaistu Internet-verkkoa koskevia artikkeleita. Ne ovat lisänneet sekä suuren yleisön että viranomaisten mielenkiintoa Internet-palveluja kohtaan. Korkeakouluissa ja tutkimuslaitoksissa Internet on ollut käytössä koko sen ajan, kun verkkoon ylipää-

[1] Mosaicin jatkokehitys on lopetettu.

tään on voitu saada yhteys. Internet-solmujen määrä on Suomessa välilukuun suhteutettuna suurempi kuin missään muussa maassa.

2.1 Navigointistrategiat

Internetin informaatiopalvelut voidaan liittää seuraavien ohjelmien muodostamaan kokonaisuuteen. Ne ovat Gopher, WAIS ja Netscape (WWW). Kun näitä ohjelmia esitellään, puhutaan usein käsitteestä navigointi (*network navigation*) tai navigointityökalut. Voidaan käyttää myös käsitettä *Internet resource discovery* [21, 26]). Hyvin monien kirjoittajien mielestä edellä mainitut ohjelmat ovat tyypillisimpiä navigoinnissa käytettävistä työkaluista. Ne ovat monessa tapauksessa aluksi julkisohjelmia. Myöhemmin ne muutetaan tai niiden pohjalta kehitetään kaupallisia tuotteita. Navigoinnilla tarkoitetaan tässä (1) informaatiopalveluja antavan tietokonejärjestelmän etsimistä ja valintaa ja (2) dokumentin etsimistä ja valintaa kyseisestä informaatiojärjestelmästä.

Navigointi on käsite, jonka käyttö on tullut merkitykselliseksi vasta maailmanlaajuisten tietoliikenneverkkojen kehittymisen myötä. Mikrotietokoneen käyttäminen tekstikäsittelyyn tai yhteydenottaminen “tyhmällä” päätteellä paikalliseen tietokoneeseen ei edellytä navigointia. Navigointia tarvitaan vasta sitten, kun käytettävissä on suuri määrä palvelimia, joihin ei välttämättä tarvita käyttäjätunnuksia ja palvelimen valintaa ei voida tehdä sattumanvaraisesti tai pelkästään kokeilemalla.

Kun edellä mainittuja ohjelmia esitellään, ohjelmat esitellään yleensä itseisarvoisesti ilman ajatuksellista kytkentää kirjastojen tai tietojenkäsittelyn vakiintuneeseen käsitemaailmaan. Ohjelmat esitellään sellaisina kuin ne ovat. Sellaiset kysymykset kuin, miksi juuri nämä ohjelmat, miksi juuri näin käyttäytyvät ohjelmat, jäävät vastaamatta. Tarkoitukseni on tässä hyvin yleisesti johdatella navigoinnin *strategioihin*. Mainitut ohjelmat eivät toiminnallisesti ole sattumanvaraisia, ne kukin edustavat erilaista navigoinnin strategiaa, Gopher hierarkkista mallia, WAIS hakusanojen käyttöä tekstitietokannoissa ja Netscape/WWW mielleyhtymien (assosiaatioiden) tai linkkien varaan rakentuvaa verkkomallia.

Kun jotain informaatiota etsitään tietokoneesta tai mistä tahansa tietovarannosta, hakijalla täytyy olla mielessään jonkinlainen käsitys etsimisessä ja hakemisessa noudatettavasta menettelytavasta, hakustrategiasta. Ongelmaa voidaan havainnollistaa kirjaston käytöllä. Kun kirjastosta halutaan lainata

jokin kirja, ensin mennään sopivalle osastolle (esim. kaunokirjallisuus), sieltä etsitään tekijän nimen mukaan sopiva hylly ja hyllystä kirja löytyy aakkosjärjestyksen mukaisesta kohdasta. Jos tekijän nimi ei ole tiedossa, mutta hakija tietää kirjan nimen, hänen on ensin turvauduttava kortistoon. Jos kirjan nimikään ei ole pysynyt mielessä, on pyydettävä apua kirjastonhoitajalta. Aihepiirin tai avaintermien tuntemisesta ei välttämättä ole välitöntä apua.

2.2 Tiedostohierarkia

Kirjan etsiminen hyllystä ja kortiston selaaminen edustavat kahta erilaista hakustrategiaa. Tarkastellaan ensin hyllyjen katselua. Hyllyjärjestys muodostaa aihepiirin ja tekijän mukaisen struktuurin, joka periaatteiltaan on hyvin samankaltainen kuin tietokoneiden tiedostojärjestelmät. Kirjastoa vastaa tietokone, osastoja ja hyllyjä eri tasoiset hakemistot, tekijän nimen mukaista järjestystä vastaa sopiva tiedostojen nimeämiskäytäntö.

Hyllyjärjestys ei periaatteessa ole yhtään sen helpompi tai vaikeampi mieltää kuin tiedostohierarkia. Suurin ero syntyy siitä, että paikallisessa kirjastossa käydään lapsesta saakka ja oikean hyllyn luo kävellään suoraan UDK-luokituksesta mitään tietämättä. Kyläkauppaan tai omaan asuntoon ei myöskään mennä yleensä osoitteen perusteella. Sinne kävellään vaistonvaraisesti. Vanha hevonen tuntee kotikylällään reitin omaan talliinsa, sekään ei käytä karttaa eikä osoitetta.

Tiedostohierarkian mieltämisen vaikeus syntyy siitä, että hierarkiaa ei yleensä muodosteta minkään vakiintuneen mallin mukaisesti. Kirjastoissa käytetty UDK-luokitus on vanha ja perinteinen ja noudattaa monen asiakkaan mielestä täysin luonnollista ja itsestään selvää tapaa jaotella aihepiirit ryhmiin ja alaryhmiin. Tietokoneiden tiedostojärjestelmistä tällaista vakiintunutta, kaikkien ymmärtämää tapaa ei ole löydettävissä. Tarvitaan vuosikymmenien kokemus ja kehitys ennen kuin vakiintuneet käytännöt syntyvät. Vain ohjelmistojen järjestämiseen tällaisia käytäntöjä on syntynyt.

Gopher-ohjelman käyttämä ajattelutapa perustuu hierarkian käyttöön. Tiedostojen ja hakemistojen nimet ryhmitellään eritasoisiksi valikoiksi, joista voidaan haluttaessa piirtää hierarkkinen kaavio. Merkittävin ero mikrojen staattiseen tiedostohierarkiaan on linkkien käytössä. Gopherin valikossa sekä hakemiston nimi että tiedoston nimi voi olla linkki jonkin toisen tietokoneen hakemistoon tai tiedostoon. Käyttäjän näkemä, aluksi yhtenäiseltä

vaikuttava hierarkia onkin todellisuudessa pienten hierarkkisten kokonaisuuksien muodostama verkosto.

2.3 Hakusanojen käyttö

Kortiston selaamisen tavoitteena on löytää tieto kirjan sijaintipaikasta (osastosta ja hyllystä) käyttämällä hyväksi tietoa kirjan nimestä tai tekijästä. Useimmiten kortistot on muodostettu vain näillä perusteilla. Atk-pohjaisissa kortistoissa on käytettävissä enemmän hakuperusteita. Dokumenttien viitetiedoissa on nimekkeen ja tekijän lisäksi joukko muita vakiokenttiä. Niihin yleensä kuuluu erityinen asia- tai avainsanojen luettelo ja monissa tapauksissa tiivistelmän eli abstraktin kaikki sanat ovat käytettävissä etsinnässä.

Riippumatta siitä millä tavalla hakusanat on poimittu ja muodostettu, periaatteena niiden käyttämisessä on, että dokumenttia haetaan sen sisällön perusteella. Atk-pohjaisten kortistojen käytössä on vakiintunut CCL-kieli (Common Command Language), joka mahdollistaa hakusanojen katkaisut ja yhdistelyt (Boolen lausekkeet) ja aikaisempien hakutulosten (hakujoukkojen) täydentämisen hakua laajentamalla tai rajaamalla. Hakuohjelmalle kirjoitetaan lähtötiedoksi etsintälauseke, joka yksinkertaisimmillaan voi olla yksi sana, esim. sana kirjan nimestä. Etsinnän tuloksena on luettelo niistä dokumenteista, jotka ovat etsintälausekkeen mukaisia. Informaatikot pystyvät muodostamaan haut niin, että useimmat halutut dokumentit löytyvät ja tuloksena on mahdollisimman vähän epärelevantteja dokumentteja.

WAIS-ohjelman avulla voi muodostaa erityyppisistä tekstitiedostoista etsinnässä käytettävät invertoidut eli käänteistiedostot ja muut tarvittavat tiedostot. Internet-verkossa olevia WAIS-tietokantoja voi käyttää WAIS:n omien asiakasohjelmien avulla tai hakulausekkeen voi kirjoittaa Gopherin tai Netscapen lomakepohjaan. WAIS oli pitkään käytössä lähes kaikissa Gopher- ja Mosaic/WWW-palvelimissa. Nyttemmin sen tilalle on tullut muita ohjelmia.

Esimerkkejä laajan suosion saaneista, hakusanojen käyttöön perustuvista järjestelmistä ovat Altavista ja Lycos. Hierarkkisia rakenteita ovat Yahoo ja W3-konsortion Virtual Library.

2.4 Mielleyhtymät

Tietokoneiden hyväksikäyttö mahdollistaa tiedonhaussa vielä kolmannen hakustrategian, jolle ei ole välitöntä vastinetta kirjaston käytössä. Tätä strategiaa voidaan kansanomaisesti havainnollistaa heräteostoilla. Kun perheen ruokkija saapuu työpäivän jälkeen kotiin ja havaitsee paikallisen ruokakaupan jakamasta mainoksesta, että kahvia on tarjouksessa, hän luonnollisesti käyttää tilaisuutta hyväkseen ja menee kauppaan. Kaupassa hän lihatiskillä havaitsee, että myös lenkkimakkara on tarjouksessa. Tästä ajatus luontevasti johdattaa kaljakorien luo ja kaikesta runsaasta ostamisesta huolimatta seurauksena voi olla, että kahvi jää ostamatta. Kaupan mainokset ja suoritetut ostokset ovat ikään kuin johdattaneet asiasta toiseen.

Tietokoneiden hyperteksti- ja multimediajärjestelmät käyttäytyvät periaatteessa samalla tavalla kuin kaupan mainokset. Kun kuvaruudulla näkyvässä dokumentissa on kirkastettuna tai erilaisella värillä esitettynä sana tai fraasi, se johdattaa dokumentin katselijan toiseen dokumenttiin, joka käsittelee tarkemmin sanan tai fraasin ilmaisemaa aihepiiriä. Tapa millä tähän toiseen dokumenttiin käytännössä päästään riippuu käytetystä tietokoneesta ja sovelluksesta. Dokumentista toiseen kulkeminen mielenkiintoisia, mutta varsinaisen tavoitteen kannalta toisarvoisia sanoja seuraamalla saattaa johdattaa käyttäjän yhtä lailla hakoteille kuin edellä tarkoitetun heräteostajan.

Netscape asiakasohjelmana ja WWW (World Wide Web) palvelinohjelmana antavat hypertekstipalvelut, jotka sisältävät rajoitetusti multimediapiirteitä. Netscapen näyttämä dokumentti voi sisältää muotoiltua tekstiä ja värikuvia. Dokumentissa oleva hyperlinkki voi johtaa seuraavaan tekstidokumenttiin tai kohteena voi olla pelkkä rasterikuva, äänidokumentti tai lyhyt video. Videon voi yleensä ainoastaan katsella alusta loppuun. Itsenäisesti näytettävissä rasterikuvissa, äänidokumenteissa tai videoissa ei voi olla hyperlinkkejä. Selausohjelmissa on nykyisin mahdollista käyttää myös lomakepohjia, painonappeja, alasvetovalikkoja ja muita graafisten käyttöliittymien piirteitä. Näiden avulla käyttäjä voi lähettää tietoja WWW-palvelimessa oleville ohjelmille, jotka niiden perusteella valikoivat selattavaksi uusia dokumentteja.

2.5 Muut navigointistrategiat

Suoralta kädeltä on vaikea kuvitella jotain muuta periaatteellisesti täysin erilaista hakustrategiaa dokumenttien etsimiseksi. Järjestys, sisältö ja asiasta toiseen siirtyminen mitä ilmeisimmin kattavat rationaaliset tavat lähestyä informaation lähdettä. Monipuoliset informaatiojärjestelmät käyttävät luovalla tavalla hyväksi näitä kaikkia. Tavat saattavat limittyä niin, että käyttäjän voi olla vaikea havaita lähestymistapojen yhteen sulautumista. Hypertekstissä jonkin sanan (vaikkapa henkilönimen) klikkaaminen saattaa käynnistää sanastohaun, jonka tuloksena löytyvistä dokumenteista uusin tulee kuvaruudulle luettavaksi. Staattisen näköinen hyperlinkki voi olla dynaaminen ja joissakin tilanteissa linkki saattaa viitata jokaisella hakukerralla erilaiseen, jatkuvasti päivittyvään dokumenttiin.

2.6 Asiakas-palvelin -malli

Kaikki edellä käsitellyt ohjelmat perustuvat asiakas-palvelin -malliin (*client-server*). Asiakas (*client*) on tässä mallissa ohjelma, joka toimii käyttäjän työasemassa tai mikrossa. Palvelin (*server*) on ohjelma, joka toimii tietopalveluja antavassa tietokoneessa. Suuri joukko asiakasohjelmia voi tietoliikenneverkon[2] välityksellä olla näennäisesti samaan aikaan yhteydessä palvelimeen. Julkisia, vapaasti käytettäviä tietopalveluja antavassa koneessa ei käytetä käyttäjätunnuksia tai salasanoja. Palvelimeen ei tarvitse kirjoittautua käyttäjäksi. Mikä tahansa tietoliikenneverkossa oleva asiakasohjelma voi lähettää kyselyn. Kun palvelin vastaa kyselyyn, kyseessä on kertaluontoinen toimenpide. Vastauksen lähettämisen jälkeen palvelin ei välttämättä muista äskeisestä asiakkaastaan enää mitään. Asiakasohjelman ja palvelimen välille ei lainkaan muodostu varsinaista yhteyttä, istuntoa. Sen vuoksi puhutaan yhteydettömästä tai tilattomasta kommunikaatiotavasta.

Asiakasohjelmasta käytetään usein nimitystä käyttöliittymä (*user interface*) tai graafinen käyttöliittymä, silloin kun halutaan korostaa käyttöliittymän mahdollisuutta kuvakkeitten ja rasterikuvien esittämiseen. Jokaisella ohjelmalla on kuitenkin jonkinlainen käyttöliittymä, eikä sen tarvitse välttämättä liittyä millään tavalla asiakas-palvelin -malliin. Tämän vuoksi käytän nimitystä asiakasohjelma, vaikka se välillä vaikuttaa hiukan kömpelöltä.

[2] Internet-verkossa tietoliikenteessä käytetään TCP/IP-protokollaperheen palveluja.

Internet-verkosta ja sen antamista palveluista on viime vuosina julkaistu runsaasti kirjallisuutta ja lehtiartikkeleita. Hyvän johdatuksen tähän aihepiiriin antaa Ed Krolin kirja *The Whole Internet* [12]. Sitä voi mainiosti lukea ilman tietojenkäsittelyopin perustietoja.

Luku 3

Internetin yhteyskäytännöt — TCP/IP

Internet on maailman laajin ja nopeimmin kasvava tietoliikenneverkko. Sen yhteyskäytäntönä, protokollana, on TCP/IP (*Transmission Control Protocol/Internet Protocol*). Yhteyskäytännön perusteet opetetaan kaikilla tietoliikennekursseilla. Silti huomattava osa kursseilla olleista ei ymmärrä mikä TCP/IP on ja miten se toimii. Tässä luvussa TCP/IP:n toimintaperiaate selostetaan epätieteellisesti ja epäteknillisesti. Tämän selostuksen perusteella kukaan ei opi tekemään tai käyttämään tietokoneohjelmia. Selostus antaa kuitenkin mielikuvan siitä, miten TCP/IP toimii, ja sillä seikalla saattaa olla pysyvämpää merkitystä kuin nopeasti muuttuvien ohjelmien käyttöohjeilla.

3.1 Toimintaperiaate

Oletetaan että Helsingistä ollaan lähettämässä paksu asiakirjapino, esimerkiksi EU-sopimus liitteineen, Joensuuhun ja että kuljetuksessa käytetään hevosilla ratsastavia kuriireja, jotka kulkevat kestikievarista toiseen. Kestikievarit sijaitsevat Helsingissä, Porvoossa, Kouvolassa, Lappeenrannassa, Savonlinnassa, Kiteellä ja Joensuussa. Hevosella ratsastava yksittäinen kuriiri kulkee vain kahden vierekkäisen kestikievarin väliä. Helsingin ja Porvoon väliä ratsastava kuriiri ei siten jatka matkaa Kouvolaan. Porvoon ja Kouvolan välillä ratsastaa eri kuriiri. Paljon käytetyillä reiteillä on useita kuriireja. Harvoin käytetyillä reiteillä vain muutama.

Asiakirjapino kuljetetaan kirjekuorissa, joista kuhunkin mahtuu vain 50 sivua. Kuriirille annetaan kuljetettavaksi vain yksi kirjekuori kerrallaan. Jotkut kuriireista ovat huolimattomia ja saattavat hukata kirjeen tai sitten kuriiri saattaa matkan varrella yltyä ryyppäämään ja toimittaa kirjeen perille vasta viikkojen kuluttua.

Asiakirjapinon lähettäjä on korkea-arvoinen virkamies, joka vie pinon aluksi Postitoimistoon. Siellä asiakirjapinosta otetaan varmuuden vuoksi kopio, joka pannaan talteen. Asiakirjapino erotellaan 50 sivun nipuiksi, jotka pannaan kirjekuoriin. Jokaiseen kirjekuoreen merkitään osoite ja lisäksi juokseva numero, joka kertoo missä järjestyksessä paperiniput on perillä kasattava.

Postitoimiston virkailija vie numerojärjestyksen mukaisesti kirjeen kerrallaan Helsingin kestikievariin ja komentaa joka kerta: *Vie tämä kirje Joensuuhun!*.

Kestikievarin isäntä katsoo kirjekuorta ja lukee siinä olevan osoitteen. Hän kaivaa hyllystään mustilla vahakansilla varustetun vihkon ja etsii sieltä vasemman puoleisesta sarakkeesta kohdan *Joensuu*. Sanan kohdalla oikean puoleisessa sarakkeessa lukee *Porvoo*.

Kestikievarin isäntä antaa kuriirille kirjeen ja komentaa: *Vie tämä kirje Porvooseen!* Kuriiri katsoo kirjekuorta ja näkee, että osoitteeksi on kirjoitettu *Joensuu*. Kuriiri kuitenkin tottelee esimiestään, nousee ratsun selkään, vie kirjeen Porvooseen ja luovuttaa sen sikäläisen kestikievarin isännälle.

Porvoossa kestikievarin isäntä katsoo kirjekuorta ja lukee siinä olevan osoitteen. Hän kaivaa hyllystään mustilla vahakansilla varustetun vihkon ja etsii sieltä vasemman puoleisesta sarakkeesta kohdan *Joensuu*. Sanan kohdalla oikean puoleisessa sarakkeessa lukee *Kouvola*, ...

Tällä tavalla kirje kulkee etapin kerrallaan kohti Joensuuta. Joensuussa ne tulevat ensin keskikievarin isännälle, joka toimittaa ne Postitoimistoon. Siellä kirjeet avataan ja asiakirjapino kasataan samanlaiseksi kuin se oli ennen kirjekuoriin sijoittamista. Jos kaikki sujuu hyvin, kirjeet saapuvat Joensuuhun numerojärjestyksessä ja vieläpä niin, että yhtään kirjettä ei ole kadonnut eikä kirjeistä ole syntynyt matkan varrella ylimääräisiä kopioita.

Joskus käy kuitenkin niin, että kestikievarin isäntä Kiteellä äityy ryyppäämään viikkokausiksi eikä lainkaan ota vastaan kirjeitä. Tällöin kuriiri tuo kirjeen takaisin Lappeenrantaan ja sanoo kestikievarin isännälle, että Kiteellä taas ryypätään.

Lappenrannan kestikievarin isäntä kaivaa hyllystään mustilla vahakansilla varustetun vihkon ja etsii sieltä vasemman puoleisesta sarakkeesta koh-

dan *Joensuu*. Sanan kohdalla oikean puoleisessa sarakkeessa lukee *Kitee*. Koska Kiteellä ryypätään, kirjettä ei voi lähettää sinne. Isäntä huomaa, että Joensuu esiintyy vasemman puoleisessa sarakkeessa useamman kerran. Seuraavalla rivillä Joensuun kohdalla lukee *Savonlinna*. Isäntä käyttää tässä erityistilanteessa vaihtoehtoista reittiä ja käskee kuriiria viemään kirjeen Savonlinnaan.

Joskus käy niin, että Joensuussa havaitaan jonkin juoksevasti numeroiduista kirjeistä puuttuvan. Joensuusta lähetetään pikakuriirilla päinvastaista reittiä pitkin tieto Helsinkiin, että yksi kirjeistä puuttuu. Tieto toimitetaan postitoimistoon, joka on varustautunut tällaisen poikkeustilanteen varalta ja lähettää puuttuvan kirjeen uudelleen.

Joskus voi käydä niinkin, että kadoksissa ollut kirje vähän myöhemmin saapuukin Joensuuhun. Se on saattanut olla Kiteen kestikievarin isännän pöydällä odottamassa ryyppäämisen lopettamista. Sillä aikaa Helsinki on saattanut lähettää toisen kirjeen Savonlinnan kautta ja Joensuuhun tulee kaksi samansisältöistä kirjettä.

Lisäksi voi käydä vielä niin, että Helsingin ja Porvoon välisellä moottoritiellä kuriirit innostuvat ratsastamaan kilpaa ja myöhemmin liikenteeseen lähtenyt kuriiri tuleekin Porvooseen ensin ja lopputuloksena on se, että kirjeet tulevat Joensuuhun väärässä järjestyksessä.

Tästä kaikesta havaitaan, että postitoimiston päällikön on sekä lähettävässä että vastaanottavassa päässä oltava todella ammattitaitoinen ja tarkka kirjanpidossaan. Päällikön on varmistettava että kaikki lähetykseen kuuluvat kirjeet todella tulevat perille. Vasta kun kaikki tämä on varmistettu, postitoimiston päällikkö lähettää Helsinkiin kuittaussanoman, että lähetys on otettu kokonaisuudessaan vastaan. Tällöin Helsingin postitoimiston hoitaja purkaa tähän kyseiseen lähetykseen liittyvät varotoimet ja ilmoittaa korkealle virkamiehelle, että asiakirjapino on asianmukaisesti toimitettu Joensuuhun.

3.1.1 Toimintaperiaate ja protokolla

Tarkasti määritellyn, paljon yksityiskohtia sisältävän teknillisen ratkaisun havainnollistaminen kansantajuisesti jää pakosta puutteelliseksi. Havainnollistaminen tekee väistämättä väkivaltaa monille teknisille yksityiskohdille ja hienouksille. Toiminnan perusidea on kuitenkin toivottavasti tullut kohtuullisella tarkkuudella esitellyksi.

Internet-verkon tietoliikenne voidaan yhteyskäytäntöjen osalta jakaa kolmeen tasoon [4, s. 66]. Ylimpänä ovat sovellusten (sähköposti, tiedosto-

jen siirto, WWW, kirjastojärjestelmät, jne.) yhteyskäytännöt. Keskimmäisen tason muodostaa luotettavan kuljetuspalvelun tarjoava TCP. Alimmalla tasolla on epäluotettava, yhteydetön pakettien välitysjärjestelmä IP. Sovellustaso pyytää TCP:tä lähettämään laajan dokumentin vastaanottajalle. TCP purkaa dokumentin osiin ja pyytää IP:tä toimittamaan kirjeen kerrallaan verkon solmupisteinä toimivien IP-palvelimien välityksellä.

Edellä kansanomaisesti selostettu TCP/IP:n toimintatapa liittyy siten itse protokolliin seuraavasti. TCP eli kuljetusprotokolla (*Transmission Control Protocol*) vastaa postitoimistojen toimintaa. TCP varmistaa sen, että lähetettävä dokumentti menee kokonaisuudessaan ja vääristymättömässä muodossa lähettävästä pisteestä vastaanottavaan pisteeseen.

IP eli verkkoprotokolla (*Internet Protocol*) vastaa kestikievareiden toimintaa. IP huolehtii vain siitä, että sanoma (s.o. yksittäinen kirje, joka saattaa olla hyvin pieni osa kuljetettavasta dokumentista) lähtee matkaan oikeaan suuntaan. IP ei varmista mitään. Sanomalle voi matkan varrella tapahtua lähes mitä hyvänsä. Se voi kadota, muuttua, monistua ja viivästyä. Lisäksi sanomat voivat tulle perille väärässä järjestyksessä.

Internetin virhetilanteiden ja ongelmien selvittämiseksi on käytettävissä pikakuriiri ICMP (*Internet Control Message Protocol*). ICMP on tarkasti ottaen IP:n osa. Jos IP:n muussa toiminnassa esiintyy ongelmia, niistä ilmoitetaan ICMP-sanomien avulla, jotka siis ovat IP-sanomia. Jos ICMP-sanoman lähettämisessä esiintyy ongelmia, asialle ei Internetin keinoin ole enää paljon tehtävissä. Jos ICMP ei todellakaan toimi, sovelluksen käyttäjän tai tietokoneen ylläpitäjän on yritettävä turvautua puhelimen käyttöön. Jos puhelinkaan ei toimi, sitä ei voi käyttää puhelimen toimimattomuudesta kertovan viestin välittämiseen. Jäljelle jää enää perinteinen kirje tai pahimmassa tapauksessa on hypättävä pyörän selkään.

3.2 Historia ja tausta

Internet-verkolla ja sen prokollilla on kohtalaisen pitkä historia. Ensimmäinen tärkeä merkkipaalu on ARPA-verkko vuodelta 1968 (tietoverkkojen historiasta ks. Salus [24]). Se oli ensimmäinen pakettikytkentäinen verkko, jonka kehittämisestä ja käytöstä saatujen kokemusten varaan kaikki myöhempi tietoverkkojen kehitystyö on rakentunut.

Tietoliikenteen kehittäminen on Yhdysvalloissa aina ollut keskeisessä asemassa puolustuslaitoksen rahoittamassa tutkimustyössä. Myös TCP/IP

on alunperin kehitetty DoD:n, *Department of Defensen* (s.o. Yhdysvaltain puolustusministeriö) toimeksiannosta. Ensimmäiset määritykset protokollasta ovat peräisin vuodelta 1974 ja ensimmäinen toimiva versio valmistui vuonna 1978.

TCP/IP:stä ei olisi koskaan kehittynyt maailman tietoliikenteen tärkeintä protokollaa ilman DoD:n ja Berkeleyn yliopiston tiivistä ja erikoislaatuista yhteistyötä. Berkeleyssä kehitettiin tuohon aikaan Unix-käyttöjärjestelmästä yliopistokäyttöön paremmin soveltuvaa BSD-versiota (Berkeley Software Distribution). Käyttöjärjestelmää levitettiin yliopistoihin lähdekielisine ohjelmineen. Se oli jo silloin ja on edelleenkin harvinaista ja poikkeuksellista.[1]

BSD-Unix toimi DEC:n (Digital Equipment Corporation) VAX-tietokoneissa (1970-luvun lopulla VAX 11/750), joita käytettiin korkeakouluissa ja tutkimuslaitoksissa hyvin yleisesti (Unix:n historiasta tarkemmin ks. Salus [23]). Käyttöjärjestelmän versio 4.2BSD vuodelta 1983 sisälsi TCP/IP:n täyden toteutuksen. Tuosta ajasta lähtien TCP/IP on kuulunut jossakin mielessä tiedeyhteisön yhteiseen omistukseen. Ohjelmia on tutkittu, opiskeltu, korjattu, parannettu ja käytetty hyväksi omissa ohjelmissa. Tämän avoimuuden perua ovat TCP/IP-toteutusten korkea tekninen taso ja virheettömyys. Ratkaisu on ollut todella ainutlaatuinen. Kaikki muut tietoliikenteen ohjelmat ovat kaupallisia tuotteita ja lähdekieliset ohjelmat ovat huolella varjeltuja salaisuuksia.

3.3 Sovellukset

Kaikki Internetin tietoliikennepalvelut toimivat tavalla tai toisella TCP/IP-protokollaperheen varassa. IP on verkon peruspalveluna käytössä kaikkialla. TCP:n rinnalla toimii muita, yksinkertaisempia protokollia.

Tärkeimmät käyttäjän näkemät TCP/IP:n sovellukset ovat sähköposti (email), tiedostojen siirto (ftp) ja pääteyhteys (telnet). Kaikissa näissä sovelluksissa käytetään tietoliikenteessä TCP/IP:n palveluja. Sovelluksissa on sitten paljon muitakin, sovelluskohtaisia toimintoja, jotka käyttäjän mielessä helposti sekoittuvat tietoliikenteen yksityiskohtien kanssa. Sähköpostiin liittyy paikallinen viestien kirjoittaminen ja katselu sekä kansioiden hallinta.

[1] Suomessa Teknillisen korkeakoulun tietojenkäsittelyopin laitos otti ensimmäisenä Unix:n käyttöön 1970-luvun lopulla. Professori REIJO SULONEN toi käyttöjärjestelmän Yhdysvalloista.

Tiedostojen siirrossa käsitellään hakemistoja, pääteyhteyteen liityvät merkkivalikoimien ongelmat ja ristiriidat eri koneiden komentotulkkien välillä.

Kirjastojärjestelmien sovellustason yhteyskäytännöt perustuvat samaten TCP/IP:n käyttöön. Vaikka esimerkiksi myöhemmin esiteltävä kirjastojärjestelmien esperanto, Z39.50, on alunperin ajateltu ISO:n OSI-mallin mukaisten protokollien maailmaan, sitä käytännössä käytetään yksinomaan TCP/IP:n avulla.

Verkkotietokoneiden käytössä hyvin keskeiseksi tulee muodostumaan verkon välityksellä toimiva tiedostojärjestelmä NFS (*Network File System*). NFS on Unix-koneita valmistavan SUN Microsystemsin avoin ja julkinen spesifikaatio, jonka mukaisesti toimivia ohjelmia käytetään kaikissa Unix-toteutuksissa. NFS toimii IP:n avulla ja käyttäjä näkee sen palvelut seuraavasti. Käyttäjän työasemassa voidaan ottaa käyttöön tietoverkon takana olevaan palvelimeen sijoitettu tiedostojärjestelmä tai sen osa niin, että käyttäjästä tai sovelluksesta näyttää aivan kuin tiedostojärjestelmä olisi itse työasemassa. Työasemassa ei omaa levytilaa tarvita välttämättä lainkaan. Verkkotietokoneiden levytilan käyttö tulee perustumaan nimenomaan NFS:n käyttöön. Ei siis ole mikään sattuma, että SUN on yksi verkkotietokoneen pääasiallisista kehittäjistä.

3.4 TCP/IP:n “virallinen asema”

TCP/IP ei ole virallinen standardi eikä siitä sellaista luultavasti koskaan tulekaan. Yhdysvaltain puolustusministeriö (DoD) on rahoittanut TCP/IP:n kehittämisen. Nykyisin sitä ylläpitää Internet Societyn asettama IETF (*Internet Engineering Task Force*). Internet Society on kansainvälinen yhdistys, johon kuuluu tutkimuslaitoksia, korkeakouluja ja suuri joukko henkilöjäseniä. IETF on suppea asiantuntijaelin, joka pitää Internetin yhteyskäytäntöjen osalta langat käsissään.

Internetin yhteyskäytäntöjen ja muiden verkkotyöskentelyssä tarvittavien “standardien” kehittäminen hoidetaan mahdollisimman kevyellä organisaatiolla, usein vieläpä nopeasti ja tehokkaasti. IETF antaa jollekin asiantuntijaryhmälle tehtäväksi laatia ehdotus jostakin uudesta asiasta. Ehdotus julkaistaan Internetissä vapaasti levitettäväksi RFC-sarjassa (*Request For Comments*), ja jos se saa taakseen riittävän laajan kannatuksen, IETF hyväksyy sen. Ehdotuksen hyväksyminen yleensä edellyttää uusien toimintojen tai käytäntöjen demonstroimista julkisohjelmien avulla. Julkisohjelmat

ovat veloituksetta levitettäviä ohjelmia, joita tehdään Yhdysvaltain yliopistoissa ja tutkimuslaitoksissa.

Kaikki Internetin tärkeimmät palvelut ovat vakiinnuttaneet asemansa nimenomaan laajasti käytettyjen, korkeatasoisten julkisohjelmien varassa. Internet-verkkoa ei olisi nykyisessä laajuudessaan koskaan syntynyt ilman laajaa julkisohjelmien tarjontaa. Julkisohjelmien kehittämiseen liittyy oleellisena osana suuri määrä yhteisöllistä työtä. Julkisohjelmat levitetään useimmiten lähdekielisinä, jolloin innostuneet ja asiantuntevat käyttäjät etsivät niissä olevat virheet je tekevät korjausehdotuksia. Tästä johtuu se, että eniten käytetyt julkisohjelmat ovat laadukkaampia ja virheettömämpiä kuin suurten ohjelmistotalojen kalliit, kaupalliset ohjelmat. On esimerkiksi luultavaa, että yksikään kaupallisista tekstinkäsittelyohjelmista ei koskaan tule saavuttamaan sitä virheettömyyden tasoa, jolle professori DONALD KNUTHIN ohjelmoima TEX on julkisohjelmana päässyt.

Yksi Internetin vanhimmista RFC-julkaisuista on RFC 822 (*Standard for the Format of ARPA Internet Text Messages*) vuodelta 1982. Se määrittelee Internet-verkon sähköpostissa käytettävien sanomien muodon. Tätä määrittelyä täydentävät MIME-standardit RFC 2045–2049 (MIME, *Multipurpose Internet Mail Extensions*) vuodelta 1996. MIME määrittelee tavan, jolla sähköpostin sanomia voidaan kirjoittaa erilaisia merkkivalikoimia käyttäen ja millä tavoin sanomiin voidaan sisällyttää liitetiedostoina kuvia, valmiiksi ladottuja dokumentteja, jne.

Kansainvälinen standardointijärjestö ISO (*Internation Organization for Standardization*) on luonut monitasoisen OSI-mallin (*Open Systems Interconnection*) tietoliikenteen kuvaamiseksi. OSI-mallin mukaisesti on vuosien mittaan kansainvälisenä yhteistyönä tehty suuri määrä tietoliikenteen standardeja. ISO:n työ on erittäin hidasta ja vaivalloista ja tämän takia äärimmäisen kallista. Myös valmistuneet standardit ovat yleensä kalliita eikä niitä ole Internetistä saatavana. OSI-mallin mukaisiin standardeihin ei yleensä liity julkisohjelmien tarjontaa. Useimmat tarjolla olevista ohjelmista ovat kaupallisia tuotteita ja yleensä hyvin kalliita. Ohjelmissa on lisäksi runsaasti virheitä ja niiden käyttöönotto on hankalaa. Seurauksena onkin, että OSI-mallin mukaista, maailmanlaajuista tietoliikennettä ei ole syntynyt. Ei sittenkään, vaikka monien maiden telelaitokset ja useat ministeriöt ovat määrätietoisesti ponnistelleet OSI-mallin mukaisen tietoliikenteen vakiinnuttamiseksi. Välillä ponnistelut, mm. EU:n piirissä, ovat saaneet koomisia mittasuhteita (ks. tältä osin Bangemannin raportin kritiikistä Kuronen [13]).

Ainoa OSI-mallin mukainen palvelu, joka edes jossain määrin on levin-

nyt käyttöön, on X.400-standardin määrittelemä sähköposti. Useimmat sähköpostin käyttäjistä ovat luultavasti jossakin vaiheessa törmänneet Internet-postin ja X.400-postin yhteensopimattomuudesta aiheutuviin ongelmiin. Myös Suomessa X.400 on ollut käytössä varsinkin ministeriöiden ja eräiden hallintovirastojen piirissä. Siitä ollaan kuitenkin vähitellen luopumassa. Mm. Eduskunta on ottanut kansanedustajien esittämän arvostelun takia Internet-postin käyttöön.

Internet-verkon tyylisen tietoliikenteen kehittäminen on yhteisöllistä työtä, jossa tarvitaan suunnattoman ihmisjoukon mielenkiintoa, ammattitaitoa ja eräänlaista talkoohenkeä. Hyvin monet näistä ihmisistä ovat yliopistojen ja korkeakoulujen opiskelijoita, jotka saattavat käyttää uskomattomia määriä omaa aikaansa julkisohjelmien parantamiseen. Näitä ihmisiä löytyy muualtakin: tutkimuslaitoksista, puolustuslaitosten yksiköistä, sairaaloista, kouluista, kirjastoista, jne. Internetin kehittyminen nykyiseen laajuuteensa ja muotoonsa on varmasti jonkin asteinen ihme (olisiko maailman kahdeksas ihme?). Monet valtiolliset tahot ja yritykset ovat aika ajoin yrittäneet kaapata jotain Internetin osa-aluetta hallintaansa — säännöllisesti hyvin huonolla menestyksellä.

Virallisen tai tosiasiallisen standardin vakiintuminen yleiseen käyttöön on siten kaikesta päätellen monitahoinen ongelma. Standardin virallisuus tai sen tekemisen kalleus eivät mitenkään yksikäsitteisesti näytä johtavan siihen, että standardia myös alettaisiin noudattaa. Ainakin tietoliikenteen standardien kohtalosta on pääteltävissä, että vakiintuakseen standardin tulee välttämättä saada laaja käyttäjien hyväksyntä, joka ilmenee monipuolisena julkisohjelmien ja kaupallisten ohjelmien tarjontana. Julkisohjelmien tarjonta tavoittaa parhaiten akateemiset yhteisöt, kaupalliset ohjelmat tulevat käyttöön hallintovirastoissa ja liike-elämän piirissä.

Luku 4

Z39.50 — Hajautetut kirjastojärjestelmät

Tämä luvun aiheena on hajautetuissa kirjastojärjestelmissä ja tekstitietokannoissa käytettävä yhteyskäytäntö Z39.50. Aihe on jossain määrin teknisluontoinen mutta se on silti hyvin tärkeä ja perusasiat ovat helposti ymmärrettävissä. Luvun tärkein sanoma on kohdassa 4.2 sivulla 32. Jos haluaa pelkän yleiskuvan tästä asiakokonaisuudesta riittää, jos lukee luvun alusta johdannon ja edellä mainitun kohdan ja tämän lisäksi ehkä kohdan 6.1 sivulla 70.

Aluksi kuitenkin pari sanaa tekstitiedonhakua koskevasta terminologiasta. Käyttäjän kannalta katsottuna tekstitiedonhaku etenee suunnilleen seuraavasti. Käyttäjä hakee tietokannoista tekstidokumentteja käyttäen lähtötietoina selväkielisiä sanoja, koodeja ja numeroarvoja [10]. Niistä kootaan tiettyjen sääntöjen mukaisesti (mm. Boolen operaattoreita ja sulkuja hyväksikäyttäen) etsintälauseke, joka lähetetään yhdelle tai useammalle tekstitietokannalle. Ensimmäisessä vaiheessa vastaukseksi saadaan luettelo, hakujoukko (*retrieval set*), niistä dokumenteista, joissa sanat esiintyvät tai jotka täsmällisemmin ilmaistuna ovat etsintälausekkeen mukaisia. Toisessa vaiheessa käyttäjä valitsee hakujoukosta ne dokumentit, jotka hän haluaa osittain tai kokonaan nähtäväkseen.

Tietokantatyöskentelyssä on siten selvästi erotettavissa kaksi vaihetta: ensin *etsitään* soveltuvat dokumentit, jotka seuraavaksi *haetaan* katseltavaksi. Englanninkielisissä teksteissä näistä vaiheista käytetään sanoja *search* ja *retrieval*. Suomen kielessä kumpikin on totuttu kääntämään sanaksi *haku*, joka hävittää näiden vaiheiden välisen tärkeän eron. Tämän vuoksi seuraa-

vassa käytetään sanan *search* suomenkielisenä vastineena sanaa *etsintä*, ja vastaavasti sanan *retrieval* vastineena sanaa *haku*. Tämä vaikuttaa samalla yhdyssanojen kääntämiseen, *search expression* on *etsintälauseke*.

4.1 Johdanto

Kirjastoautomaation kehitys käynnistyi bibliografioiden tekemisestä ja läheinen ajatuksellinen kytkentä luetteloihin ja bibliografioihin on edelleenkin olemassa. Automaation ensisijaisena tarkoituksena on ollut ja on edelleen viitetietojen nopeampi, tarkempi, luotettavampi ja taloudellisempi käsittely. Ja tämä aivan siitä riippumatta, kuka on tarkasteltavana olevan järjestelmän varsinainen käyttäjä — olipa hän kirjastonhoitaja, informaatikko, kirjojen ostaja, luetteloija tai joissakin tapauksissa tietojen tarvitsija, lukija.

Lukijan kannalta viitetietojen käsittelyllä on kuitenkin vasta toissijainen merkitys. Lukija hakee ensisijaisesti informaatiota jostakin tietystä asiasta ja viitetiedoilla on vain väliaikaista merkitystä; ne auttavat varsinaisen informaatiolähteen paikallistamisessa. Viitetietojen käsittelyn hallitseva asema kirjastoautomaatiossa tulee vähitellen väistymään. Painopiste tulee jollakin aikavälillä siirtymään pelkästä viitteiden käsittelystä täystekstidokumenttien käsittelyyn. Viitetiedot eivät silti katoa mihinkään, niillä on merkitystä myös tulevaisuudessa.

Viitetietojen hyväksikäytön varaan on vuosien varrella kehitetty huomattava joukko automatisoituja kirjastopalveluja. Näitä ovat hankinta-, saapumis-, lainaus- ja kierrätysjärjestelmät. Viimeisimpänä on otettu käyttöön kirjastojen välinen kaukopalvelujärjestelmä (ILL, *Interlibrary Loan*). Kirjaston järjestelmistä vain viitetietokantojen käytöllä on suoranaista mielenkiintoa asiakkaille. Kaikki muut järjestelmät on tarkoitettu kirjastojen henkilökunnan käyttöön.

Viitetietojen käsittelyyn on tehty lukuisia erillisiä järjestelmiä (VTLS, TRIP, GEAC, BRS, STAIRS, Primas, Pallas, Minttu jne.), jotka monissa tärkeissä toiminnallisissa seikoissa poikkeavat oleellisesti toisistaan. Tietokantojen tietuemuodoista on laadittu standardeja (mm. USA:n Kongressin kirjaston MARC-formaatti ja sen lukuisat johdannaiset) ja tietokantakyselyissä käytettävien etsintälausekkeiden muoto on standardoitu (CCL-kieli, ISO 8777). Kaikki muu onkin ollut aivan viime vuosiin saakka standardoimatta ja se aiheuttaa käyttäjille ongelmia. Käyttäjän näkökulmasta katsottuna eri kirjastojärjestelmät toimivat toisistaan poikkeavalla tavalla

eivätkä järjestelmät pysty kommunikoimaan keskenään.

Vallitseva tilanne on kohtalaisen ongelmaton sellaisen käyttäjän kannalta, joka käyttää ammattimaisesti yhtä ainoaa kirjastojärjestelmää. Tällainen käyttäjä ei koskaan törmää standardien puutteesta aiheutuviin ongelmiin, sillä hänelle käytössä oleva järjestelmä on standardi *par excellence*. Ongelma tulee esille vasta silloin, kun käyttäjä joutuu vuoronperään käyttämään *eri* järjestelmiä. Ammattimainen käyttäjä on tässä tilanteessa vielä suuremmissa vaikeuksissa kuin satunnainen käyttäjä. Jos ammattikäyttäjä on ensisijaisesti tottunut jonkin tietyn järjestelmän käyttöön, hän on harjaantunut juuri tämän järjestelmän mukaisten komentojen, lyhenteiden ja oikopolkujen käyttöön (tyypillisesti funktionäppäimet). Tästä hyvästä harjaantumisesta on enemmän haittaa kuin hyötyä muiden, aivan eri tavalla käyttäytyvien järjestelmien käytössä. Refleksinomaiset toiminnot voivat aiheuttaa sekavia virhetilanteita väärässä yhteydessä käytettyinä.

Kirjastojärjestelmien standardoinnilla on mahdollista yhtenäistää periaatteessa mitä tahansa järjestelmien rakenteita, toimintoja tai komponentteja. Ensimmäinen ajatus voisi olla käyttöliittymän standardoiminen sellaiseksi, että kaikki ohjelmat näyttäisivät käyttäjän kannalta täsmälleen samanlaisilta ja myös toimisivat täsmälleen samalla tavalla. Tämä tekisi käyttäjän elämän helpoksi ja jättäisi silti pelivaraa järjestelmän rakenteen, toteutuksen, tiedosto- ja tietuemuotojen valinnan suhteen. Tämä ei kuitenkaan ole ainoa eikä välttämättä paras mahdollinen ratkaisu.

Asiakas-palvelin -malli mahdollistaa hyvin elegantin tavan standardoinnin kohdistamiseen. Jos standardoidaan sopivalla tavalla pelkästään asiakasohjelman ja palvelimen välinen vuoropuhelu, tehdään mahdolliseksi se, että *mikä tahansa asiakasohjelma voi keskustella minkä tahansa palvelimen kanssa*. Toisin sanoen TRIPin asiakasohjelmalla voi hakea tietoja VTLS-palvelimelta ja VTLS:n asiakasohjelmalla voi vastaavasti hakea tietoja TRIPin palvelimelta. Standardointi luo tällöin eräänlaisen keinotekoisen *esperanton*, rajoitetun kielen, jolla asiakasohjelmat ja palvelimet keskustelevat keskenään. Kirjastojärjestelmien esperanto ei välttämättä ole ilmaisuvoimaltaan paras mahdollinen. Jonkin yksittäisen kirjastojärjestelmän oma, natiivi kieli voi olla ilmaisuvoimaisempi. Yhteisen kielen etuna on kuitenkin se, että kaikki ymmärtävät tätä kieltä.

Kirjastojärjestelmien *esperanto* on nimeltään Z39.50. Kielen ja sen oudon näköisen nimen historia selostetaan hiukan jäljempänä. Aluksi kuitenkin lyhyesti Z39.50:n toimintaperiaate. Se kannattaa lukea hyvin hitaasti ja ajatuksen kanssa. Toimintaperiaate on nimittäin erittäin yksinkertainen ja

juuri yksinkertaisuutensa takia se helposti unohtuu, kun standardin yksityiskohtiin alkaa tarkemmin paneutua.

4.2 Z39.50:n toimintaperiaate

Z39.50:n mukainen toiminta perustuu asiakas–palvelin -malliin. Käyttäjällä on asiakasohjelma (client), joka useimmiten toimii työpöydällä olevassa mikrossa. Asiakasohjelma voi olla myös Unix-tyyppisessä tietokoneessa, jota käytetään työasemana tai X-päätteen (verkkotietokoneen) avulla. Tietoliikenneyhteyksien takana on erillisiä palvelimia (server), joissa tietokannat sijaitsevat. Z39.50 määrittelee pelkästään asiakkaan ja palvelimen välillä käytävän keskustelun määrämuodot. Se määrittelee ne puheenvuorot, joita kumpikin osapuoli voi käyttää. Lisäksi se määrittelee minkälaisia puheenvuoroja missäkin tilanteessa on mahdollista esittää. Tietoliikenteen kannalta Z39.50 määrittelee vain sovellustason yhteyskäytännön. Varsinainen tietoliikenne kaikkine teknisine yksityiskohtineen hoidetaan Internet-verkossa TCP/IP-protokollien avulla.

Z39.50:n asiakasohjelma (standardi käyttää nimitystä *origin*) ja palvelinohjelma (vastaava nimitys on *target*), keskustelevat keskenään sovellustason sanomia (PDU, Protocol Data Unit) vaihtamalla. Keskustelu on hyvin yksitotista. Asiakas lähettää pyynnön (Request), johon palvelin vastaa (Response). Erilaisia pyyntöjä ei ole kovin monia ja pyynnöt ovat helposti ymmärrettävissä. Tärkeimmät pyynnöt ovat: yhteyden muodostaminen palvelimeen (s.o. tietokantaan), tietokantakysely, tietueiden haku ja yhteyden sulkeminen.

Asiakkaan ensimmäisen sanoman tarkoituksena on yhteyden muodostaminen palvelimeen. Palvelin vastaa siihen joko hyväksymällä tai hylkäämällä yhteyden muodostamista koskevan pyynnön. Seuraavaksi asiakas lähettää tietokantakyselyn, johon palvelin vastaa ilmoittamalla mm. hakujoukon koon ja muita hakujoukon tietoja. Hakujoukon sisältämien osoitetietojen avulla asiakas voi pyytää katseltavaksi varsinaisia tietokannan tietueita, jotka palvelin lähettää vastaussanomassa. Tämän jälkeen asiakas voi lähettää lisää kyselyjä ja tietuepyyntöjä. Kyselyjen ja tietuepyyntöjen loputtua asiakas sulkee yhteyden.

Z39.50-standardi on laaja (156 sivua) ja se on tungettu täyteen vaikean tuntuisia käsitteitä ja teknisiä yksityiskohtia. Standardin mukainen perustoiminta on kuitenkin juuri niin yksinkertainen kuin edellä on kuvattu. Standar-

dia tai siitä laadittuja artikkeleita lukevan henkilön on tärkeää jo etukäteen omaksua ajatus siitä, että standardi on perusasioiden osalta selkeä ja kenen tahansa helposti ymmärrettävissä. Yksityiskohtien ei saa siis antaa hämärtää tätä lähtökohtaa. Jos etukäteen suhtautuu niin, että pitää asiaa ylitsepääsemättömän vaikeatajuisena, sille käsitykselle saa helposti vahvistuksen standardia selailemalla. Oikea ennakkoasenne on sen vuoksi hyvin tärkeä.

4.3 Z39.50:n historia ja nykytilanne

Kirjastojärjestelmien esperanto on nimeltään Z39.50. Täydellisempi nimi on *Information Retrieval: Application Service Definition and Protocol Specification* [2, 9, 16]. Kielen koodinimi on outo, aivan kuin jostakin scifi-julkaisusta. Nimellä on kuitenkin hyvin arkipäiväinen tausta. Z39.50 on amerikkalaisen ANSI/NISO:n standardi.[1] ANSI on liittovaltion standardoinnista vastaava virasto *American National Standards Institute*, joka on valtuuttanut NISO:n, *National Information Standards Organization*, tekemään kirjastoalan, kustannustoiminnan ja tietopalvelujen standardit. NISO on ANSI:n asettama komitea, josta käytetty koodi on Z39. NISO:n standardit on puolestaan juoksevasti numeroitu ja ne ovat muotoa Z39.nn. Esimerkiksi MARC-formaatin määrittelevä standardi on Z39.2, Z39.53 on luonnollisten kielten kolmikirjaimisten koodien luettelo (esimerkiksi englannin kieli on ‘eng’) ja CCL:n ANSI-vastine on Z39.58. Kaikki numerot eivät ole enää käytössä, osa standardeista on vuosien varrella vanhentunut.

Standardista Z39.50 on vuosien mittaan syntynyt useita eri versioita, jotka tunnistetaan koodiin liitetyn vuosiluvun perusteella. Vanhin niistä on Z39.50-1988, jonka kehitystyö käynnistyi jo vuonna 1984. Standardin uudemmat versiot ovat Z39.50-1992 ja Z39.50-1995, joka on virallinen, voimassa oleva versio. Standardin seuraavan version kehitystyö on jo käynnissä.

Z39.50-1988 Standardin vanhimman version (1988) mukaisesti toimivia ohjelmia ei ilmeisesti kehitetty koskaan. Hiukan muunnettuna sitä käytettiin WAIS:n (*Wide Area Information Servers*) yhteyskäytäntönä. WAIS on asiakas-palvelin -mallin mukainen hajautettu, tekstidokumenttien hallintaan

[1] Standardin tekstin saa erilaisina versioina WWW:stä. Paras lähtökohta etsimiselle saattaa olla:
`http://ds.internic.net/z3950/z3950.html`.

tarkoitettu tietokantaohjelmisto [11]. Sen kehittivät yhteistyössä yritykset Dow Jones, Thinking Machines, Apple ja KPMG Peat Marwick. Kyseessä on siis kaupallisten yritysten tuottama julkisohjelma (ilmaisohjelma) — ratkaisu, joka monestakin syystä on sekä erikoinen että mielenkiintoinen. WAIS oli useita vuosia laajassa käytössä kutakuinkin ainoana tietokantaohjelmana Gopher- ja WWW-palvelimissa. Nykyisin WAIS on kaupallinen ohjelma. Siitä on kuitenkin edelleen olemassa muutamia julkisversioita (freeWAIS), joita voi etsiä Internetistä. Yksi parhaista julkisversioista on saatavissa Dortmundista.[2]

WAIS:n yhteyskäytäntö perustui vain osittain Z39.50-1988 -standardiin. Läheskään kaikkia standardin piirteitä ei käytetty hyväksi mutta toisaalta WAIS:ssa on piirteitä, joita Z39.50-1988 ei tunne. Niistä mainittakoon esimerkkinä relevanssipalautteet [10, s. 128–129]. Niiden ideana on rakentaa uudet kyselyt niin, että lähtötietoina käytetään aiemmin haettuja, relevanteiksi havaittuja dokumentteja tai niiden osia.

WAIS:n merkittävin ominaispiirre on tilattomuus, joka ilmenee siinä, että WAIS ei lainkaan tallenna etsintöjen tuloksena syntyviä hakujoukkoja. Tilattomuus on hyvin tärkeä ominaisuus silloin, kun käyttäjiä on paljon (ja varsinkin, jos käyttäjät eivät maksa saamastaan palvelusta) ja tallennuskapasiteettia (hakujoukkojen käsittelyyn) on rajallisesti.

Z39.50-1992 Standardin seuraava versio (1992) korvasi täysin vuoden 1988 version mutta ei ollut sen kanssa yhteensopiva. Uusi versio suunniteltiin mahdollisimman tarkasti yhteensopivaksi vastaavan ISO:n standardin kanssa (ISO 10162/10163, *Search and Retrieve*, SR), joka valmistui vuonna 1991. Käytännössä Z39.50-1992 on SR:n yhteensopiva laajennus. Se on siis määrittelyltään laajempi mutta sisältää SR:n kokonaisuudessaan.

Z39.50-1995 Standardin uusimman version lukuisista luonnoksista käytettiin nimitystä Z39.50-1994. Uusimman version (1995) lopullisesti valmistuttua, luonnokset ovat vanhentuneet. Niitä on kuitenkin levinnyt laajalle erilaisina kopioina, joten luonnoksiin saattaa joskus törmätä. Uusimman version tärkein uusi piirre on EXPLAIN-palvelu, jota käytetään tietokantojen paikallistamiseen ja tietokantojen antamien palvelujen esittelemiseen. Kyseessä on siten metatason palvelu: tietoa tiedosta.

[2] `http://ls6-www.informatik.uni-dortmund.de/ir/projects/freeWAIS-sf`.

On tärkeää erottaa *standardin* versio ja standardin määrittelemän *protokollan* (yhteyskäytännön) versio toisistaan. Standardin versio Z39.50-1988 ei tue millään versionumerolla nimettyä protokollaa. Kyseisen standardin pohjalta ei vielä tehty toimivia ohjelmia. Kysymys protokollan versionumeroista nousi esiin siinä vaiheessa, kun käytössä alkoi olla ohjelmia, joista osa perustui ANSI/NISO:n standardiin, osa ISO:n standardiin. Jotta nämä, samoin kuin myöhemmin syntyvät ohjelmat kykenisivät menestykselliseen vuoropuheluun, keskustelun aluksi on voitava sopia käytettävän protokollan tasosta.

Standardin versio Z39.50-1992 määrittelee varsinaisesti protokollan version 2, mutta samalla se määrittelee myös protokollan version 1 (jota ei siis oikeasti ole olemassakaan). Tämä muodollinen järjestely on tehty yhteensopivuussyistä ISO:n vastaavan standardin kanssa. Standardin versio 3 määrittelee protokollasta sekä version 2 että version 3.

Z39.50:n, joka on siis amerikkalainen ANSI standardi, ja vastaavan ISO:n standardin *International Standards Organization* suhde on osoittautunut monimutkaiseksi. ISO on kehittänyt omaa standardiaan hajautettujen tietokantojen käsittelyyn yhtä kauan kuin ANSI/NISO omaansa. ISO:n standardointityö on kuitenkin hyvin hidasta, raskasta ja kallista. Lisäksi ISO:n standardit ovat monissa tapauksissa osoittautuneet mahdottomiksi toteuttaa käytännössä ja niinpä niihin käytetyt mittavat investoinnit ovat suurelta osin menneet hukkaan. Näin on käynyt erityisesti monille ISO:n OSI-mallin mukaisille tietoliikennestandardeille, joiden asemasta käytetään Internetin tosiasiallisia standardeja (mm. TCP/IP).

Viimeisin vaihe kehityksessä Z39.50:n osalta on, että ISO on luopunut oman SR-standardin jatkokehittelystä. Syksyn 1996 aikana käydyssä äänestyksessä Z39.50-1995 on hyväksytty täysin muuttamattomana (*verbatim*) ISO:n standardiksi ISO 23950. Standardin koodi on hauskasti valittu, numero ‘2’ muistuttaa nopeasti silmättynä kirjainta ‘Z’. Myös ISO:n standardin nimi on täsmälleen sama kuin Z39.50-1995:n virallinen nimi.

Z39.50 on lyönyt itsensä niin sanotusti läpi kirjastomaailmassa. Kaikista tärkeimmistä ulkomaisista kirjastojärjestelmistä on joko saatavana tai ainakin tekeillä tätä standardia käyttävä versio. Myös eräät suomalaiset kirjastojärjestelmiä tekevät ohjelmistotalot ovat alkaneet tehdä ohjelmiinsa standardin edellyttämiä muutoksia. Kirjastojen ulkopuolella Z39.50 on tulossa käyttöön informaatiopalvelujen paikannuksessa ja julkisten asiakirjojen jakelussa (ks. USA:n GILS-palvelu kohdassa 6.1 sivulla 70). Myös EU:n keskeisimmissä WWW-palvelimissa Z39.50 ollaan ottamassa käyttöön.

4.4 Z39.50:n palvelut ja osapalvelut

Z39.50-standardi ryhmittelee siinä määritellyn toiminnallisuuden palveluihin (*facility*) ja osapalveluihin (*service*). Palveluja on kaikkiaan yksitoista. Suurin osa palveluista sisältää vain yhden osapalvelun. Osapalveluja on yhteensä kaksitoista. Tällainen ryhmittely ja nimityksillä leikkiminen voi tuntua turhanaikaiselta. Sillä on kuitenkin tulevaisuutta silmällä pitäen selvä tarkoitus: palvelut ja osapalvelut lisääntyvät standardin kehittyessä ja niiden mielekäs ryhmittely helpottaa toimintojen jäsentämistä ja ymmärtämistä.

4.4.1 Yhteyden muodostaminen

Yhteyden muodostamisessa johonkin tietokantapalvelimeen on lukuisia ongelmia, joita kaikkia ei suinkaan selvitetä ja ratkaista Z39.50-standardissa. Asiakasohjelman käyttäjän tai käyttöönottajan (asentajan) on esimerkiksi tiedettävä, mihin Internet-verkon tietokoneeseen (ja mihin porttiin kyseisessä tietokoneessa) yhteys kussakin tapauksessa on tarpeen muodostaa. Standardi ei kerro mitään siitä, mistä tällaisia tietoja saa tai miten tulisi menetellä tietojen saamiseksi. Esimerkiksi Willow-ohjelman mukana tulee yhteyden muodostamisessa tarvittavat osoitetiedot muutamaan amerikkalaiseen tietokantaan (esim. Washingtonin ja Madison-Wisconsinin yliopistot)[3]. Muita yhteystietoja joutuu etsimään WWW:n dokumenteista ja postituslistojen viesteistä.

Z39.50-standardin tuntema ja määrittelemä toiminta alkaa vasta siitä hetkestä, jolloin tietoliikenneyhteys asiakasohjelman ja palvelimen välille on jo muodostettu. Internet-verkossa yhteys muodostuu käytännössä kahden TCP-ohjelman välille. Z39.50 herää siis henkiin siten, että asiakasohjelma lähettää palvelimelle sovellustason yhteyspyynnön ja tämä yhteyspyyntö samoin kuin siihen myöhemmin tuleva vastaus kulkevat TCP/IP:n kuljettamissa “kirjekuorissa” (tästä kuriiripostista ks. kohta 3.1 sivulla 21).

Aloituspalvelu (*initialization facility*) sisältää yhden osapalvelun, joka on yhteyden muodostaminen (*init service*). Yhteyspyynnössä (*Initialize-Request*) on lukuisien vähemmän tärkeiden yksityiskohtien ohella kolme keskeistä tietoa: protokollan versiot, yhteydenpidon aikana halutut osapalvelut ja keskinäisessä viestinnässä käytettävien sanomien pituudet. Palveli-

[3] Willow on Washingtonin yliopistossa kehitetty julkisohjelma, jota ensisijaisesti käytetään paikallisten kirjastotietokantojen yhteydessä. Willow on kuitenkin samalla monipuolinen Z39.50-asiakasohjelma.

men lähettämässä vastauksessa (*InitializeResponse*) kerrotaan, onko yhteys muodostettavissa asiakasohjelman toivomilla ehdoilla.

Protokollan (yhteyskäytännön) versiot ilmaisevat palvelimelle, mitä kaikkia protokollan versioita asiakasohjelma tuntee. Jos asiakas ymmärtää protokollan versiota 3, se pakostakin ymmärtää myös versiota 2, joka sisältyy kolmoseen. Määritelmän mukaan versio 2 on identtinen ykkösen kanssa, joten tässä tapauksessa asiakas hallitsee kaikki toistaiseksi määritellyt protokollan vesiot. Kun palvelin vastaa yhteydenmuodostukseen, se vastavuoroisesti ilmaisee kaikki tuntemansa protokollan versiot. Suurin yhteinen protokollan versio on sitten se, jota keskustelussa tullaan käyttämään. Uusin versio on aina ilmaisuvoimaisin ja sen vuoksi viimeisintä kummankin tunnistamaa versiota on tarkoituksenmukaisinta käyttää. Jos yhteistä kieltä ei jostain syystä lainkaan löydy, palvelin hylkää yhteyspyynnön.

Yhteyspyynnössä asiakas luettelee tuntemiensa protokollaversioiden ohella kaikki ne osapalvelut, joita se tulee palvelimelta pyytämään. Tärkein osapalveluista on luonnollisesti etsintä (search) tietokannasta. Palvelin kertoo vastaussanomassaan, mitä kaikkia toivotuista osapalveluista se pystyy antamaan. Esimerkiksi tietueiden hakeminen hakujoukosta erillisenä toimintona ei ehkä ole mahdollista, jos kyseessä on WAIS-tyyppinen tilaton tietokantapalvelu. Osapalveluista sopiminen osana yhteydenmuodostamista on tärkeää senkin vuoksi, että palvelinten toiminnallisuus vaihtelee suuresti toteutusten laajuuden mukaan.

Viestinnässä käytettävien sanomien suositeltava pituus ja enimmäispituus on myös tarpeellista sopia etukäteen. Asiakasohjelman käytettävissä saattaa olla paljon vähemmän keskusmuistia kuin palvelimessa ja sen vuoksi erilaisiin tilanteisiin on pakko varautua. Palvelimen saattaa olla tarpeen esimerkiksi pilkkoa (segmentoida) pisimmät tietueet osiin ennen postitusta. Normaalisti palvelin kasaa kokonaisia tietueita yhteen sen verran, että sanoman tavanomainen, suositeltava koko (PreferredMessageSize) ei ylity. Jos jokin yksittäinen tietue on tätä pitempi mutta kuitenkin lyhyempi kuin sovittu enimmäispituus (ExceptionalRecordSize), palvelin varoittaa tästä etukäteen, jolloin asiakasohjelma osaa varautua tällaisen poikkeuksellisen pitkän tietueen vastaanottamiseen. Jos tietue on tätäkin pitempi, sitä ei voi välittää lainkaan, ellei segmentoinnista ole sovittu. Jos segmentointi on tarpeen, kummankin osapuolen on hallittava tietueen pilkkomisessa ja kokoamisessa käytettävät pelisäännöt.

Useimpien tuotantokäytössä olevien tietokantojen käyttöön tarvitaan käyttäjätunnus ja salasana. Nämä tiedot lähetetään yhteyspyynnön valinnan-

varaisessa (optionaalisessa) identifiointikentässä (IdAuthentication). Monet Z39.50-toteutukset eivät tue tätä piirrettä ja sen vuoksi yhteys saattaa jäädä muodostumatta.

Merkkivalikoimat ovat eurooppalaisissa kielissä tunnetusti hyvin ongelmallisia. Vaikka laajasti käytetty ISO Latin-1 (ISO 8859) sisältää hyvin suuren määrän erilaisilla diakriiteilla varustettuja kirjaimia, se kattaa silti vain pienen osan kaikista käytössä olevista eurooppalaisista (alkujaan latinalaisista) kirjaimista. Z39.50-1995 tarjoaa tämän ongelman hoitamiseksi neuvottelumenetelmän, jolla käytettävästä merkkivalikoimasta voidaan sopia. Asiakasohjelma voi sisällyttää yhteyspyyntöön eräänlaisen neuvottelutarjouksen (*negotiation record*), jossa se ehdottaa sopimusta sekä käytettävästä kielestä että merkkivalikoimasta. Palvelin voi hyväksyä tehdyn tarjouksen tai tehdä siihen vastaehdotuksen.

4.4.2 Tietokantakysely

Etsintäpalvelu (*search facility*) sisältää yhden osapalvelun, joka on luonnollisesti etsintä (*search service*). Etsintäpyynnön (*SearchRequest*) tärkein sisältö koostuu tietokantakyselystä ja yhdestä tai useammasta tietokannan nimestä. Pyynnössä annetaan palvelimelle lisäksi ohjeita siitä, mitä kaikkea etsinnän tuloksesta halutaan vastauksena (*SearchResponse*) jo paluupostissa. Hakujoukon tietueet voidaan nimittäin pyytää kahdella eri tavalla. Ne voidaan pyytää joko osana etsintäpyynnön vastausta tai sitten ne pyydetään erikseen hakupalvelua käyttämällä (ks. seuraava kohta).

Etsintäpyynnössä olevan kyselyn ajatellaan lähtökohtaisesti kohdistuvan tietokantaan, jossa on useita indeksejä. Oletetaan vaikka, että lainaaja haluaa tietoja kirjasta, jonka on kirjoittanut Tolstoi ja jonka nimenä on Sota ja rauha. Tekijänimiä sisältävästä indeksistä etsitään sanaa ‘Tolstoi’ ja nimekkeitä sisältävästä indeksistä sekä sanaa ‘sota’ että sanaa ‘rauha’. Kyselyssä oleva lauseke on tällöin jotain seuraavan kaltaista:

(author = ‘Tolstoi’) and (title = ‘sota’ and ‘rauha’)

Hakujoukon koko Siltä varalta, että hakujoukko joko kokonaan tai ainakin osittain halutaan nähtäväksi jo etsintäpyynnön vastauksessa, kerrotaan

palvelimelle, miten paljon tietueita voidaan ottaa vastaan. Tähän tarkoitukseen on käytettävissä kolme eri kokoa: pieni koko, suuri koko ja keskikoko (vähän niinkuin sukkahousuissa — ennen kuin oivallettiin, että yksi koko riittää vallan mainiosti).

Pienen joukon yläraja (SmallSetUpperBound) on suurimman sellaisen hakujoukon koko, joka voidaan ottaa paluupostissa vastaan kokonaisuudessaan. Jos paluupostissa ei haluta lainkaan hakujoukon tietueita vaan pelkästään tieto hakujoukon koosta, tämä pienen joukon yläraja tulee asettaa nollaksi. Kuitenkin, riippumatta siitä mikä tämän ylärajan koko on, vastauksen tulee aina mahtua niihin tilankäyttöä koskeviin raameihin, jotka yhteydenmuodostamisen aikana sovittiin.

Suuren joukon alarajan (LargeSetLowerBound) asettamisella kerrotaan, että jos hakujoukon koko ylittää tämän rajan, kaikki tietueet haetaan erikseen myöhemmin. Tietueiden hakemiseen käytetään luonnollisesti hakupalvelua. Jos hakujoukon koko siis ylittää tämän rajan, kyseessä on nimensä mukaisesti suuri joukko, ja sellaiseenhan tulee suhtautua asianmukaisella kunnioituksella.

Jos hakujoukon koko osoittautuu olevan suurempi kuin pienen joukon yläraja, mutta pienempi kuin suuren joukon alaraja, ollaan mitä ilmeisimmin tekemisissä keskisuuren hakujoukon kanssa. Tämän tilanteen varalta on käytettävissä keskikoon esittämisluku (MediumSizePresentNumber), joka kertoo montako tietuetta toivotaan nähtäväksi etsintäpyyntöön saatavassa vastauksessa.

Attribuuttijoukot Etsintälausekkeeseen liittyy hyvin tärkeänä osana tieto käytettävästä attribuuttijoukosta (AttributeSetId). Lausekkeeseen saattaa nimittäin indekseihin kohdistuvien etsintätermien lisäksi sisältyä operaattoreita (yhtäsuuruus, suurempi kuin), tieto termin sijainnista tietueen kentässä (kentän alussa, missä tahansa kohdassa), katkaisusta (oikealta, vasemmalta, kummastakin suunnasta), jne. Kaiken tämän informaation esittäminen yksikäsitteisellä tavalla on vaikeaa. Kun tietokannat ja tietokantaohjelmat ovat monenkirjavia, ja kun kuitenkin pyritään tarjoamaan mahdollisuus eri ohjelmien yhteiskäyttöön, on standardissa yritetty löytää keino, jolla tämä kirjavuus voitaisiin kiertää.

Ratkaisu perustuu attribuuttijoukon käyttöön. Attribuuttijoukko määrittelee jokaiselle etsintälausekkeessa olevalle indeksiä, operaattoria, sijaintia, jne. tarkoittavalle seikalle yksikäsitteisen numeerisen koodin. Tietoliikenteen välittämässä sanomassa ei siten missään kohdassa sanota, että termiä

‘Tolstoi’ pyydetään etsimään indeksistä *author*, vaan sama asia esitetään koodien avulla. Etsintätermi ‘Tolstoi’ esiintyy tietysti sellaisenaan, sitä ei ole tarpeen millään erityisellä tavalla koodata.

Z39.50 määrittelee viitetietokantojen käyttöä varten attribuuttijoukon *Bib-1*, joka on suunniteltu aivan erityisesti viitetietojen käsittelyyn. Kun asiakasohjelma lähettää etsintäpyynnössä lausekkeen, se ensinnäkin kertoo, että koodauksessa käytetään nimenomaan attribuuttijoukkoa *Bib-1* (se taas kerrotaan objektitunnisteen avulla, tästä ks. hiukan tuonnempana). Bib-1 sisältää 6 erilaista attribuuttityyppiä, jotka ovat käyttö, relaatio, sijainti, rakenne, katkaisu ja täydellisyys. Attribuuttityypit on numeroitu juoksevasti 1–6. Nimeke ja tekijä kuuluvat attribuuttityyppiin 1, eli *käyttö*. Nimekettä vastaa attribuutin arvo 4, ja tekijää attribuutin arvo 1003. Attribuutin *käyttö* erilaisia arvoja on kaikkiaan noin 100.

Bib-1:n käytössä ongelmina ovat asiakasohjelman puolella käyttäjän käsitemaailman kuvaaminen attribuuttijoukolle ja palvelimen puolella attribuuttijoukon kuvaaminen tietokannan käsitemallille. Jos tietokanta on hyvin erikoislaatuinen, sen kenttien nimille ei välttämättä löydy mielekkäitä vastineita Bib-1:n attribuuteista. EU:n rahoittamassa ONE-projektissa kävi ilmi, että ainoa kaikkia mukana olevia tietokantoja yhdistävä tekijä on nimekkeen yksittäinen sana. Kaikki muulla tavoin konstruoidut etsinnät epäonnistuivat ainakin jossakin tietokannassa.

Z39.50 määrittelee kaksi muutakin attribuuttijoukkoa. Ne ovat *Exp-1* tietokantojen esittelytiedoille ja *Ext-1* laajennettuja palveluja varten. Standardissa on lisäksi varattu attribuuttijoukkojen tunnistenumero CCL-, GILS- ja STAS-joukkoja varten. GILS käyttää siis omaa, kyseistä tarkoitusta varten erityisesti määriteltyä attribuuttijoukkoa.

Tietueen muoto Kyselyjen kohteena olevan tietokannan tietueet voivat sisältää lukemattomia alkioita (fyysisellä tasolla puhutaan tietueen kentistä) ja joissakin tapauksissa osa alkioista saattaa olla hyvinkin pitkiä. Tämän vuoksi on tärkeää kertoa kyselyssä, mitä alkioita tietueista missäkin tilanteessa halutaan nähtäväksi. Kirjastonhoitajille ovat varmasti tuttuja viitetietokantojen erilaiset formaatit, joilla poimitaan osa tietueiden alkioista ja muutenkin vaikutetaan näytettävien tietojen ulkoasuun. Tästä aiheesta hiukan lisää jäljempänä hakupalvelun yhteydessä.

Objektitunnisteet Sekä attribuuttijoukot että tietueformaatit (kuten monet muutkin standardin osat) ovat sellaisia, että niiden tarkka sisältö riip-

puu suuresti käytetystä sovelluksesta. Sen vuoksi on oletettavissa, että jo tehdyt määritykset muuttuvat ja uusia määrityksiä tarvitaan paljonkin. Standardissa onkin varauduttu siihen, että kumpiakin määritellään lisää ja itse standardiin on kirjattu ainoastaan määritysten tunnistamisessa käytettävän tunnisteen muodostamis- ja kirjaamistapa. Näitä tunnisteita kutsutaan objektitunnisteiksi (*object identifier*) ja esim. attribuuttijoukon *Bib-1* tunniste on 1.2.840.10003.3.1, jota luetaan seuraavasti: 1 = ISO, 2 = member-body, 840 = US, 10003 = ANSI-standard-Z39.50, 3 = attribute set definition ja 1 = ATR.1 eli Attribute Set Bib-1. Standardista löytää kaikki sen valmistumiseen mennessä määritellyt objektitunnisteet, uusimmat tunnisteet saa Yhdysvaltain kongressin kirjastosta, joka on virallinen standardin säilytyspaikka (Z39.50 Maintenance Agency).

Etsintälauseke Itse kysely voidaan muodostaa usealla eri tavalla. Standardin uusin versio tuntee tyypit 0, 1, 2, 100, 101 ja 102. Kyselyn ja erityisesti siihen sisältyvän lausekkeen muotojen runsaudella halutaan varmistaa vähintään yhden yhteisen ja kaikkien osapuolten tunnistaman muodon käytettävyys ja sen lisäksi toimittajakohtaisten ratkaisujen ja aikaisempien standardien määrittelemien muotojen käytettömahdollisuus.

Kyselyn tyyppi 0 on varattu ohjelmistokohtaisten (natiivien) kyselyjen käytölle. Kun sekä asiakas että palvelin ovat saman ohjelmistotoimittajan tuottamia (esim. VTLS) ja kun nämä keskustelukumppanit toteavat yhteydenmuodostuksen aikana tämän yhteisen alkuperän, on mahdollista, että keskustelu kyselyjen osalta käydään VTLS:n kaikkia erityispiirteitä hyödyntämällä. Asiakas lähettää kyselyn VTLS:n omassa erityisformaatissa ja ilmaisee tämän asettamalla kyselyn tyypiksi 0:n. Kun palvelin puolestaan havaitsee, että kysely on tyyppiä 0, se empimättä olettaa, että kysely on tehty sen omaa erityistä kyselykieltä (palvelimen äidinkieltä) käyttäen.

Kyselyn tyyppi 1 muodostaa Z39.50:n varsinaisen ytimen ja se on tyyppi, jota jokaisen Z39.50-ohjelman on pystyttävä tulkitsemaan. Kyselyn lauseke esitetään ns. käänteisessä puolalaisessa muodossa (RPN, Reverse Polish Notation). Sen ideana on, että etsintätermien ja operaattorien käsittely- ja suoritusjärjestys ilmaistaan yksikäsitteisellä, joskin hiukan eriskummallisella tavalla. RPN on helpoimmin havainnollistettavissa aritmeettisen lausekkeen avulla. Esimerkiksi lauseke $((3+a)*(c/d)+k)$ esitetään RPN-muodossa $((3a+)(cd/)*)k+$. Aivan vastaavan logiikan mukaisesti muuntuu lauseke, joka koostuu termeistä ja Boolen operaattoreista. Tässä on syytä korostaa, että kyse on sovellustason protokollan määrittele-

mästä kuvaustavasta. On kokonaan eri asia, mitä käyttäjä käytännössä joutuu kirjoittamaan asiakasohjelman kyselylomakkeeseen. Lausekkeen tavanomainen esitystapa on ohjelmallisesti hyvin helposti muunnettavissa RPN-muotoon. Tämä muunnos itse asiassa tulee samalla tarkastaneeksi, onko lauseke oikein muodostettu, eli onko se ylipäätään yksikäsitteisellä tavalla suoritettavissa.

Kyselyn tyyppi 2 on ISO 8777:n muotoinen, eli ISO:n standardin määrittelemä CCL-kielinen muoto (Common Command Language). CCL-kielestä on myös ANSI-standardin Z39.58 määrittelemä muoto, joka poikkeaa jossakin määrin ISO:n CCL-kielestä. Tätä ANSI:n CCL-kieltä varten on **kyselyn tyyppi 100**.

Kyselyn tyyppi 101 on nykyisin sama kuin tyyppi 1. Tyypin 101 kysely määriteltiin standardin versiossa 2. Siinä voi käyttää läheisyysoperaattoreita ja hakujoukkoa voi etsinnän suorittamisen jälkeen rajoittaa. Standardin versiossa 3 nämä uudet piirteet on lisätty kyselyn tyyppiin 1 ja siten tyypit ovat nyt identtiset.

Kyselyn tyyppi 102 on Ranked List -kysely. Standardin versiossa 3 mainitaan, että tämä uusi kyselyn tyyppi tullaan määrittelemään standardin myöhemmässä versiossa. Tätä tyyppiä tullaan käyttämään lähinnä kokotekstitietokannoissa eikä se tule olemaan relevantti viitetietokantojen yhteydessä.

Etsinnän tulos Edellä on pitkästi ja monia teknisen tuntuisia yksityiskohtia sisältävästi selostettu etsintäpyyntöä. Etsinnän vastaus *SearchResponse* on paljon lyhyemmin selitetty. Vastaus sisältää lyhimmillään tiedon etsinnän onnistumisesta ja hakujoukon koon. Jos vastauksen mukana on pyydetty tietueita, vastauksessa kerrotaan, montako tietuetta siinä on, itse tietueet (toivotussa muodossa — toivottavasti) sekä tieto siitä, mistä kohtaa hakujoukossa tietueet jatkuvat (NextResultSetPosition) myöhemmin lähetettävän hakupyynnön varalta.

4.4.3 Tietueiden haku

Hakupalvelu (*Retrieval Facility*) koostuu kahdesta osapalvelusta, jotka ovat tietueiden esittäminen (*Present Service*) ja pitkien vastausten segmentointi (*Segment Service*). Osapalvelut esitellään tässä jokseenkin suppeasti, sillä niihin ei liity sen ihmeempiä yleiskuvan saamisen kannalta oleellisia asioita.

Tietokantakyselyn tuloksena syntyy siis hakujoukko, joka koostuu niiden tietueiden osoitteista, jotka vastaavat etsintälauseketta. Jotta hakujoukosta voitaisiin hakupalvelun avulla hakea tietueita, palvelimen täytyy säilyttää hakujoukko tallessa. Tämä on hakupalvelun käytön välttämätön edellytys. Kaikissa palvelimissa se ei kuitenkaan ole käytettävissä. Niissä tietueet on palautettava osana etsintäpyynnön vastausta.

Palvelimella saattaa olla väliaikaisesti tallessa lukemattomia hakujoukkoja ja sen vuoksi hakujoukon tunniste (ResultSetId) on esityspyynnön (PresentRequest) pakollinen tieto. Muita pakollisia tietoja ovat pyydettävien tietueiden määrä ja aloituskohta hakujoukossa. Lisäksi kerrotaan, mitä elementtejä tietueista halutaan — aivan vastaavaan tapaan kuin etsintäpyynnössä — sekä toivomus tietueiden esitysformaatista. Muuta oleellista esittämispyynnössä ei ole. Esittämispyynnön vastaus (PresentResponse) on hyvin analoginen etsintäpyynnön vastauksen kanssa.

Kyselyssä voidaan siis määritellä mitä alkioita halutaan sisällytettävän haun tulokseen (ElementSetNames). Määrittely koostuu yhdestä tai useammasta elementtijoukon nimestä. Standardi vaatii, että palvelimen on tunnistettava vähintäänkin nimet *F*, joka tarkoittaa tietueen kaikkien alkioiden (elementtien) hakemista (Full), ja *B*, joka on tarkemmin määrittelemätön lyhyt tietueen muoto (Brief). Tämän enempää standardi ei kerro elementtijoukoista tai niiden nimistä.

Tietueen alkioiden valinnan lisäksi kyselyyn voidaan liittää toivomus halutusta tulostietueen formaatista (PreferredRecordSyntax). Tulos voidaan pyytää jossakin lukemattomista MARC-formaateista, mm. Finnmarc-formaattia varten on varattu tunnistenumero. Formaatti voi myös olla jokin Z39.50-standardissa määritellyistä. Yksinkertaisin niistä on SUTRS (*Simple Unstructured Text Record Syntax*), johon ei sisälly mitään sen kummempaa rakennetta. Kyseessä on pelkkä juokseva teksti, joka on pätkitty enintään 72 merkin riveiksi. Tarkin formaattimääritys liittyy tietokantojen esittelytietoihin, määritys on peräti 14 sivua hyvin teknisen näköistä kuvausta.

Standardi sanoo, että palvelimen on esitettävä tietueet asiakkaan toivomassa muodossa. Jos se ei ole mahdollista tietueet on esitettävä jossakin niistä muodoista, joista yhteydenmuodostamisen aikana on sovittu. Käytännössä on kuitenkin jouduttu toteamaan, että yhteisen formaatin löytäminen saattaa olla ongelmallista. Jos yhteistä formaattia ei löydy, palvelin saattaa jättää tietueet kokonaan lähettämättä.

Segmentointi on standardin kolmosversion uusi ja merkittävä osapalvelu. Se mahdollistaa suurten tietuemäärien tehokkaamman välittämisen ja kaikki kohtuuden rajat ylittävien pitkien tietueiden esittämisen. Standardin aikaisemmat versiot edellyttivät että vastaus sekä etsintä- että hakupyyntöön mahtuu yhteen postitukseen. Segmentointi tekee mahdolliseksi tilaa vievän vastauksen pilkkomisen useampaan postitukseen.

Tason 1 segmentointi koskee suuren tietuejoukon hakemista. Jos hakujoukon koko on esimerkiksi vaikka 10 000 tietuetta, on todennäköistä, että ne eivät mahdu niihin raameihin, jotka yhteydenmuodostuksen aikana on sovittu. Standardin versio 2:n mukaisesti toimittaessa tietueita jouduttaisiin hakemaan erillisillä pyynnöillä niin monta kertaa, että kaikki tulisivat haetuksi. Nopeissa verkoissa tämä moninkertainen, loogisesti tarpeeton hakeminen on omiaan hidastamaan toimintaa. Tason 1 segmentointi tekee mahdolliseksi sen, että palvelin lähettää kaikki tietueet erillisissä, perättäisissä paketeissa yhden ainoan hakupyynnön perusteella.

Tason 2 segmentointi koskee poikkeuksellisen pitkiä tietueita. Kyseessähän voi olla vaikkapa kokonaisen kirjan teksti kuvineen. Segmentointi koskee tässä siis yhtä yksittäistä tietuetta, jota ei voida kerralla lähettää esim. asiakkaan tietokoneen keskusmuistin rajallisuuden takia. Asiakasohjelma ottaa segmentoidun tietueen vastaan pala kerrallaan ja kokoaa sen jollekin tallennuslaitteelle eikä yritäkään pitää sitä kerralla muistissa.

4.4.4 Tietokantojen esittely

Tietokantojen esittely (*Explain Facility*) on palvelu, johon ei kuulu yhtään osapalvelua. Esittely hoidetaan kokonaan etsintä- ja hakupalvelujen avulla. Esittely tuli standardiin vasta vuoden 1995 versiossa mutta sitä voidaan silti pitää Z39.50:n keskeisimpänä piirteenä. Esittely luo nimittäin edellytykset informaatiopalvelujen paikallistamiselle (*locator service*). Kyse on siis palvelusta, jossa ensin etsitään ja paikallistetaan mielenkiintoisilta vaikuttavat tietokannat ja vasta sen jälkeen aletaan tutkia tietokantojen sisältöä. Palvelu tuli standardiin Yhdysvaltain liittovaltion monivuotisen projektin työn tuloksena. Esittelypalveluun perustuva järjestelmä on Yhdysvalloissa ollut käytössä vuodesta 1994 alkaen (järjestelmästä tarkemmin jäljempänä kohdassa 6.1 sivulla 70).

Tietokantojen esittelyn hyväksikäyttö ei edellytä asiakasohjelmalta välttämättä mitään lisäpiirteitä. Esittelytiedot löytyvät palvelimen tekstitietokannasta. Niitä etsitään ja haetaan samalla tavalla kuin mitä tahansa tieto-

kantojen tietoja. Jos asiakasohjelma pystyy hyödyntämään hyperlinkkejä tai muita siirtymiä dokumentista ja palvelusta toiseen, siirtyminen informaatiopalvelun paikallistamisesta tietokannan selaamiseen voi parhaimmillaan ollaan lähes huomaamaton.

Tietokantojen esittelyä tarjoavan palvelimen on sen sijaan varauduttava seuraavaan:

- etsimään esittelytietoja tietokannasta, jonka nimenä on *IR-Explain-1*.
- käyttämään attribuuttijoukkoa *exp-1*, joka on tarkoitettu nimenomaan esittelytietojen välittämiseen, ja
- käyttämään esittelytietojen viimeistelyssä syntaksia *Explain*, joka vastaavasti on tarkoitettu esittelytietojen asianmukaisen esitysmuodon tekemiseen.

4.4.5 Laajennettu palvelu

Laajennettu palvelu (*Extended Services Facility*) sisältää yhden osapalvelun, jolla on sama nimi kuin itse palvelulla. Laajennettua palvelua tarjoavassa palvelimessa on oltava tietokanta, jonka nimenä on (*IR-Extend-1*). Siihen on tallennettu eräänlaisia palvelupaketteja, joiden luonteesta voi saada selvyyden tietokannan esittelyn avulla.[4]

Standardi määrittelee ja rekisteröi yhteensä 7 laajennettua osapalvelua. Niistä tärkeimmät ovat aineiston tilaus (Item Order Extended Service) ja tietokannan päivitys (Database Update Extended Service). Aineiston tilaaminen voi liittyä esim. kirjastojen väliseen lainausjärjestelmään (ILL), jolloin pyyntö on kyseisen järjestelmän määrittelevän standardin (ISO 10161) edellyttämässä muodossa. Tilauspyyntöön voidaan tarvittaessa liittää tiedot toimitusosoitteesta ja laskuttamisesta. Tietokannan päivittämisessä käytettäviä toimintoja ovat tietueiden lisääminen, korvaaminen ja poistaminen sekä tietueiden sisältämien kenttien muuttaminen.

4.4.6 Yhteyden sulkeminen

Loogisesti tarkastellen yhteyden sulkeminen päättää aikanaan pyyntöjen ja vastausten muodostaman vuoropuhelun. Lopettamispalvelu (*Termination*

[4] Asiakas voi esimerkiksi pyytää, että suoritapa nyt seuraavaksi pizzan lähetyspalvelu (*pizza-on-demand*), joka on tallennettu tietokannan tietueeksi nro se-ja-se. Pyynnön attribuuteilla voidaan ilmaista halutut mausteet.

Facility) sisältää vain yhden osapalvelun, joka siis on yhteyden sulkeminen (*Close Service*). Kumpi tahansa voi sulkea yhteyden ja tietoliikenteeseenkin luontevasti kuuluva kohteliaisuus edellyttää, että vastapuoli ilmoittaa hyväksyvänsä lopetuspyynnön.

4.4.7 Muut toiminnot

Standardi määrittelee vielä huomattavan joukon muita palveluja ja niihin sisältyviä osapalveluja. Ne eivät toiminnan peruslogiikan kannalta ole kuitenkaan samassa mielessä oleellisia kuin edellä esitellyt. Mainitsen ne seuraavassa lyhyesti palveluittain.

Hakujoukon tuhoamispalvelu (*Result-set-delete Facility*) sisältää nimensä mukaisesti osapalvelun, jolla yksi tai useampia hakujoukkoja tuhotaan (*Delete Service*).

Selauspalvelu (*Browse Facility*) sisältää yhden osapalvelun, jolla voi selata (*Scan Service*) indeksistä esimerkiksi tekijänimiä, jostakin halutusta kirjailijan nimestä alkaen.

Lajittelupalvelussa (*Sort Facility*) asiakas voi pyytää palvelinta lajittelemaan hakujoukon tietueet johonkin tiettyyn järjestykseen.

Pääsyn valvonta (*Access Control Facility*) sisältää yhden osapalvelun, jolla on sama nimi kuin palvelulla. Palvelin voi pyytää asiakasta ilmaisemaan valtuutensa (tyyliin käyttäjätunnus ja salasana) tilanteessa, jossa asiakas yrittää hakea tietoja suojatusta tietokannasta. Yhteydenmuodostuksen aikana puolin ja toisin ilmaistaan tullaanko tällaista palvelua mahdollisesti käyttämään.

Resurssien kirjanpidon ja käytön valvonta (*Accounting/Resource Control Facility*) sisältää kolme osapalvelua, jotka ovat resurssien valvonta (*Resource-control service*), resurssien valvonnan aloittaminen (*Trigger-resource-control service*), ja resurssien käyttöraportti (*Resource-report service*). Myös resurssien valvonnan hyödyntämisestä voidaan sopia jo yhteydenmuodostuksen aikana.

Palvelin voi pyynnöstä seurata resurssien käyttöä ja lähettää villiintyneiksi osoittautuneiden kyselyjen aikana asiakkaalle tiedustelun, että nyt on mennyt CPU-aikaa jo tuhannen dollarin edestä, jatketaanko kyselyn toteuttamista. Huolestunut asiakas voi omasta puolestaan lähettää palvelimelle pyynnön resurssien käyttöä koskevan valvonnan aloittamisesta. Lisäksi asiakas voi pyytää raportin siihen mennessä käytetyistä resursseista.

4.5 Z39.50:n jatkokehitys

Ray Denenberg Yhdysvaltain kongressin kirjastosta on esitellyt standardin jatkokehitystä Z39.50-seminaarissa Belgiassa syyskuussa 1996 [5]. Jatkokehitys koskee standardin profiileja, kyselytyyppejä ja attribuutteja.

Standardin profiili liittyy ISO:n terminologiaan. Profiilin määrittelyllä tarkoitetaan standardin rajaamista ja avoimien kysymysten täsmentämistä jonkin tietyn sovelluksen tai aihepiirin tarpeisiin. Laajoja standardeja ei yleensä koskaan toteuteta täydellisenä. Niiden monenkirjavuutta on pakko rajoittaa, jotta saataisiin järjellisessä ajassa käyttökelpoisia ja toimivia ratkaisuja. Profiileja ei määritellä itse standardissa. Käyttäjäyhteisöt sopivat profiilista ja standardissa pelkästään luetellaan luodut profiilit. Tärkeimmät niistä ovat GILS ja WAIS.

Uusia profiileja on suunnitteilla useitakin. Denenberg mainitsee kokoelmat (*Collections Profile*), museoiden CIMI-profiilin (*The Consortium for the Interchange of Museum Information*), digitaaliset kirjastot (*Z39.50 Profile for Access to Digital Library Objects*), luetteloiden yhteiskäytön (*CIP Profile, the Catalogue Interoperability Protocol*) ja pari paikallista hanketta (Australian kansalliskirjasto ja Stanfordin digitaalinen kirjasto).

Uusista kyselytyypeistä tärkein on tyyppi 102 RLQ (*Ranked List Query*), joka on lähinnä tarkoitettu suurten kaupallisten palveluntarjoajien käyttöön. Näistä tarjoajista Denenberg mainitsee esim. Chemical Abstractin ja NLM:n (National Library of Medicine). Kyselyssä palvelimelle voi kertoa, miten tuloksia tai etsintälausekkeen osia painotetaan. Kyselyssä voi käyttää relevanssipalautetta. Etsinnän tarkentamiseen voi käyttää tesaurusta, jne.

Luku 5

SGML ja rakenteiset dokumentit

Dokumenttien pitkäaikaisessa hyväksikäytössä on välttämätöntä varautua sekä tietovälineiden että tietokoneohjelmien jatkuviin muutoksiin. Aineisto on aika-ajoin kopioitava uudempaa teknologiaa edustaville tietovälineille ja tämän lisäksi aina tarpeen tullen muunnettava kokonaan toisenlaiseen tallennusmuotoon. Mitä nopeammin teknologia kehittyy, sitä lyhytikäisempiä ovat tietovälineet ja osittain myös tallennusmuodot. Papyruskääröt ja savitaulut ovat säilyttäneet niihin “tallennetun” informaation käyttökelpoisessa muodossa vuosituhansia. On mahdollista, että tietotekniikka ei koskaan tule kehittämään tietovälinettä, joka olisi yhtä pitkäikäinen.

Tietojenkäsittelyssä käytettävien laitteiden osalta ongelman luonteesta saa hyvän mielikuvan, kun palauttaa mieliin mitä kaikkea tähän mennessä on jo ehtinyt tapahtua. Ensimmäiset tietovälineet olivat reikänauha (eri leveyksiä) ja reikäkortti (eri kokoja). Niiden käsittelyyn sopivia laitteita ei ole ollut käytössä enää pitkiin aikoihin. Seuraavaksi tulivat avokeloilla olevat magneettinauhat (useita eri tiheyksiä ja nauhaformaatteja). Niiden viimeinen sukupolvi on parhaillaan poistumassa käytöstä. Muutaman vuoden kuluttua ei enää mistään löydy magneettinauhoja lukevia laitteita. Nauhakaseteista on käytössä useita keskenään yhteensopimattomia malleja (ainakin TK, DLT, DAT, QIC ja Exabyte). Niistä TK ja 4 mm:n DAT-nauha poistunevat ensimmäisenä käytöstä. Myös vanhemmat tietolevyt ovat jo poistuneet käytöstä (floppy disk, kaksi eri kokoa).

Tietovälineiden vanheneminen on täysin väistämätön ongelma — se on osa teknologian vääjäämätöntä jatkuvaa kehityskulkua. Ongelma on kuitenkin käsitteellisessä mielessä helposti ymmärrettävissä ja käytännössä kohtuullisella työllä hallittavissa. Tallennusmuotoja koskeva ongelma on sen

sijaan hyvin erilainen luonteeltaan ja paljon vaikeammin lähestyttävissä. Ei ole mitenkään itsestään selvää, mihin seikkoihin pitäisi ensisijaisesti kiinnittää huomiota pitkäaikaisen käytettävyyden turvaamiseksi.

Ongelmaa on lähestytty kahdesta, lähes vastakkaisesta suunnasta. (1) Dokumentit tallennetaan niin, että ne säilyttävät mahdollisimman tarkasti paperidokumenttien ulkoasun. Taulukkojen, lomakkeiden ja kuvien osalta tämä lähestymistapa on helposti perusteltavissa. (2) Toinen lähestymistapa kiinnittää huomionsa dokumentin rakenteeseen ja sisältöön. Dokumentti tallennetaan niin, että sen yleinen rakenne ja eri rakenneosat pystytään tunnistamaan ja erottamaan. Raakateksti tallennetaan kummassakin ratkaisussa mahdollisuuksien mukaan sellaisenaan.

Dokumenttien ulkoasun kuvaamisessa voidaan käyttää ISO:n ODA-standardia (ISO 8613)[1]. ODA:n avulla voidaan kuvata tekstilohkojen ja kuvien sijainti paperilla, taulukkojen sijainti ja ulkoasu, jopa teksteissä käytetyt kirjasinlajikkeet. Standardia on sovellettu organisaatioiden välisessä määrämuotoisten dokumenttien käsittelyssä ja siirrossa. Lähestymistapa ei kuitenkaan ole riittävän monipuolinen dokumenttien pitkäaikaisen hyväksikäytön kannalta. Parempi ratkaisu on löydettävissä dokumenttien rakenteen kuvaamisesta.

Rakenteen kuvaamista varten on kehitetty ISO:n SGML-standardi (ISO 8879)[2]. Standardi on varsin laaja ja käsitteellisesti hiukan vaikeasti ymmärrettävissä. SGML on tavallaan metatason standardi. Se ei ole sellaisenaan vielä mikään merkintäkieli — nimestään huolimatta. SGML on *yleistetty* merkintäkieli, se on pikemminkin yleinen kielioppi, jonka avulla voidaan määritellä hyvinkin erilaisia merkintäkieliä. WWW:n dokumenttien koodaamisessa käytetty HTML on yksi SGML-pohjaisista merkintäkielistä. Yritän jäljempänä selventää tätä asetelmaa.

5.1 Pari sanaa tekstinkäsittelystä

Mutta palataanpa asiassa hiukan taaksepäin ja katsotaan mitä johtopäätöksiä tekstinkäsittelystä voitaisiin tämän ongelman osalta tehdä. Jokainen mikronkäyttäjä tuntee jonkin tekstikäsittelyohjelman. Useimmiten käytössä on

[1]ISO 8613, Information technology — *Open Document Architecture* (ODA) and interchange format.

[2]ISO 8879, Information processing – Text and office systems — *Standard Generalized Markup Language* (SGML).

joko WP tai Word. Matemaatikot ja Unix-harrastajat käyttävät sellaisia ohjelmia kuten TEXja LATEX. Valtion virkamiehet ja muut vähäosaisemmat joutuivat aikaisemmin tyytymään VTKK:n myymään TEKO-ohjelmaan. Useimmille on tullut tutuksi myös tekstien siirtämisen hankaluus ohjelmasta toiseen. Jokainen tekstinkäsittelyohjelma käyttää juuri sille ominaisia merkintätapoja muotoilutietojen esittämiseen. Merkintätavat voivat muuttua ohjelmaversiosta toiseen eikä mitään yhtenäistymistä ole edes näköpiirissä.

Yksittäisten dokumenttien ja pienten tekstimäärien käsittelyssä tekstin muotoilussa tai rakenteen kuvaamisessa käytettävillä merkintätavoilla ei ole kovinkaan suurta merkitystä. Jos ohjelma on syystä tai toisesta vaihdettava, tekstidokumentin voi aina välillä muuntaa raakatekstiksi (ASCII-muotoinen teksti) ja muotoilla toisella ohjelmalla uudelleen.

Jos dokumentteja on kymmeniä tuhansia tai enemmän ja jos dokumenttien tuotantoon ja käsittelyyn liittyy erilaisia jalostusketjuja automatisoituine välivaiheineen, ongelma muuttuu monin verroin vaikeammaksi. Suurten tekstimäärien tallentamiseen liittyy vielä usein perusteltu vaatimus dokumenttien pitkäaikaisesta käytettävyydestä. Yhdenkään nykyisen tekstinkäsittelyohjelman pitkäikäisyyteen (ehkä TEX-ohjelmaa lukuunottamatta) ei voi luottaa.

Ongelma voidaan esittää vielä hiukan vaativammassa sävyssä. Eri alojen tutkijat ovat siirtämässä tietokoneella käsiteltävään muotoon papyruskääröillä, savitauluissa ja muissa vanhoissa dokumenteissa olevaa aineistoa. Aineiston siirtäminen vaatii paljon työtä. Sen vuoksi tutkijat haluavat tehdä sen vain yhden kerran. He haluavat, että aineisto olisi käytettävissä ilman uudelleen koodausta vuosituhansia eteenpäin. On itsestään selvää, että mikään WP:n tai Wordin käyttämä tallennusmuoto ei tähän tarkoitukseen sovellu. Itse asiassa käytettävän tallennusmuodon on välttämättä oltava sellainen, että mainittuja ohjelmia tekevät kaupalliset yritykset eivät pysty estämään tai haittaamaan pitkäaikaista käyttöä ohjelmien käyttämiä tallennusmuotoja muuttelemalla.

SGML-standardi on luotu juuri tällaista käyttötarvetta silmällä pitäen. Tärkeimpänä tavoitteena on ollut sellaisen kuvausjärjestelmän kehittäminen, joka tekee dokumenteista rakenteellisesti ja sisällöllisesti mahdollisimman pitkäikäisiä — mahdollisesti ulkoasun kustannuksella. SGML-muotoinen (s.o. SGML-standardin mukaisella merkintäkielellä koodattu) dokumentti voidaan siirtää periaatteessa mistä tahansa ohjelmasta/koneesta mihin tahansa ohjelmaan/koneeseen. SGML-muotoiset dokumentit ovat immuuneja ohjelmisto- ja teknologiamuutoksille.

Tämä ei suinkaan tarkoita sitä, että käyttäjä voisi WP:n tai Wordin avulla hyödyntää SGML:n mahdollisuuksia. Käytössä olevan ohjelmiston on riittävässä määrin tuettava SGML-pohjaisia toimintoja. Tietokantaohjelma voi olla sellainen, että dokumenttien rakenteiden mukaiset kyselyt ovat mahdollisia. Käytössä olevan tekstinkäsittelyohjelman on pystyttävä vaivattomalla tavalla tuottamaan haluttuja merkintöjä. Ohjelman on pysytyttävä tarkistamaan, onko jokin dokumentti sille tarkoitetun rakennemäärittelyn mukainen. Lisäksi tarvitaan muunnoksia käytössä olevasta tallennusmuodosta erilaisiin siirto- ja esitysmuotoihin (painettavat julkaisut, kuvaruudulla luettava teksti) sekä toisenlaisiin tallennusmuotoihin. Kaikki nämä muunnokset on voitava tehdä ohjelmien avulla automaattisesti.

5.2 SGML:n historia

SGML on ISO:n standardi. Se on vuosikymmenien työn tulos. SGML:n taustana on 1960-luvulla kehitetty GML (*Generalized Markup Language*). Sekä GML:n että SGML:n takana on pitkälti yksi henkilö, CHARLES GOLDFARB. Hän oli standardin kehittämisen ajan IBM:n palveluksessa ja SGML on tavallaan hänen elämäntyönsä. Yhden henkilön pitkäaikainen ja määrätietoinen paneutuminen tähän työhön on varmistanut standardin yhtenäisyyden ja erityisesti sitä koskevien dokumenttien korkean tason. SGML ei ole sekalainen kokoelma väkinäisiä kompromisseja, kuten on laita monien muiden standardien kohdalla.

Charles Goldfarb, Edward Mosher ja Raymond Lorie julkaisivat GML:n IBM:n julkaisuna vuonna 1969. GML:n pohjalta kehitystyötä jatkettiin ensin ANSI-standardin luomiseksi ja myöhemmin ISO:n standardin luomiseksi. Standardin kehittäminen on vaatinut useita vuosia aikaa ja suunnattomasti kärsivällisyyttä. Charles Goldfarb on vastannut kaikkien näiden vuosien ajan standardin editoinnista. Virallinen standardi on vuodelta 1986 ja sen täydellinen nimi on “Information processing — Standard Generalized Markup Language (SGML)”. Standardia on hiukan täydennetty vuonna 1988, lisäyksen nimi on “Amendment 1, ISO 8879/A1:1988”.

SGML-standardin voi ostaa ISO:n normaalijakelun kautta. Huomattavasti parempi tapa todella perehtyä tähän standardiin on hankkia Goldfarbin ainutlaatuinen kirja *The SGML Handbook* [7]. Kirja sisältää SGML:n historian ja hyvän johdatuksen SGML:n käsitteisiin ja piirteisiin. Tärkeimmän osan kirjasta vie itse standardi ja siihen liittyvät Goldfarbin lukuisat selityk-

set. Kirja kokonaisuutena on kuin hyperteksti — se on täynnä viittauksia kirjan muihin osiin.

Lyhyemmän ja käytännön läheisemmän johdatuksen SGML:ään tarjoaa vaikkapa ERIK VAN HERWIJNENIN *Practical SGML* [8]. Herwijnen kirjoitti kirjan ollessaan CERNin palveluksessa, jossa hänen työnsä liittyi HTML:n kehittämiseen. SGML:stä ja siihen liittyvistä muista standardeista on olemassa erittäin runsaasti kirjallisuutta.

5.3 SGML:n rakenne ja käyttö

SGML-muotoinen dokumentti koostuu *kolmesta osasta*: SGML-esittelystä *(SGML Declaration)*, dokumentin tyyppimäärittelystä *(Document Type Definition)* ja itse dokumentista, jota kutsutaan dokumentin esiintymäksi *(Document Instance)*. Kaikki kolme osaa voivat olla samassa tiedostossa peräkkäin. Tavallisin tilanne on kuitenkin se, että SGML-esittelynä käytetään ohjelmaan sisäänrakennettua oletusesittelyä eikä käyttäjä koskaan sitä edes näe. Dokumenttien tyyppimäärittelyjä voi olla käytössä useitakin mutta nekin ovat yleensä erillisissä tiedostoissa. Niitä tarvitaan vain, jos dokumentin koodauksessa tarvitaan yksityiskohtaista tietoa koodien muodosta tai kieliopillisista seikoista. Useimmiten käyttäjä kirjoittaa jollakin tekstinkäsittelyohjelmalla pelkän dokumentin esiintymän (jatkossa käytän pelkästään nimitystä dokumentti, jolla asiayhteydestä riippuen tarkoitetaan dokumentin alkuperäistä muotoa tai SGML-muotoista, koodattua esitystä).

5.3.1 Esimerkki

Aluksi lyhyt esimerkki SGML-koodatusta dokumentista. Sitä käytetään hyväksi myöhemmin standardin tärkeimpien piirteiden esittelyssä.

```
<!doctype article system "document.dtd">

<article opts="twoside">
<titlepag>
<title>Esimerkki dokumentin koodaamisesta
<author>
<name>Ville Virtanen
<inst>Dokumenttien hallinta Oy
```

```
</author>
<date>30.6.1994
</titlepag>

<abstract>
Dokumentti voi sisältää tiivistelmän, joka useimmiten
ladotaan hiukan eri tavalla kuin muu teksti.
</abstract>

<sect><heading>Dokumentin ensimmäinen luku</heading>
<p>
Varsinainen tekstiosa dokumentista voidaan jakaa monella
tavalla osiin. Dokumentti voi koostua luvuista, jotka
koostuvat aliluvuista, jotka koostuvat jaksoista jne.
Tässä esimerkkitapauksessa jaotuksessa käytettävät nimet
ovat tyyppiä sect, sect1, sect2, jne.

<p>
Kappaleet voi erotella tyhjällä rivillä.
</sect>
</article>
```

Esimerkissä on suhteettoman paljon koodeja varsinaisen tekstin määrään verrattuna. Normaalisti dokumentin sisällöstä valtaosa on tavallista raakatekstiä, joka sisältää hyvin vähän SGML-koodeja.

Esimerkki on siinä mielessä täydellinen, että se voidaan käsitellä SGML-ohjelmilla kunhan käytettävissä on vain sopiva dokumenttityypin määrittelytiedosto, tässä tapauksessa `document.dtd`. Päällisin puolin esimerkki muistuttaa HTML-koodattuja WWW:n dokumentteja. Syy tähän on hyvin yksinkertainen, HTML on SGML-pohjainen merkintäkieli ja kaikki SGML-pohjaiset merkintäkielet muistuttavat ulkoasunsa puolesta toisiaan.

5.3.2 SGML-esittely

SGML-esittelyn tärkeimpänä tarkoituksena on määritellä käytettävä merkkivalikoima (tavallisimmin ISO Latin 1, eli ISO 8859-1) ja koodauksessa käytettävät tärkeimmät erikoismerkit. SGML-koodaus perustuu erityisten merkintäkoodien (tagien) eli erottimien käyttöön. Jokaisen tekstielementin

alussa on alkuerotin, lopussa on mahdollisesti loppuerotin. Esimerkiksi henkilön nimi voidaan dokumentissa ilmaista muodossa:

```
<name>Ville Virtanen</name>
```

Alkuerotin on `<name>` ja loppuerotin `</name>`. Kulmasulut ja kauttaviiva ovat tässä tapauksessa tärkeimmät koodauksessa käytettävät erikoismerkit. Ne ovat itse asiassa samat kuin varsinaisessa standardissa. Niitä ei ole kuitenkaan pakko käyttää. Jos on aivan erityisiä syitä, nekin merkit voidaan vaihtaa joksikin muiksi. Uudet koodauksessa käytettävät erikoismerkit esitellään SGML-esittelyssä.

Esittelyssä hoidetaan myös eri tietokoneiden käyttämien erilaisten merkistöjen eroista aiheutuvat ongelmat. Tietokoneet eivät käsittele näkyviä merkkejä (esim. kirjainmerkki ‘A’) vaan merkkejä vastaavia numeroarvoja. Kirjainmerkkiä ‘A’ vastaava numeroarvo ASCII-koodissa on 65, IBM:n tietokoneiden käyttämässä EBCDIC-koodissa 193 (kumpikin luku on ilmaistu 10-järjestelmässä). Numeroarvojen ohella merkistöt poikkeavat toisistaan myös merkkivalikoimien osalta. Esimerkiksi ASCII-merkit on alunperin suunniteltu pelkästään englannin kielen tarpeiden mukaan. Siitä puuttuvat kokonaan mm. skandinaaviset kirjaimet. ISO Latin 1 ne sen sijaan sisältää.

5.3.3 Dokumentin tyyppimäärittely

SGML:n käyttöönottamisessa tärkein ja eniten työtä vaativa tehtävä on dokumentin tyyppimäärittelyn tekeminen — ellei sitten jokin olemassaolevista tyyppimäärittelyistä kelpaa sellaisenaan. Tavalliselle tekstin kirjoittajalle tyyppimäärittely on abstraktin ja oudon tuntuinen asia. Sen rooli dokumentin rakenteen kuvaamisessa on kuitenkin aivan oleellinen ja sen vuoksi perusideat on syytä opetella ymmärtämään.

Dokumentin tyyppimäärittely (DTD, *Document Type Definition*) on SGML-standardin mukaisesti laadittu muodollinen kuvaus dokumenttien rakenteesta. Tyyppimäärittely laaditaan yleensä niin, että se kattaa johonkin tiettyyn käyttötarkoitukseen kuuluvat tai jonkin organisaation käyttämät dokumentit. Yhden organisaation dokumenttityyppejä voivat olla esimerkiksi pöytäkirjat, muistiot, selvitykset, pysyväismääräykset, ohjeet ja tiedotteet. Jokaisen dokumenttityypin mukaisia dokumentteja on vaihteva määrä ja dokumentit voivat keskenään olla tietyissä rajoissa vaihtelevan näköisiä.

Dokumentin tyyppimäärittely on eräänlainen formaali kielioppi. Kun jokin dokumentti on koodattu tämän tyyppimäärityksen edellyttämällä tavalla, tietokoneella voidaan yksikäsitteisesti tarkistaa, onko dokumentti todella kyseisen tyyppimäärityksen mukainen. Ohjelma lukee ensin muistiin tyyppimäärityksen ja sen jälkeen käy dokumentin yksityiskohtaisesti läpi. Ohjelma havaitsee, jos dokumentissa on sellaisia rakenneosia, joita tyyppimääritys ei tunne. Ohjelma havaitsee myös, jos dokumentista puuttuu jokin osa, joka on tyyppimäärityksessä määritelty pakolliseksi tai jos rakenneosat esiintyvät väärässä järjestyksessä.

Dokumentin tyyppimäärityksessä määritellään:

- dokumentin rakenneosien eli elementtien nimet,
- kuinka usein elementit voivat esiintyä dokumentissa,
- missä järjestyksessä elementtien on oltava,
- ovatko elementtien alku- ja loppuerottimet pakollisia, vai voidaanko jompi kumpi niistä tai molemmat tunnistaa välillisesti,
- jokaisen elementin sisältö, joka voi koostua toisista elementeistä ja lopulta varsinaisesta tekstistä,
- attribuutit ja niiden oletusarvot, joiden avulla täsmennetään elementtien luonnetta,
- entiteetit, joilla viitataan tyyppimäärityksen sisäisiin rakenteisiin tai tyyppimäärityksen ulkopuolella oleviin, ohjelmien hyvin tuntemiin “tosiasioihin” ja
- dokumentin kirjoittamisessa käytettävät säännöt, kuten kappalejaon tunnistaminen tyhjän rivin perusteella ilman alku- ja loppuerottimien käyttöä.

Yksinkertainen artikkeleita koskeva dokumentin tyyppimäärittely (`document.dtd`) näyttää seuraavalta:

```
<!DOCTYPE dokumentti [
<!ELEMENT article        - -
         (titlepag, abstract?,
          toc?, p*, sect+,
          (appendix, sect+)?, biblio?) +(footnote)      >
```

```
<!ATTLIST article
          id    ID              #IMPLIED
          opts  CDATA           "null"                  >
<!ELEMENT titlepag       - - (title, author, date?)     >
<!ELEMENT title          - O (#PCDATA)                  >
<!ELEMENT author         - O (name, inst?,
                              (and, name, inst?)*)      >
<!ELEMENT name           O O (#PCDATA)                  >
<!ELEMENT and            - O (#PCDATA)                  >
<!ELEMENT inst           - O (#PCDATA)                  >
<!ELEMENT date           - O (#PCDATA)                  >
<!ENTITY % sect "heading, p* "                          >
<!ELEMENT heading        - - (#PCDATA)                  >
<!ELEMENT sect           - O (%sect, (p+ | sect1*))     >
<!ELEMENT sect1          - O (%sect, (p+ | sect2*))     >
<!ELEMENT sect2          - O (%sect, p+)                >
<!ELEMENT p              - - (#PCDATA)                  >
]>
```

Tyyppimäärittely ei ole aivan täydellinen. Se antanee kuitenkin riittävän hyvän mielikuvan asiasta. Tyyppimäärittely on tarkoitettu aiemmin esitetyn esimerkin mukaisen dokumentin määritykseksi. Käyn seuraavassa pääpiirteissään läpi esimerkkidokumentin rinnastamalla sitä tyyppimääritykseen.

5.3.4 SGML-koodattu dokumentti

Dokumentin alussa kerrotaan, että käsillä oleva dokumentti on tyyppiä `article` ja että siinä käytettävät merkinnät perustuvat tyyppimääritykseen `document.dtd` (lyhyesti DTD).

Artikkeli kokonaisuudessaan sijoittuu alkuerottinen `<article>` ja loppuerottimen `</article>` väliin. Dokumentin alkuerottimessa ilmoitettu attribuutti täsmentää sen käsittytapaa. Artikkeli on tarkoitus tulostaa kaksipuolisena.

DTD:n toiselta riviltä alkaen (`<!ELEMENT article ...`) nähdään artikkelin määritelty rakenne. Se koostuu pakollisesta otsikkosivusta, valinnaisesta abstraktista, valinnaisesta sisällysluettelosta, tekstin alkuosaan sijoittuvasta otsikottomasta joukosta kappaleita, joka voi myös puuttua (`p*`)

ja niiden jälkeen tulevista luvuista ja aliluvuista. Artikkelin lopussa on vielä valinnainen liite ja valinnainen kirjallisuusluettelo.

Dokumentin rakenteen kuvaus perustuu elementteihin, jotka jakaantuvat edelleen elementeiksi ja lopulta koostuvat merkkijonoista tai vakiosymboleista. Elementtejä ja niistä muodostettuja nimeämättömiä rakenneosia ryhmitellään alla esitettävien sääntöjen mukaan sulkujen avulla.

Lukumääriä ohjaavat erikoismerkit elementtinimien ja rakenneosien lopussa tarkoittavat seuraavaa. Elementtinimi ilman erikoismerkkiä tarkoittaa, että elementti on pakollinen, se voi esiintyä ainoastaan kyseisessä kohdassa ja vain kerran (`titlepag`). Elementtinimeä seuraava kysymysmerkki ilmaisee, että elementti on valinnainen (`abstract?`). Elementtinimeä seuraava ‘`+`’-merkki tarkoittaa, että elementin on esiinnyttävä kyseissä kohdassa vähintään kerran (`sect+`). Elementtinimen perässä oleva kertomerkki ilmaisee, että kyseinen elementti voi esiintyä kyseisessä kohdassa kuinka monta kertaa hyvänsä tai olla esiintymättä lainkaan (`p*`).

Indeksien ja sanastojen tekemistä varten on tekstiin voitava vapaasti sijoittaa tarpeellisia merkintöjä. Samoin ala- ja loppuviitteitä voi olla lähes missä kohdassa tahansa. Näiden varalta on käytettävissä poikkeusmerkintä, josta esimerkkinä on DTD:n alaviitteiden määrittely `+(footnote)`.

Pilkut eri rakenneosien välissä ilmaisevat, että rakenneosien on oltava täsmälleen määritellyssä järjestyksessä. Pystyviiva elementtien välissä tarkoittaa vaihtoehtoa, kyseiseen kohtaan tulee jompikumpi pystyviivalla erotetuista rakenneosista. Jos rakenneosien välissä on ‘`&`’-merkki, se tarkoittaa, että kyseiseen kohtaan tulevat molemmat rakenneosat jommassa kummassa järjestyksessä.

Esimerkkidokumentissa artikkelin alussa oleva otsikkosivu on alku- ja loppuerottimien avulla rajattu. Erottimien pakollisuus tai valinnaisuus ilmaistaan DTD:ssä elementin nimen ja sisällön määrittämisen välissä olevilla ‘`O`’ (optionaalinen, eli valinnainen) ja ‘`-`’ (pakollinen) merkeillä. Esimerkiksi nimekkeen (`title`) alkuerotin on pakollinen mutta loppuerotin ei.

Nimekkeen sisältömäärityksenä on (`#PCDATA`) ja se tarkoittaa, että nimeke koostuu nollasta tai useammasta dokumentissa käytetyn merkistön mukaisesta merkistä. Sana ei viittaa mikrotietokoneen merkistöön, akronyymi auki kirjoitettuna on *parced data characters*.

Suurin osa esimerkkidokumentin ja DTD:n riveistä on samantapaisia kuin edellä esitellyt rivit. DTD:stä on varsinaisesti esittelemättä enää vain attribuuttilista ja yksi entiteetti.

Dokumentin elementtiin `article` voi liittyä attribuutteja ja tätä mahdollisuutta on käytettykin dokumentin koodauksessa (tulostus kaksipuolisena). DTD:ssä attribuutit määritellään elementtikohtaisesti attribuuttilistan avulla. Tässä tapauksessa artikkelilla voi olla yksikäsitteinen tunniste ID, jolle ei ole määritelty mitään oletusarvoa. Dokumentin käsittelytapaa täsmentävä optio on merkkijono, jonka oletusarvona on tyhjä merkkijono.

Entiteetti `sect` on DTD:n kirjoittamisessa käytetty lyhennysmerkintä. Entiteettimäärityksen jälkeen olevilla riveillä merkintä `%sect` korvataan aina entiteetin arvolla "`heading, p* `". Entiteettien tavallisin käyttötapa on erikoismerkkien yhteydessä. Jos tekstin joukkoon halutaan esimerkiksi merkki '`<`' sellaisenaan, on käytettävä entiteettiä '`&lt`'. Ampersandi eli '`&`'-merkki ilmaistaan entiteetillä '`&amp`'.

5.4 SGML:n hyötykäyttö

Edellä olevasta, ehkä liiankin pinnallisesta esityksestä voidaan tehdä joitakin johtopäätöksiä. Hyvinkin luotettavana voinee pitää sellaista johtopäätöstä, että SGML:n ottaminen tuotantokäyttöön isossa organisaatiossa on vaativa ja monitahoinen tehtävä. Hyötyjen on oltava todellisia ja kiistattomia. Niiden on lisäksi oltava taloudellisilla indikaattoreilla mitattuna merkittäviä, jotta tarvittavat investoinnit on perusteltua tehdä. Pienten aineistomäärien käsittelyssä SGML saattaa osoittautua ylimitoitetuksi.

Esimerkkinä monien näkökohtien samanaikaisesta huomioon ottamisesta esittelen lyhyesti eduskunnan asiakirjatuotannon ja tekstiarkiston siirtymistä SGML:n käyttöön. Eduskunnan tietohallinnon tekemät ratkaisut ovat monella tavoin edelläkävijän asemassa suomalaisessa yhteiskunnassa. Siellä tehdyt ratkaisut ovat säteilleet laajalti muuhun yhteiskuntaan.

SGML ja Internetin käyttöönotto kietoutuvat saumattomasti yhteen eduskunnan ratkaisuissa. Eduskunnan tietohallinto selvitti vuonna 1994 samanaikaisesti asiakirjatuotannon ongelmia ja edustajien käyttöön tarkoitetun FAKTA-järjestelmän teknisiä perusteita [19]. Tietohallinnossa päädyttiin yhtenäiseen ja kokonaisvaltaiseen ratkaisuun, jonka ytimenä on asiakirjojen tuottaminen ja tallentaminen SGML-muodossa. Järjestely mahdollistaa automaattiset muunnokset sekä painatuksessa tarvittavaan muotoon että Internetin selainten käyttämään muotoon. Internetin selainten käyttö on puolestaan osoittautunut parhaaksi ratkaisuksi integroitaessa eri puolilla julkishallintoa tuotettuja informaatiopalveluja

riittävän helppokäyttöisiksi.

Kun edellä olevat rivit kirjoittaa julki syksyllä 1997, suurin osa lukijoista ei huomaa niissä mitään erityisen merkille pantavaa. Jotta eduskunnan tekemän ratkaisun ennakkoluulottomuuden pystyy ymmärtämään, on ensinnäkin muistettava, että valtiovalta ei vielä vuonna 1994 ollut lainkaan ottamassa Internetiä käyttöön ainakaan keskeisimpien hallintovirastojen tuotantokäytössä. Valtionhallinnon tietohallinnosta vastaavien virkamiesten, päällikköjen ja johtavien asiantuntijoiden kanta Internetin suhteen oli myönteisimmilläänkin torjuva. Suurille tietojenkäsittelyalan yrityksille Internet oli täysin tuntematon eikä mistään nykyisen kaltaisesta tietoverkkojen vyörystä pystytty edes haaveilemaan.

Ei myöskään SGML:n käyttöönotto ollut mitenkään itsestäänselvyys. Kun eduskunnan tietohallinto käynnisti SGML-hankkeen yhteistyössä Jyväskylän yliopistossa toimivan, AIRI SALMISEN johtaman työryhmän kanssa [22], valtioneuvoston kanslia selvitti omalla tahollaan WP 5.1:n käyttöä valtioneuvoston tekstiarkistojen tallennusmuotona. Lähestymistavat tuskin voisivat olla kauempana toisistaan. Ne eroavat toisistaan sekä tietojenkäsittelyn pitkäaikaista kehitystä koskevan arviointikyvyn osalta että varsinkin sen osalta, miten asiakirjajulkisuuteen ja kansalaisten demokraattisiin oikeuksiin suhtaudutaan.

Eduskunnan tietohallinnon tavoitteena SGML-hankkeen ensi metreiltä alkaen on ollut tarjota lainsäädäntötyöhön liittyvät ja sen tuottamat asiakirjat mahdollisimman tehokkaalla ja taloudellisella tavalla maksutta koko suomalaisen yhteiskunnan käyttöön. Eduskunnassa asiakirjoja ei tuoteta ja tallenneta pienen virkamiespiirin ja — kuten yleisesti käytetty eufemismi kuuluu — tärkeimpien sidosryhmien käyttöön (vrt. WP 5.1 asiakirjojen tavoitettavuuden esteenä). Eduskunta ei myöskään myy asiakirjojaan tai informaatiopalvelujaan, laki ei anna siihen edes mahdollisuutta. Eduskunta edustaa kansaa ja eduskunnan työn tärkeimmät tulokset, so. SUOMEN LAKI, on tarkoitettu kansakunnan käyttöön.

Vuonna 1994 käynnistyneet hankkeet ovat tulossa vuoden 1998 aikana varsinaiseen tuotantokäyttöön. Valiokuntien tekstintuotannossa siirrytään SGML-pohjaisten työvälineiden käyttöön ja koko eduskunnan tekstiarkisto muutetaan SGML-muotoon. Käytännössä ratkaisu toimii niin, että tavalliset mikronkäyttäjät tuottavat perustekstit tavanomaisilla tekstinkäsittelyohjelmilla, joiden tyylitiedostojen avulla saadaan aikaan SGML:n perusmerkinnät. Valiokuntien osastosihteerit tuottavat lopulliset SGML-muotoiset asiakirjat, jotka viedään heti valmistuttuaan tekstitietokantaan. Edustajat ja vir-

kamiehet käyttävät eduskunnan sisäistä tekstiarkistoa WWW:n selainten avulla. Kun asiakirjat tulevat julkisiksi, ne kopioidaan automaattisesti eduskunnan julkiseen tekstitietokantaan, jota kuka tahansa kansalainen voi tutkia oman selaimensa avulla. Aikaviive asiakirjan julkiseksi tulon ja julkiseen WWW-palveluun tulon välillä on muutamia tunteja.

Tässä on tärkeää korostaa, että muunnos SGML-muodosta HTML-muotoon on täysin automaattinen. HTML-muotoisissa dokumenteissa olevat lukuisat hyperlinkit syntyvät nekin automaattisesti. Valtaosa hyperlinkeistä on itse asiassa tarkasti kohdennettuja hakuja tekstitietokantaan. Kun käyttäjä jotain pöytäkirjaa selatessaan valitsee hyperlinkin osoittaman hallituksen esityksen, esittelymuistion, edustajan suullisen kysymyksen tai jonkin muun asiakirjan, dokumentti haetaan linkkiin “piilotetun” kyselyn avulla tekstitietokannasta SGML-muotoisena, rakennetaan hyperlinkit, muunnetaan DTD:n ja muunnostaulujen avulla HTML-muotoon ja näytetään selaimessa. HTML-muodossa asiakirja on ainoastaan selailun ajan. HTML-muotoisia asiakirjoja ei tallenneta mihinkään.

Tällä järjestelyllä on useita merkittäviä etuja. Ensinnäkin SGML:n kontrolloitu määrämuotoisuus varmistaa ajonaikaisesti rakennettavien hyperlinkkien tarkkuuden. Yhteenkään dokumenttiin ei koskaan lisätä hyperlinkkejä käsityönä. Kymmeniin tai satoihin tuhansiin dokumentteihin linkkien lisääminen käsityönä vaatisi suunnattoman määrän työtä ja suuren määrän työntekijöitä. Vielä merkittävämpi on kuitenkin etu, joka jo muutaman vuoden sisällä saavutetaan, kun nyt käytössä oleva HTML-koodaus vanhenee käyttökelvottomaksi. Vanhentuneen HTML-koodauksen muuntaminen automaattisesti johonkin uudempaan ja mitä todennäköisimmin ilmaisuvoimaisempaan merkintäkieleen on jo periaatteessa mahdotonta. Muunnos nykyisestä HTML:stä johonkin tulevaisuuden järjestelmään on ennemmin tai myöhemmin pakko tehdä käsin.

Eduskunnan tekstiarkiston käyttöön tuleva SGML-koodaus sen sijaan mahdollistaa automaattiset muunnokset mihin tahansa tulevaisuudessa käyttöön tulevaan selainten merkintäjärjestelmään, kunhan vain myös uusi järjestelmä on SGML-standardin kanssa yhteensopiva. Muunnoksessa tarvitaan vain uudet muunnossäännöt, jotka koskevat samalla kertaa kaikkia dokumentteja. Ihmisen tekemää työtä tarvitaan vain uusien muunnossääntöjen laatimiseen ja se on kertaluonteinen tehtävä. Dokumentteja ei tallennusta silmällä pitäen muunneta yksitellen tai tietokannoittain, niitä ei tarvitse lainkaan tallentaa muunnetussa muodosssa. Dokumenttien annetaan olla tekstitietokannassa alkuperäisessä SGML-muodossaan. Uudenlainen muunnos

tehdään ainoastaan ajonaikaisesti uusien muunnossääntöjen avulla kyseisen ajankohdan selainten käyttämään katselumuotoon. Toisin sanoen, jos tulevaisuudessa WWW:n ja sen selainten palvelut muuttuvat radikaalilla tavalla ja niiden hyödyntämisessä tarvittava koodaus muuttuu täysin nykyisestä koodauksesta, myös ne voidaan ottaa täysimääräisesti käyttöön.

Eduskunnan tietohallinnon tekemiä ratkaisuja on jo jonkin aikaa voinut eräänlaisena välivaiheen versiona katsella Internetin välityksellä. Tekstiarkiston asiakirjat ovat vielä toistaiseksi pelkkää raakatekstiä ja sen vuoksi WWW:n selainten näyttämät dokumentit ovat muotoilujensa osalta hyvin vaatimattomia. Dokumentit ovat pääasiassa pelkkää raakatekstiä ja dokumenttien sisältä hyperlinkit puuttuvat kokonaan. Vain dokumenttien alussa olevien määrämuotoisten tietojen perusteella pystytään ajonaikaisesti tekemään toimivia hyperlinkkejä. Tämä tilanne siis muuttuu täysin vuoden 1998 aikana. Syksyllä 1998 eduskunnan kokoontuessa kesän jälkeen uudet palvelut tulevat olemaan käytössä.

Suomen eduskunnan ratkaisujen edistyksellisyydestä saa parhaan käsityksen, kun toteaa, että Euroopassa vain kaksi muuta parlamenttia käyttää tai harkitsee SGML:n käyttöä. Norjassa parlamentin asiakirjat muunnetaan niiden jalostusketjun loppuvaiheessa SGML-muotoon painatusprosessin nopeuttamiseksi ja tehostamiseksi. SGML-muotoa ei hyödynnetä muulla tavoin. Saksassa SGML:n käyttöä tiettävästi parhaillaan tutkitaan ja kokeillaan rajoitetusti. Kaikkialla muualla, myös Ruotsissa, parlamenttien WWW-palveluihin tuottamat asiakirjat koodataan joko käsin tai tekstinkäsittelyohjelmien avulla HTML-merkinnöillä. On itsestään selvää, että kyseessä on nopeasti umpikujaan johtava ratkaisu. Ruotsissa onkin jouduttu jo toteamaan, että kymmenien tuhansien dokumenttien ajantasalla pitäminen koodauksen osalta ajaa ylläpidosta vastuussa olevan henkilöstön kestämättömään tilanteeseen. Ylläpito on toisin sanoen räjähtämässä käsiin.

5.5 SGML:n “suuret” sovellukset

SGML on eri puolilla maailmaa hyvin laajassa käytössä. Sen mukaisia tyyppimäärityksiä on tehty kaikkiin kuviteltavissa oleviin käyttötarkoituksiin ja osa näistä tyyppimäärityksistä on julkisia. On olemassa hyvin suppeita määrityksiä, kuten WWW:n hypertekstidokumenteissa käytettävä HTML (asiasta tarkemmin hiukan jäljempänä). Lisäksi on olemassa erittäin laajoja määrityksiä, joista käytän tässä nimitystä SGML:n “suuret” sovellukset.

Esittelen muutamalla sanalla USA:n puolustushallinnon tekemää määritystä CALS ja humanistisen tutkimuksen käyttöön laadittua TEI-suositusta.

CALS (*Continuous Acquisition and Life-Cycle Support*) on Yhdysvaltain puolustushallinnon tekemä SGML:n mukainen teknisten dokumenttien kuvausjärjestelmä. Kun muistetaan, että suurimpien suihkukoneiden tekninen dokumentaatio on niin laaja, että kone ei pysty nousemaan ilmaan, jos dokumentit lastataan siihen mukaan, on helposti ymmärrettävissä, että Yhdysvaltain armeijan dokumentaatio kokonaisuudessaan käsittää suunnattoman määrän aineistoa. Dokumentteja silti jatkuvasti tarvitaan ja niitä joudutaan käyttämään hyvinkin vaikeissa ja vaihtelevissa olosuhteissa. CALS:n kantavana ideana on tehdä dokumentaatiosta sellainen, että yksittäisiä dokumentteja voidaan atk:n keinoin muuntaa joustavasti muodosta ja laajuudesta toiseen niin, että kaikki kuviteltavissa olevat käyttötilanteet voidaan kattaa.

CALS:n DTD on vapaasti saatavissa ja sitä on hyödynnetty sekä Ruotsin että Suomen puolustushallinnon vastaavissa hankkeissa. Vaikka perusmääritys on saatu käyttöön veloituksetta, sen soveltamisessa paikallisiin olosuhteisiin ja tarpeisiin ja tarvittu useiden henkilötyövuosien työpanos.

TEI (*Text Encoding Initiative*) on humanistisen tutkimuksen puolella syntynyt aloite, jonka tarkoituksena on ollut luoda mahdollisimman pitkäikäinen dokumenttien kuvausjärjestelmä. Järjestelmää on tarkoitus käyttää mm. kirjallisuuden, kielitieteen, historian ja arkeologian dokumenttien kuvaamisessa ja tallentamisessa. Ajatuksena on, että kun nyt koodataan ja tallennetaan esimerkiksi papyruskääröjen tekstejä, jotka ovat tuhansia vuosia vanhoja, tallennetun aineiston tulisi muotonsa puolesta olla sellaista, että se olisi käyttökelpoista tuhansia vuosia eteenkin päin. On varmaa, että aineiston tallennusmuotoa joudutaan tulevaisuudessa muuttamaan. Oleellisinta on, että muunnokset pystytään tekemään tietokoneiden avulla automaattisesti.

TEI-suositus (*Guidelines for Text Encoding for Interchange*) on laaja, se on yli 1400 sivua. TEI on silti nopeasti vakiinnuttanut asemansa ja levinnyt varsinkin suurimpien yliopistojen tekstiarkistojen käyttöön.

5.6 HTML, XML ja World Wide Web

WWW-dokumentit koodataan HTML-merkinnöillä (HTML, *HyperText Markup Language*). HTML-koodit on määritelty SGML-standardin mu-

kaisesti, joten HTML on SGML:n sovellus. Ensimmäisen HTML:n määrityksen kirjoitti TIM BERNERS-LEE CERNissä ollessaan. Nykyisin määritystyötä tehdään MIT:ssa toimivan WWW-konsortion (W3C) puitteissa, johon kuuluvat mm. IBM, Microsoft, Netscape Communications Corporation, Novell, Spyglass ja SoftQuad. Berners-Lee toimii W3C:n johtajana.

Alkuperäinen HTML osoittautui hyvin nopeasti liian suppeaksi käytännön tarpeisiin ja siihen alettiin suunnitella laajennuksia. Näin syntyi vähitellen kirjava joukko erilaisia HTML:n murteita ja laajennuksia. Seuraava yhtenäinen ja kohtalaisen pitkään käytössä pysynyt versio oli HTML 2.0, joka on julkaistu IETF:n dokumenttina RFC 1866. Alkuperäiseen määritykseen verrattuna versio 2.0 toi lisäpiirteinä lomakkeet ja niiden käsittelyssä tarvittavat kuvakkeet (painonapit ja alasvetovalikot).

Seuraavaksi syntyi HTML+ tai HTML 3.0, joka sisälsi suunnilleen kaiken, mitä hyperteksteissä ajateltiin tarvittavan. Siinä oli tietenkin käytettävissä taulukot mutta sen lisäksi matemaattiset lausekkeet. Osa näistä lisäyksistä osoittautui sellaisiksi, että tarkkaa määrittelyä ei pystytty tekemään tai sitten selainten tekeminen kävi liian vaikeaksi. HTML:n versio 3.0 ei sen vuoksi tullut koskaan varsinaiseen käyttöön.

Määrittelytyö käynnistettiin uudelleen ja samalla pyrittiin systemaattisempaan ja realistisempaan lopputulokseen. Tätä kirjoitettaessa versio HTML 3.2 on "virallinen".[3] Se on päivätty 11.1.1997 eli se on ollut hyväksyttynä vasta vajaan vuoden verran. Version HTML-4.0 alustava määrittely on jo julkistettu (viimeisimmän päiväys 17.9.1997). Uusi versio on jälleen hyvin laaja on pelättävissä, että HTML-4.0:n kohtalo on sama kuin HTML-3.0:n, eli se ei ehkä koskaan vakiinnuta asemaansa laajasti käytettyjen selainten yhteydessä.

Kirjastoalan ihmisten mutta miksei myös kotisivuja harrastuksekseen tekevien on tärkeää ymmärtää, että HTML on todellakin SGML-sovellus. Vaikka WWW-dokumentteihin voi vielä nykyisellään kirjoittaa lähes mitä tahansa SGML:ää muistuttavia merkintöjä ilman, että selaimet niistä häiriintyvät, tilanne voi koska tahansa muuttua. On hyvin mahdollista, että tietoturva-, arkistointi- tai muista syistä WWW:n selaimet ja muut ohjelmat alkavat edellyttää, että WWW:n dokumentit ovat tarkasti jonkin "virallisesti" hyväksytyn ja julkisesti saatavissa olevan DTD:n mukaisia. Silloin WWW-dokumenttien alussa on välttämättä oltava merkintä tyyppimäärityk-

[3] Ks. `http://www.w3.org/MarkUp/`, josta löytyy tiedot määrittelyn kunkin hetkisestä tilanteesta ja linkkejä suureen joukkoon HTML-aihetta käsitteleviä dokumentteja.

sestä ja dokumenttien on oltava rakenteeltaan seuraavalla sivulla kuvatun kaltaisia.

Lisäksi kunkin dokumentin tulee todella olla sen alkutiedoissa mainitun DTD:n mukainen. Määritys on täysin formaali ja sen noudattaminen on ohjelmallisesti tarkastettavissa. WWW:stä löytyy julkisia validointipalveluja. Janne Himangan tekemä versio on osoitteessa `http://oyt.oulu.fi/validointi.html`.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
  <HTML>
  <HEAD>
  <TITLE>A study of population dynamics</TITLE>
  ... other head elements
  </HEAD>
  <BODY>
  ... document body
  </BODY>
  </HTML>
```

Kuten edellä esitetystä on helposti pääteltävissä, kehitys etenee henkeä ahdistavalla ja ajattelukykyä sumentavalla nopeudella. WWW:stä ja HTML:stä kirjoittaminen onkin hyvin ongelmallista siksi, että esitellyt asiat, viittaukset verkossa oleviin dokumentteihin ja arvelut tulevasta kehityksen suunnasta vanhenevat jo sinä aikana, kun kirjoitusta ladotaan ja painetaan. Tämä on kuitenkin myös standardien kehittäjien ongelma. Mikä tahansa huolella tehty standardi vanhenee käsiin ennen kuin se on saatu edes julkaistavaan kuntoon.

Tämän vuoksi HTML:n jatkokehityksessä ollaan suuntautumassa dynaamisiin määrittelyihin, jotka tekevat mahdolliseksi merkintäkielten laajentamisen ilman, että tyyppimääritystä tarvitsee muuttaa. Tästä päästäänkin luvun viimeiseen aiheeseen, joka on XML.

5.6.1 Laajennettava merkintäkieli

HTML:n standardoinnissa ja HTML-koodauksessa on useita vakavia ongelmia. Internetin kaupallisten selainten kova kilpailu maailmanmarkkinoilla näkyy standardin paisumisena ja rönsyilynä. HTML-4.0:n luonnos on yli 300 sivuinen ja sisältää ristiriitaisia piirteitä, jotka estävät täysitoimisten selainten tekemisen. Luonnokseen on jokainen halukas saanut oman puumerkkinsä ja on odotettavissa, että selaimet eivät pysty näyttämään kilpailijoiden tuottamia dokumentteja. Toisena ongelmana on, että luonnos ehtii tosiasiallisesti vanheta ennen kuin se ehditään hyväksyä WWW-konsortion suositukseksi.

Ratkaisuksi tähän ongelmaan ollaan kehittämässä laajennettavissa olevaa merkintäjärjestelmää XML[4]. Tavoitteet ovat erittäin kunnianhimoiset. Samalla kertaa pyritään ratkaisemaan kaikki keskeiset HTML:n ongelmat. (1) Merkintäkielestä halutaan tehdä dynaamisesti laajennettava, (2) hyperlinkeistä halutaan nykyiseen verrattuna oleellisesti monipuolisempia ja (3) selaamisen ja ladonnan ohjaamiseen halutaan laajennettavissa olevat tyylitiedostot. Lisäksi näyttää siltä, että XML:stä on nopeasti tulossa tietoverkoissa olevien resurssien hallinnan yleinen merkintäjärjestelmä.

XML ei ole merkintäkieli samaan tapaan kuin HTML. XML (*Extensible Markup Language*) on pikemminkin verrattavissa SGML:ään, se on rajoitettu SGML — se noudattaa SGML-standardia. Rajoittaminen on tehty niin, että sellaiset SGML:n piirteet on pudotettu pois, jotka aiheuttavat tietoverkoissa käytettävien ohjelmien tekemisessä tehokkuus- tai johdonmukaisuusongelmia. SGML:n, HTML:n ja muiden standardien laatimisesta ja niiden hyödyntämiseksi tarvittavien ohjelmien — erityisesti selainten ja ladontaohjelmien — tekemisestä on kertynyt huomattava määrä kokemuksia. XML:n kehittämisessä on mukana suuri joukko alan kaikkein kokeneimpia ihmisiä. Edustettuna ovat kaikki keskeiset SGML-tuotteita ja WWW:ssä käytettäviä ohjelmia tuottavat yritykset.

(1) Merkintäkielen laajennettavuus antaa mahdollisuuden ottaa käyttöön uusia palveluja voimassa olevaa standardia muuttamatta. WWW:stä löytyvässä dokumentissa voidaan määritellä uusia rakenneosia, niiden keskinäisiä suhteita, rakenneosiin liittyvää toiminnallisuutta ja käyttäjän ominaisuuksien huomioon ottamista. Näiden uusien palvelujen käyttömahdollisuus tulee luonnollisesti riippumaan käytetyn selaimen ja sitä tukevien ohjelmien monipuolisuudesta. Standardin mukaiset laajennukset turvaavat kuitenkin vähimmillään sen, että selaimet tekevät dokumentin kanssa parhaansa; näyttävät mitä osaavat ja jättävät muun osan rakenteesta ja toiminnallisuudesta vaille huomiota.

Esimerkiksi dokumentin sisäisen rakenteen laajennuksesta käy hyvin potilastietojen siirtäminen sairaalan WWW-palvelimesta yksityisen terveyskeskuksen WWW-palvelimeen. Käytetyt tietokannat voivat olla keskenään täysin yhteensopimattomia. XML:n avulla voidaan kuitenkin kuvata potilastietojen hierarkkinen rakenne SGML:n rakenneosia hyödyntämällä. Potilastietoja sisältävä dokumentti sisältää tällöin varsinaisten henkilötietojen lisäksi tietojen rakenteen ja keskinäisten suhteiden formaalin kuvauksen.

[4]Hankkeen kotisivu on `http://www.w3.org/XML`.

Riittävät valtuudet omaava lääkäri voi selaimessa hiirellä siirtää potilaskansion tietokannasta toiseen ja käyttää luottamuksellisia ja tarkkoja tietoja tilannekohtaisen diagnoosin tekemiseen. On helppo kuvitella, että tällaisella mahdollisuudella voi hätätilanteessa olla suuri merkitys.

Laajennettavuus liittyy myös dokumenttien sisältämään toiminnallisuuteen. WWW:n selaimet ovat muuttumassa sellaisiksi, että ne pystyvät yhä enemmän suorittamaan dokumentteihin upotettuja toiminnallisia piirteitä. Yksikertaisimmillaan kyse voi olla vilkkuvista tai vilistävistä tekstinosista. Kehittyneemmissä ratkaisuissa selain voi suorittaa hyvinkin vaativaa tietojen muokkausta. Esimerkiksi karttapiirros voidaan muotoilla, viimeistellä ja yhdistää muuhun aineistoon vasta selaimessa. Tulevaisuudessa kuvien jakelu ei enää perustu ennalta muodostettuihin ja tallennettuihin lopullisessa muodossa oleviin rasterikuviin.

Käyttäjän ominaisuuksien huomioon ottaminen liittyy käyttäjän mielenkiintoprofiilin hyödyntämiseen. Käyttäjä voi määritellä etukäteen, että hän haluaa aivan erityisesti selata johonkin tiettyyn aihepiiriin kuuluvaa aineistoa eikä halua lainkaan nähdä johonkin toiseen aihepiiriin kuuluvaa aineistoa. Jos aineisto on kuvailtu asianmukaisella tavalla, selain voi korostaa linkkejä haluttuun aineistoon ja jättää kokonaan näyttämättä linkit eihaluttuun aineistoon.

(2) HTML:n käyttämät hyperlinkit edustavat hyvin pientä osaa kaikesta siitä, mitä hypertekstien käsittelyyn oli kehitetty jo 1970-luvulla. Hyperteksti, kuten on hyvä tietää, on varsin vanha keksintö. XML:n avulla pystytään hyperlinkeistä tekemään monipuolisempia. Linkit voivat olla kaksisuuntaisia ja dynaamisia, niillä voidaan hallita osista koostuvaa koottua dokumenttia (esim. erillisissä tiedostoissa olevat kirjan luvut) ja ne voivat kohdistua rasterimuotoisten kuvien osiin tai osoittaa jotain videon kohtaa. Videokuva, jossa on monikanavainen ääni ja mahdollisesti useita valinnaisia tekstityksiä sekä muuta linkitettyä oheismateriaalia on hyperlinkkien muodostamiselle ja hallinnalle eräänlainen haaste.

(3) Tyylimäärittelyt ovat osa SGML:n jatkokehitystä. SGML:n esittelyn alkupuolella jo totesin, että asiakirjojen pitkäaikaista käytettävyyttä on SGML:n avulla turvattu osittain ulkoasun kustannuksella. Kirjojen ja lehtien kustantajat eivät ole tyytyneet pelkkään SGML:ään. Rinnalle on kehitetty DSSSL-standardi (ISO/IEC 10179)[5]. Sen avulla SGML:n mukaiseen dokumenttiin voidaan liittää lisämerkintöjä, joiden avulla dokumentin ra-

[5]ISO/IEC 10179:1996: *Document Style Semantics and Specification Language*.

kenneosat kytketään tulostusta ja katselua ohjaaviin tyylimäärittelyihin. Yksinkertaisimmillaan kyse on otsikkorivien lihavoinnista ja sivunvaihtokohdista. Monimutkaisemmaksi asia muuttuu, kun tyylitiedostojen avulla ohjataan automaattista sisällysluettelojen, sanastojen ja hakemistojen tekemistä. Tyylitiedostoilla voidaan ohjata myös dokumentin eri osien näkymistä selaimessa. Yhdessä ikkunassa voi olla sisällysluettelo, toisessa dokumentin rakennetta kuvaava hierarkkinen kaavio, kolmannessa dokumenttiin piilotetut lukijoiden kommentit, neljännessä valikoituja osia jostain tärkeimmistä kysymyksistä, jne.

Tyylitiedostoille on luonteenomaista, että makukysymykset tulevat vahvasti esille. Sen vuoksi tyylimäärittelyjä on pystyttävä soveltamaan kasautuvasti (engl. *cascading*). Dokumentin tekijä laatii ensin oman tyylimäärityksensä, informaatiopalvelujen jakelija liittää siihen omansa ja lopulta käyttäjä muuttaa jotkut yksityiskohdat omien mieltymystensä mukaisiksi. Kasautuvia tyylimäärityksiä on pystyttävä hallitsemaan niin, että ristiriitoja ei synny eikä alkuperäinen dokumentti jää joiltakin osin vahingossa kokonaan näkemättä.

Luku 6

Nimeäminen ja paikantaminen

Tietoverkkojen käytössä informaatiolähteiden paikantaminen ja dokumenttien löytäminen tulevat olemaan kirjastoalan ja tietojenkäsittelyalan ammattilaisten vaikein ja haastavin yhteinen ongelmakentta ainakin lähimpien tulevien vuosien ajan. Tähän ongelmakenttään viitataan usein käsitteellä *metatieto*, eli tieto tiedosta. Kirjastoammattilaisen ensi reaktio tämän käsitteen käyttöön saattaisi olla, että kysymyksessä on vain viitetietojen kutsuminen jollakin uudella nimellä. Viitetiedot ovat todella yksi osa metatietoja, mutta niitä on paljon muitakin.

Meneillään on kokonaisuutena tarkastellen hyvin mielenkiintoinen ja monitahoinen integroitumiskehitys. Automaattinen tietojenkäsittely on nyt toden teolla tulossa kirjastojen käyttöön. Vuosikymmenien kehitys on lopultakin alkanut tuottaa sellaisia laitteita ja ohjelmia, että suurtenkin kirjastojen kaikkien kirjojen tekstit kuvineen voidaan kohtuullisin kustannuksin tallentaa tietokoneiden muistilaitteille. Tällaisen dokumenttimäärän tallentamismahdollisuus muuttaa samalla merkittävästi tietoverkkojen tiedonhallinnan luonnetta. Suurten tietovarantojen hyödyntäminen edellyttää sellaista tiedonhallintaa, joka rakentuu kirjastoalan ammattitaidon varaan, mutta jossa perinteistä ammattitaitoa on tarpeen soveltuvilta osin täydentää tietoverkkojen hallintaa koskevilla tiedoilla ja taidoilla.

Seuraavassa käsitellään tästä monitahoisesta ongelmakentästä joitakin ajankohtaisia ja paljon huomiota herättäneitä kysymyksiä. Ensimmäinen koskee informaatiopalvelun paikantamista. Toisena ongelmana on dokumenttien tunnisteiden laatiminen niin, että dokumentin tyyppi, sijainti ja yksikäsitteinen nimi pystytään sisällyttämään tunnistetietoihin. Kolmantena käsitellään dokumenttien kuvailua verkkokäyttöön soveltuvalla tavalla.

Lopuksi esitellään aivan alkuvaiheessa olevaa hanketta, jonka tarkoituksena on luoda yleiset puitteet tietoverkoissa olevien resurssien kuvailulle ja hallinnalle.

Kaikissa näissä kysymyksissä edellä käsitellyt standardit ja yhteyskäytännöt ovat tavalla tai toisella mukana. Kyseessä on alue, joka kaikilta osin kehittyy tavattoman nopeasti ja vireillä on lukuisia keskenään kilpailevia hankkeita. Tähän on kuitenkin yritetty valita sellaisia hankkeita, joilla näyttäisi olevan riittävän vahva tausta vakiintumista ajatellen. Vähintäänkin ne ovat tutustumisen ja seuraamisen arvoisia.

6.1 Tietojen paikantaminen

Yhdysvalloissa viranomaisten keräämiä tietovarantoja pidetään julkishyödykkeinä (*public good*) ja kansalaisten oikeus tietoon on lainsäädännössä hyvin vahvasti suojattu. Liittovaltion hallinto käyttää vuosittain miljardeja dollareja tietojen keräämiseen ja käsittelyyn. Koska veronmaksajat ovat jo kertaalleen maksaneet nämä tiedot, heille on sen vuoksi syntynyt oikeus niiden käyttämiseen.

Keskeisenä esteenä tietojen laajemmalle julkiselle hyväksikäytölle nähdään erilaisten hakemistojen ja muiden tiedon paikannuksessa tarvittavien apuvälineiden puuttuminen. Käyttäjät eivät yksinkertaisesti tiedä, mitä tietoja on tarjolla ja miten olemassa olevia tietoja voisi katsella ja kopioida omaan käyttöönsä. Ratkaisuksi tähän ongelmaan on otettu käyttöön GILS *Government Information Locator Service* [3]. Se on hajautettu tiedon paikannusjärjestelmä, jonka avulla käyttäjä voi tunnistaa, paikallistaa, selata ja hankkia käyttöönsä julkisesti saatavilla olevia liittovaltion tietovarantoja. GILS käyttää sovellustason protokollana ANSI/NISO:n Z39.50-standardin määrittelemää protokollaa.

GILS-järjestelmän ovat suunnitelleet Yhdysvaltain Geologian tutkimuskeskus (USGS *United States Geological Survey*) ja *Syracusan yliopisto*. Viimeisin suunnitteluvaihe (jatkossa GILS-projekti) käynnistyi syyskuussa 1993 ja loppuraportti on päivätty 7.9.1994 [17]. Tätä vaihetta on edeltänyt mainittujen tahojen monivuotinen ja monivaiheinen selvitystyö, joka osittain on tähdännyt GILS-järjestelmän suunnittelemiseen, osittain se on liittynyt laajempaan NREN-hankkeeseen.

GILS-projektin suoranaisina tehtävinä oli

- Z39.50-standardia koskevan tutkimus- ja kehitystyön edistäminen erityisesti liittovaltion julkisten tietovarantojen käytössä
- yhteisen näkemyksen löytäminen Z39.50-standardin käytössä eri osapuolten kesken
- Z39.50:n ja muiden standardien käyttöä koskevan sovellusprofiilin määritteleminen GILS-järjestelmää varten, sekä
- testitoteutusten tukeminen ja niiden tekemiseen kannustaminen profiilin soveltuvuuden arvioimiseksi ja eri toteutusten yhteentoimivuuden varmistamiseksi.

Valkoisen talon budjetointitoimisto (OMB *Office of Management and Budget*) on antanut liittovaltion virastoja velvoittavat ohjeet GILS:n käyttöönottamisesta erityisellä tiedotteellaan. Tiedote perustuu OMB:n kiertokirjeeseen OMB Circular No. A-130, “Management of Federal Information Resources”, joka on tärkein liittovaltion tietohallintoa käsittelevä asiakirja.

OMB:n tiedotteen kohdassa 5 määritellään GILS:n ohjelmistopolitiikka. GILS-palvelut tullaan järjestämään siten, että asiakkaat voivat käyttää ei-kaupallisia ohjelmia (*nonproprietary*) tietojen hakemiseen. Lisäksi todetaan, että Government Printing Office (GPO), National Technical Information Service (NTIS) ja Clearinghouse for Networked Information Discovery and Retrieval (CNIDR) tulevat levittämään tähän tarkoituksen soveltuvia julkisohjelmia (*public domain*). Mainittakoon, että CNIDR on NSF:n (*National Science Foundation*) rahoituksella toimiva hanke, joka kehittää WAIS:n pohjalta Z39.50-1992:n mukaista tekstitietokantaohjelmistoa (julkisohjelma).

GILS-palvelut otettiin käyttöön joulukuussa 1994.

6.1.1 GILS:n palvelut

Kullakin liittovaltion virastolla on oma GILS-palvelin. Ne eivät muodosta keskenään mitään hierarkiaa. GILS-palvelua voi käyttää suoraan tai välillisesti. Suorakäyttäjä kytkeytyy GILS-palvelimeen Internetin kautta ja jos asiakasohjelma (*client*) ymmärtää GILS-profiilia, kaikki haku- ja selauspalvelut ovat käytettävissä. Z39.50-protokollan käyttö GILS-palvelussa turvaa sen, että mikä tahansa mainittua protokollaa ymmärtävä kirjastojen asiakasohjelma pystyy antamaan käyttäjälleen ainakin vähimmäispalvelut. Myös

VTLS:n Z39.50-client ja erityisesti Willow ovat käytettävissä GILS:n yhteydessä.

GILS-palvelua voi käyttää myös välillisesti. Välillisen palvelun tarjoaja voi olla jokin muu julkishallinnon elin tai vaikkapa kaupallinen yrittäjä, joka antaa jonkin lisäarvon virastojen muodostamille paikannustiedoille.

GILS-palvelimen tärkein tehtävä on käsitellä asiakkaan lähettämä kysely (etsintälauseke) ja muodostaa siihen vastaus (hakujoukko). Hakujoukko muodostuu paikannustietueista, jotka voivat viitata palvelinta ylläpitävän viraston omiin tietovarantoihin tai jonkin toisen viraston tietovarantoihin. GILS-palvelin sinänsä on yksi tietovarannoista. Sehän sisältää tietoja muista tietolähteistä.

Viraston omaa tietovarantoa kuvaava paikannustietue sisältää riittävät tiedot tietolähteen paikantamiseksi ja tunnistamiseksi sekä selvityksen tietojen saatavuudesta. Viimeksi mainittu on sekä ihmisen että koneen luettavassa muodossa. Koneluettavaa muotoa käytetään siirryttäessä hakemaan ja selaamaan itse tietolähteen sisältämiä tietoja.

GILS-palvelin voi hakujoukon muodostamisen lisäksi tukea lyhyessä muodossa esitettävien paikannustietueiden selaamista. Mikäli käyttäjä ei keksi mitään tapaa etsintälausekkeen muodostamiseen, hän voi turvautua selaamiseen. Tätä varten on käytettävissä ennaltamääritelty haku (*well-known search*), jossa etsintätermejä ei tarvita. Selauspalvelu ei kuulu GILS-palvelimen pakollisiin vähimmäispalveluihin.

Varsinainen tietolähteiden hakeminen tai dokumenttien selaaminen ei kuulu GILS-profiilin piiriin. Siihen soveltuvat jo nykyisinkin Internetin käytössä tarvittavat pelisäännöt.

6.1.2 GILS:n seuranta

GILS:n käyttöönotosta on kulunut kolme vuotta ja tänä aikana maailma on tietoliikenteellisessä mielessä muuttunut melkoisesti. GILS:n käyttöönotosta, käytöstä ja kertyneistä kokemuksista on kesällä 1997 ilmestynyt yli 400-sivuinen seurantaraportti [18]. Tekijöinä ovat samaiset William E. Moen ja Charles R. McClure, jotka laativat myös alkuperäisen suunnitteludokumentin [17].

Seurantaraportissa todetaan, että GILS on jäänyt osittain WWW:n ennakoimattoman vyöryn jalkoihin. WWW:n laajamittainen käyttöönotto liittovaltion virastoissa on jossain määrin korvannut GILS:n tarvetta. Kaikkia alkuperäisiä tavoitteita ei kuitenkaan tällä tavoin pystytä saavuttamaan.

WWW:n keskeisimpina ongelmina ovat juuri tiedonlähteiden paikantaminen ja aineiston systemaattinen kuvailu. Kunnollista paikantamista ei pystytä toteuttamaan strukturoimattomalla täystekstidokumenttien haulla (esim. AltaVista). Jos sen sijaan aineiston kuvailu hoidettaisiin Dublin Coren (ks. kohta 6.3.1 sivulla 76) määrittelemällä tavalla ja WWW:n sovellustason yhteyskäytännössä hyödynnettäisiin soveltuvin osin Z39.50:n ominaisuuksia, päästäisiin hyvin lähelle alkuperäisiä tavoitteita.

Periaatteellisessa mielessä tärkeämpi muutos kuluneiden vuosien ajalta koskee liittovaltion lainsäädäntöä. GILS:n suunnittelun taustalla olevat ajatuksen tietojen helpommasta tavoitettavuudesta ja vapaammasta saatavuudesta ovat tulleet kirjatuiksi lukemattomiin lakeihin, asetuksiin ja alemman tasoisiin säännöksiin ja ohjekirjeisiin. Vaikka nämä lausumat eivät realisoidukaan käytännössä kovin nopeasti, ne vastaansanomattomalla tavalla osoittavat tulevan kehityksen suuntaa. Näyttääkin siltä, että Yhdysvallat on informaation ja asiakirjojen julkisuudessa jättämässä Pohjoismaat pahasti jälkeen.

Toinen tärkeä periaatteellinen muutos koskee GILS:n periaatteiden maailmanlaajuista tunnetuksi tulemista. Akronyymi GILS onkin monissa yhteyksissä purettu auki käsitteeksi *Global Information Locator Service*, globaalinen informaation paikannuspalvelu. Tällaisen palvelun käyttötarpeen tunnustanee jokainen, joka on kesken työpäivän kiireiden yrittänyt vaikkapa EU:n lukuisista palvelimista jäljittää jotain tiettyä dokumenttia.

On siis mahdollista, että GILS tulee jäämään välivaiheeksi informaation helpomman tavoitettavuuden mutkikkaalla tiellä. Kunnianhimoisena ja periaatteellisessa mielessä tärkeänä hankkeena, se kuitenkin ansaitsee kunniapaikan länsimaisen tietoliikenteen kehityshistoriassa.

6.2 Dokumenttien nimeäminen

Viitattaessa Internetin sisältämiin dokumentteihin[1] tällä hetkellä käytetään enimmäkseen mekanismia nimeltä URL (Uniform Resource Locator, “yhtenäinen resurssipaikannin”). URL:n muoto määritellään Internet-standardissa RFC 1737. Kun URL on tiedossa, dokumentti voidaan periaatteessa noutaa käyttöön mistä päin maailmaa hyvänsä. Viittausmekanismina URL

[1]Tämän osuuden on kirjoittanut JANNE HIMANKA Oulun yliopiston Informaatiotutkimuksen laitoksesta.

on kuitenkin pahasti puutteellinen. Koska se ilmaisee tarkasti dokumentin sijainnin, pienikin muutos dokumentin sijainnissa tekee URL:n käyttökelvottomaksi. Tällaiset pienet muutokset ovat väistämättömiä: tietokantojen kasvaessa kokoelmia on organisoitava eri tavalla, organisaatiomuutosten seurauksena tietokoneiden nimet muuttuvat jne. Tilanne on verrattavissa siihen, että tieteellisten artikkelien lopussa viitattaisiin teosten signumeihin (paikkanumeroihin) tietyn kirjaston hyllyillä.

URL:n puutteet viittausmekanismina on tiedostettu jo pitkään, ja IETF:n (*Internet Engineering Task Force*) työryhmät ovat laatineet muita mekanismeja URL:ia täydentämään. Näitä ovat ennen kaikkea URN (*Uniform Resource Name*) ja URC (*Uniform Resource Characteristics)*. Yksinkertaistaen URN:ää voi verrata ISBN-numeroihin ja URC:tä MARC-tietueisiin. Näistä URN on pitemmälle kehitetty, sen syntaksi on pääpiirteissään määritelty.

URN:n toimintaperiaate on lyhyesti seuraava. Kun asiakasohjelma (esimerkiksi WWW-selain) saa käsiinsä URN:n, se kysyy ensin jostain lähellään olevasta palvelimesta (esim. nimipalvelimesta, jollainen on aina saatavilla), mistä se voisi löytää resoluutiopalvelimen tämäntyyppiselle URN:lle. Saatuaan vastauksen asiakasohjelma ottaa yhteyttä sopivaan resoluutiopalvelimeen, joka ottaa URN:n ja palauttaa joko URL:n, URC:n tai itse haetun dokumentin.

URN:ssä on kaksi osaa: nimityyppi ja tyyppikohtainen tarkennin. URN:llä voi ilmaista myös olemassaolevia nimiavaruuksia, kunhan resoluutiopalvelu on olemassa. Esimerkiksi Tähtitieteellinen Yhdistys URSAn Tähdet 1997 -teoksen elektroninen versio voisi löytyä seuraavalla URN:lla: `urn:isbn:951-9269-82-7`.

Verrattuna URL:n käyttöön tässä on siis kaksi uutta vaihetta. Ne yhdessä takaavat URN:lle pitkän elinkaaren. Ensinnäkin dokumentin sijainti voi muuttua kuinka usein hyvänsä, kunhan resoluutiopalvelin pysyy ajan tasalla. Toiseksi, vaikka resoluutiopalvelin poistuisi käytöstä, tämä mekanismi sallii myös muiden resoluutiopalvelinten käytön. Jos resoluutiopalvelinten löytämiseen käytetään nimenomaan Internetin nimipalvelua (DNS, *Domain Name Service*), on kyseessä ns. NAPTR URN -protokolla. Se on ensimmäinen URN-toteutus, josta on jo olemassa prototyyppejä.

URN:ien laajamittainen käyttöönotto on raskas prosessi. Se vaatii resoluutiopalvelinten perustamisen sekä muutoksia asiakasohjelmiin ja nimipalveluun. Etenkin nimipalvelun muuttaminen on työlästä, koska se on erittäin kriittinen osa Internetin protokollaperhettä, eikä siinä voida sallia toimin-

tahäiriöitä. Tästä syystä URN:t ovat edelleenkin prototyyppiasteella, vaikka niitä on kehitelty jo vuosia. URN-työryhmät haluavat tehdä huolellista jälkeä. Tällä hetkellä esimerkiksi URN-työryhmän postituslistalla käydään keskustelua siitä, pitäisikö URN:ssä käyttää ISO-10646 -merkistöä, jotta URN:t voisivat sisältää vaikkapa kiinalaisia kirjoitusmerkkejä.

URL:ien vanheneminen on akuutti ja laajamittainen ongelma. On vaikea käyttää URL-viittauksia esimerkiksi tieteellisissä artikkeleissa, väitöskirjoissa ja opinnäytteissä, koska URL:t voivat olla vanhentuneita kun kirjoitus tulee painosta.

6.3 Dokumenttien kuvailu

Kirjastoihin hankittava aineisto käy yleensä läpi useita käsittelyvaiheita ennen kuin se on valmis käytettäväksi. Aineisto vähintäänkin luokitellaan ja luetteloidaan ja tärkeimmälle osalle tehdään varsinainen sisällönkuvailu. MARJATTA HAIMI sanoo Eduskunnan kirjaston ohjeessa sisällönkuvailusta seuraavasti:

> Eduskunnan kirjaston kokoelmiin otetun aineiston sisällönkuvailun tarkoitus on tehdä mahdolliseksi kiinteän sanaston ja luokitusjärjestelmän käyttöön pohjautuva
>
> - aiheenmukainen tiedonhaku julkaisuista
> - aiheenmukaisten bibliografioiden, erikoisluetteloiden ja uutuusluetteloiden tuottaminen
>
> ottaen huomioon
>
> - kirjaston vastuualueeksi määritellyt tieteen- ja aihealat
> - kirjaston eri käyttäjäryhmien tiedontarpeet
> - tietojärjestelmän ominaisuudet sisällönkuvailun ja tiedonhaun kannalta.

Sisällönkuvailun suurin käytännön ongelma on siihen tarvittavan työn määrä. Sellainen aineisto, joka sijoitetaan kirjaston hyllyihin tai kirjaston omien tietokoneiden tallennuslaitteisiin, on vielä kohtuullisella työllä kuvailtavissa. Virtuaalikirjaston aikakaudella aineistot eivät kuitenkaan enää rajoitu tähän. Kirjaston tulisi asiakkaitaan varten pystyä tavalla tai toisella

kuvailemaan tai vähintäänkin järjestämään ja esittelemään myös tietoverkossa olevaa aineistoa. Mahdollisuudet tällaiseen työhön ovat käytännössä hyvin rajalliset ja sen vuoksi on välttämätöntä miettiä muita vaihtoehtoja. Sisällönkuvailu on joko pystyttävä automatisoimaan tai sitten kuvailussa tarvittava työ on järjestettävä kokonaan toisella tavalla.

6.3.1 Dublin Core

Pitkällä aikavälillä ainoa realistinen vaihtoehto aineiston kuvailuun tietoverkkojen jatkuvasti laajetessa on se, että aineiston tuottajat kuvailevat dokumenttinsa itse ja sisällyttävät tai liittävät kuvailutiedot dokumentteihin. Tähän tarkoitukseen ollaan kirjastoihmisten, atk-ammattilaisten ja dokumenttien hallinnan asiantuntijoiden yhteistyönä laatimassa kuvailutapaa, josta käytetään nimitystä *Dublin Core*[2]. Se on kuvailussa käytettävien tietojen vähimmäisjoukko, jonka rakenne ja esittämistapa on hyvin huolellisesti suunniteltu. Ideana on siis se, että mahdollisimman laajana ja mahdollisimman laajapohjaisena kansainvälisenä yhteistyönä sovitaan aineistonkuvailun menetelmästä ja sisällöstä. Kun aineistot kaikkialla kuvaillaan saman järjestelmän mukaisesti, yksittäiset dokumentit voidaan löytää erilaisista tietokannoista samojen hakumenetelmien avulla.

Kuvailujärjestelmän on välttämättä oltava sekä suppea että selkeä. Laaja ja vaikeasti tulkittava järjestelmä ei koskaan tulisi niin laajaan käyttöön, että sillä olisi todellista merkitystä koko Internet-yhteisön kannalta. *Dublin Core* sisältää 15 elementtiä, joiden englannin kieliset nimet ovat: title, author or creator, subject and keywords, description, publisher, other contributors, date, resource type, format, resource identifier, source, language, relation, coverage ja rights management. Elementit on määritelty niin, että ne mahdollisuuksien mukaan perustuvat johonkin laajasti hyväksyttyyn standardiin. Siten päiväys voidaan esittää muodossa YYYYMMDD (ANSI X3.30-1986), resurssin tunnisteena on esimerkiksi URL tai ISBN, kieli on Z39.53:n mukainen kolmikirjaiminen koodi, jne.

Työtä *Dublin Core*:n kehittämiseksi ja tunnetuksi tekemiseksi on tehty vuodesta 1995 lähtien. Laajoja seminaareja on tähän mennessä (marraskuu 1997) järjestetty yhteensä viisi kertaa. Pohjoismaissa on käynnissä laaja yhteisprojekti, joka on herättänyt laajalti kiinnostusta seminaarien osallistu-

[2]Hankkeen kotisivu on `http://purl.org/metadata/dublin_core`.

jissa.[3] Osoituksena pohjoismaisesta ja varsinkin suomalaisesta asiantuntemuksesta ja aktiivisuudesta viides seminaareista järjestettiin Helsingissä lokakuussa 1997.

Ensimmäinen Dublin Core -seminaareista (DC-1) järjestettiin OCLC:ssä maaliskuussa 1995. Silloin määriteltiin alustavasti DC:n ensimmäiset 13 elementtiä ja päästiin periaatteelliseen yhteisymmärrykseen erilaisia taustayhteisöjä edustavien osapuolten kesken työn jatkamisesta. Työssä ovat alusta alkaen olleet edustettuina kirjastot, museot ja paikkatietojen käyttäjät. Toinen seminaari (DC-2) oli Warwickissa huhtikuussa 1996. Silloin luotiin tapa paketoida (Warwick Framework) erilaisia resurssien kuvailun osa-alueita. Jäljempänä esiteltävä RDF (ks. kohta 6.4 sivulla 78) sai tällöin alkunsa ja DC asettui osaksi laajempaa resurssien kuvailun kokonaisuutta.

DC-3 pidettiin jälleen OCLC:ssä syyskuussa 1996 ja silloin aiheena olivat kuvat. Samalla elementtien määrä nostettiin nykyiseen viiteentoista. Kuvien osalta tärkeä päätös koski niiden kuvailemista DC:n elementtien avulla sen sijaan, että kuvia varten olisi alettu kehittää kokonaan toisenlaista kuvailutapaa. DC-4 pidettiin Canberrassa maaliskuussa 1997. Tällöin elementtien rakennetta kehitettiin hienojakoisemmaksi. Elementtejä voidaan täsmentää erilaisten määritteiden avulla (Canberra qualifiers) ja niihin voidaan lisätä alikenttiä pistenotaation avulla.

Havainnollisemman kuvan saamiseksi esitetään lyhyt esimerkki. Kyseessä on STUART WEIBELIN kotisivun DC-kuvailu XML:n merkintätapojen mukaisesti. Esimerkki on peräisin Weibelin Helsingin DC-5:ssä esittämiltä kalvoilta.

<namespace href="`http://purl.org/metadata/dublin_core`" as="DC">
<namespace href="`http://iso.org/spec/8601` " as="iso8601">
<ablock href="`http://purl.net/weibel`">

<DC:title> *Stuart Weibels's Home Page* </DC:title>
<DC:creator> *Stuart Weibel* </DC:creator>
<DC:date> <iso8601:date> *1997-04-02* </iso8601:date>
</DC:date>
<DC:subject>
metadata, Dublin Core, URN, electronic publication, book reviews

---

[3] Pohjoismaisen projektin kotisivu on `http://linnea.helsinki.fi/meta/`. Suomen edustajana tässä projektissa on JUHA HAKALA TKAY:stä.

```
    </DC:subject>
</ablock>
```

Esimerkissä määritellään aluksi kaksi nimiavaruutta, DC ja iso8601, ja kerrotaan niiden verkko-osoitteet. Myöhemmin niitä käytetään elementtitunnisteiden yhteydessä. Esimerkiksi otsikko (title) kuuluu nimiavaruuteen DC ja sen täsmällinen käsittelytapa on tarvittaessa löydettävissä alussa mainitussa verkko-osoitteessa olevasta kuvauksesta. Päiväyksessä yhteydessä on määrite (qualifier), joka kertoo päiväyksen esittämisessä käytetyn koodauksen. Kyseessä on esimerkki skeemasta ja ideana on, että jonakin päivänä skeemat ovat verkon palvelimissa tietokoneella tulkittavassa muodossa. Muita skeemoja voisivat olla esim. LCSH (Library of Congress Subject Heading), MESH (MEdical Subject Headings) ja DDC (Dewey Decimal System).

Viimeisin seminaari DC-5 oli siis lokakuussa 1997 Helsingissä. Seminaarissa käytiin läpi kokemuksia niistä eri puolilla maailmaa käynnissä olevista hankkeista, joissa DC:tä ollaan ottamassa todelliseen käyttöön. Seminaarissa keskusteltiin myös DC:n tärkeimpien osien “virallisesta” standardoimisesta. Tällä luonnollisesti tarkoitetaan dokumenttien julkaisemista IETF:n RFC-sarjassa. Lisäksi keskusteltiin valmiudesta käyttäjille tarkoitettujen oppaiden tekemiseen.

Dublin Core on GILS-profiilin osajoukko ja se on vaikeuksitta esitettävissä SGML:n mukaisesti koodattuna. Sen elementit voidaan helposti sisällyttää HTML-koodattuihin WWW-dokumentteihin. Toistaiseksi DC on tarkasti spesifioitu vain HTML-2.0 ja HTML-4.0 määritysten mukaisesti. Syy tähän on hyvin yksinkertainen. Ainoastaan WWW selaimineen tarjoaa valmiin maailmanlaajuisen ympäristön DC:n laajalle käyttöönotolle. DC:n elementit voidaan luonnollisesti esittää myös varsinaisten kirjastostandardien edellyttämällä tavalla, esim. MARC-formaatissa.

6.4 Resurssien kuvailun puitteet

Dublin Core aineiston kuvailun muotona on vain yksi näkökulma tietoverkoissa olevien dokumenttien ja muun aineiston käsittelyyn ja hallintaan. Warwickin DC-2:ssa käynnistyi kehitystyö, joka sittemmin on alkanut käyttää itsestään nimitystä RDF (Resource Description Frame-

work)[4]. Hanke on yksi lukemattomista WWW-konsortion käynnistämistä yhteistyöhankkeista, joilla WWW:n kehitystä ohjataan.

RDF:n tavoitteena on muodostaa yhtenäiset puitteet erilaisten ongelmaryhmien hallinnalle, jotka ovat tavallaan eri näkökulmia samoihin tietoverkkojen resursseihin. Näitä näkökulmia ovat resurssien tavoitettavuus, luettelointi ja kuvailu, älykkäiden agenttien käyttö, sisällön arviointi, kokonaisuuksien hallinta, henkisen omaisuuden suoja, jne.

Sisällön arviointi (content rating) on sekä Yhdysvalloissa että EU:n piirissä vakavaksi koettu ongelma, kun lapsia halutaan varjella pornografiselta aineistolta. Kuviin ja teksteihin liitetyt luotettavat arviointitiedot mahdollistaisivat selainten konfiguroinnin niin, että lapset eivät saisi dokumentteja lainkaan katseltavakseen.

Kokonaisuuksien hallinnalla tarkoitetaan useista erillisistä osista muodostuvan yhtenäisen dokumentin sisäisten suhteiden esittämistä. Esimerkiksi kirja saattaa koostua useista luvuista, jotka ovat erillisinä dokumentteina WWW-palvelimessa. Jos käyttäjä kuitenkin haluaisi kirjan yhtenäisenä kokonaisuutena selaimelleen niin, että sisällysluettelo ja sivunumerointi olisivat konaisen teoksen mukaisesti, se vaatii sekä lisätietoja että lisätoimintoja nykyisen kaltaisiin selaimiin.

Henkisen omaisuuden suoja (copyright) liittyy esimerkiksi kuvien kopiointiin ja niiden mahdolliseen hyödyntämiseen kaupallisessa toiminnassa. Kuviin on liitettävä tietoja siitä, kuuluvatko ne mahdollisesti public domain -alueeseen, ovatko ne lainkaan kopioitavissa ja jos ovat, millä ehdoilla niitä voi käyttää omissa julkaisuissaan.

WWW:n selainten tekemisen kannalta on välttämätöntä, että erilaiset RDF:n osat toteutetaan yhtenäisellä tavalla. Tulevaisuuden selaimesta löytyy toimintoja, jotka osaavat näitä osia aina tarpeen mukaan hyödyntää. Yhdestä alasvetovalikosta löytyvät kaikki kaupankäyntiin liittyvät toiminnot, toisessa ovat aineiston sisällön arviointitietoja hyödyntävät toiminnot, kolmannen avulla hallitaan moniosaisia dokumentteja jne. Näyttää siltä, että kaikkien näiden toimintojen kuvaukset WWW-dokumenteissa tulevat perustumaan XML:n käyttöön.

[4]Hankkeen kotisivu on `http://www.w3.org/Metadata/RDF`.

Luku 7

Laitteiden kehitys

Viimeisessä luvussa siirrytään informaation muotoa, tallennusta ja välitystä koskevista standardeista aivan toisenlaisten ongelmien pariin. Lopuksi kysytään, minkälainen laite on järkevin tietoverkkojen käytössä ja minkälaisilla ohjelmilla vähimmillään pystyisi tulemaan toimeen. Tarkoituksena on toisin sanoen pohtia eräänlaista niukkuuden kulttuuria.

Tämän kysymyksen taustalla on se useimpien atk:n hyväksikäyttäjien havaitsema tosiasia, että nykyaikaiset mikrotietokoneet soveltuvat hyvin huonosti käyttötarkoitukseensa. Varsinkin kirjastoissa tilanne on sellainen, että ensin hankitaan monipuoliset ja kalliit laitteet asiakaskäyttöön ja sitten kirjaston mikrotukihenkilöt näkevät hyvin paljon vaivaa siinä, että asiakkaat eivät käyttäisi mikroja mikroina vaan tietoverkon selaimina.

Mikrotietokoneen perusidea on hyvä ja kestävä mutta markkinat ovat vääristäneet sen pahasti. Perusideana oli vapautua ylikuormitetun keskustietokoneen käytöstä ja tehdä yksinkertaiset tehtävät halvalla, henkilökohtaisessa käytössä olevalla laitteella. Tämä perusidea pystyttäisiin nykyisellä teknologialla toteuttamaan sekä taloudellisesti että luotettavasti. Markkinat eivät kuitenkaan toimi tällä tavoin.

Erilaisista syistä johtuen on vähitellen ajauduttu sellaiseen tilanteeseen, että mikrot ovat laitteiden osalta hyvin epäluotettavia, lyhytikäisiä ja usein liian hitaita. Ohjelmia on puolestaan alkanut riivata jonkinlainen mammuttitauti. Niissä on suunnaton määrä ominaisuuksia, joita kukaan ei tarvitse mihinkään. Esimerkiksi MS Wordin komentojen määrä on lisääntynyt vuodesta 1992 lähtien noin 300 sadasta yli 1000:een. Kukaan ei tarvitse päivittäiseen tekstinkäsittelyyn tuollaista määrä toimintoja.

Jo useiden vuosien ajan on vaikuttanut siltä, että mikrotietokoneiden oh-

jelmia kehitetään laajalevikkisten mikroalan lehtien avustajakunnan tarpeisiin. Näiden varmastikin hyvin ammattitaitoisten avustajien tärkein tehtävä on arvioida markkinoille tulevien ohjelmien uusia piirteitä. Avustajan ammattitaitoa mitataan sen mukaan kuinka monta koskaan ennen näkemätöntä uutta piirrettä hän pystyy uudesta ohjelmaversiosta tunnistamaan. Avustajatkaan eivät siis tarvitse uusia piirteitä tekstinkäsittelyssään, he tarvitsevat niitä ainoastaan tekstinkäsittelyä koskevien artikkeleittensa täytteeksi.

7.1 Verkkotietokone

Internetin palvelujen käyttämiseksi mikrotietokone on tarpeettoman monipuolinen, liian kallis ja varsinkin kotikäytössä laitevikojen takia helposti varsinainen “murheen kryyni”. Internetin palveluissa käytettävän laitteen tulisi olla yksinkertainen, helppokäyttöinen, ja lisäksi sen käyttöiän tulisi olla verrattavissa väritelevision käyttöikään. Mikrotietokoneet eivät pääse lähellekään tätä aikaväliä. Mikron tehollinen keskimääräinen käyttöikä on luokkaa 2–4 vuotta. Television vastaava käyttöikä on 10–15 vuotta. Tähän markkinarakoon on tulossa verkkotietokone *Network Computer*.

Apple, IBM, Netscape, Oracle ja Sun esittelivät kesäkuussa 1996 suunnitelman verkkotietokoneesta. Julkistuksen yhteydessä esiteltiin uusi avoin standardi nimeltä *Network Computer Reference Profile*. Sen hyväksikäyttöä on mainittujen organisaatioiden lisäksi päättänyt tukea ainakin 50 muuta yritystä. Mukana on sekä laitevalmistajia että ohjelmistotuottajia. Verkkotietokone on suunniteltu erityisesti Internet-työskentelyä varten ja sen ohjelmat kirjoitetaan Java-kielellä. Ensimmäiset laitteet tulivat myyntiin syksyllä 1996. Täydellä volyymillä laitteiden ja ohjelmien myynnin odotetaan käynnistyvän vuoden 1998 aikana.

Uuden verkkotietokoneen tavoitteena on tehdä Internetin multimediasta yhtä edullista ja helppoa kuin esimerkiksi TV:n katselu tai puhelun soittaminen. Ratkaisun perusajatus on, että ohjelmat ja palvelut ovat verkossa; verkkotietokoneeseen ei erikseen hankita eikä asenneta ohjelmia. Yksittäisen laitteen ylläpito on sen vuoksi äärimmäisen yksinkertaista. Kun verkkotietokone käynnistyy, se hakee tarvitsemansa ohjelmat verkosta. Verkkotietokone toimii käyttäjän näkökulmasta katsottuna kuten WWW:n selausohjelma. Kaikki tarpeelliset ohjelmat käynnistyvät selausojelman avulla, mitään erillistä käyttöjärjestelmää ei tarvita.

Verkkotietokoneessa on tietenkin kuvaputki, näppäimistö ja hiiri, jotka

ovat täsmälleen samoja laitteita kuin mikrotietokoneissa. Lisäksi siinä on prosessori ja riittävästi muistia sekä tietoliikennevalmius. Kiintolevyä tai levykeasemaa ei lainkaan tarvita. Tyypillinen hintaluokka on dollareina 500 USD, markoissa laskettu hinta lienee jossain välillä 2500 – 6000 mk laitekokoonpanosta riippuen. Markkinoille oletetaan tulevan eri tavoin kokoonpantuja ja hinnoiteltuja verkkotietokoneita.

Verkkotietokoneet ovat erityisen käyttökelpoisia sellaisissa tietoverkkojen käyttöön liittyvissä tehtävissä, joissa ei tarvita mikrotietokoneen monipuolisuutta. Esimerkkeinä voidaan mainita kirjastojen asiakaskäyttö ja osittain kirjastotyöntekijöiden ammattimainen käyttö, pankki- ja vakuutussovellukset, ammattimainen tietokantojen ylläpito, koululaisten monet käyttötilanteet jne. Kyseisissä tehtävissä mikrotietokoneiden monipuolisuudesta on enemmänkin haittaa kuin hyötyä. Pidemmällä aikavälillä verkkotietokoneet ovat selvästi edullisempia halvemman ylläpidon ja pidemmän käyttöiän ansiosta.

Vaikka verkkotietokoneet vaikuttavat erittäin lupaavilta, niiden tulevaisuutta on vaikea ennustaa. Käyttökelpoisuutensa ja edullisuutensa takia niitä tullaan kuitenkin mitä todennäköisimmin käyttämään hyvinkin yleisesti ja laitteita tullaan valmistamaan monissa yrityksissä. Verkkotietokoneita pystytään valmistamaan myös Suomessa, jos sellainen nähdään tarpeelliseksi. Kotimainen valmistus voisi turvata riittävän pitkän siirtymäajan verkkotietokoneista johonkin tulevaisuuden järjestelmään. Markkinoiden mielivaltaisuus saattaa jossakin tilanteessa tehdä kysymyksen siirtymäajasta ajankohtaiseksi.

7.2 Verkkotietokoneen ohjelmisto

Verkkotietokoneen ohjelmat kirjoitetaan Java-kielellä, joka on Sun Microsystemsin kehittämä oliopohjainen ohjelmointikieli. Javassa on hyödynnetty SmallTalkin, C:n ja C++:n toteutuksesta ja käytöstä saatuja kokemuksia. Javalle näyttää olleen eräänlainen sosiaalis-teknologinen tilaus. Mikään ohjelmointikieli ei ole koskaan levinnyt käyttöön yhtä nopeasti ja yhtä vähäisten markkinointiponnistusten turvin. Tietojenkäsittelyalan lehdet kirjoittavat Javasta ahkerasti. Javaa käsitteleviä kirjoja on myynnissä useita satoja. Myös suomen kielistä kirjallisuutta on saatavana ja kurssitarjonta on vilkasta.

Javan laajamittainen käyttöönotto sisältää monia kiehtovia mahdol-

lisuuksia, jotka parhaimmillaan saattavat kokonaan mullistaa vakiintuneet ohjelmistokaupan toimintatavat. Tämän seikan ymmärtämiseksi on lyhyesti kerrattava tietojenkäsittelyä koskevan liiketoiminnan painopisteen muutokset muutaman vuosikymmenen ajalta.

Tietokoneiden kaupan alkukausi 1970-luvulle saakka painottui itse tietokoneisiin fyysisinä laitteina. Tietokoneet olivat isoja, tavattoman kalliita ja hitaita. Yliopistoihin ja tutkimuslaitoksiin hankitut koneet maksoivat useita miljoonia markkoja, joissakin tapauksissa kymmeniä miljoonia. Erikseen hankitut ohjelmistot maksoivat yleensä murto-osan tietokoneen hinnasta. Käyttäjät ja sovellukset olivat vahvasti sidoksissa laitemerkkeihin. Myös julkisen vallan ohjaus oli merkkisidonnaista. Valtiovarainministeriö päätti, minkä merkkisiä laitteita sai käyttää.

Vähitellen koneet ovat halventuneet ja samalla tehostuneet. Muutamalla sadalla tuhannella markalla saa nykyisin useimpiin käyttötarkoituksiin riittävän tehokkaan tietokoneen. Hinnan alhaisuudesta johtuen tietokoneiden kauppa on jossakin mielessä menettänyt hohtoaan. Maahantuojien maksamat pitkät ulkomaanmatkat ovat lähes kokonaan loppuneet. Ohjelmistot ovat vallanneet tietokoneiden aseman sekä taloudellisessa että toiminnallisessa mielessä. Muutos on näkynyt myös ohjaustoimissa. Monien hallinnollisten järjestelmien osalta valtiovallan ohjaus 1990-luvun puolivälin tienoille saakka koski ohjelmien valintaa. Tietokoneen on saanut ostaa miltä valmistajalta tahansa.

Mutta myös ohjelmistojen asema on alkanut vähitellen järkkyä. Muutos ei ole vielä kovin vahva mutta se on kuitenkin selvästi havaittavissa. Tyytymättömyys amerikkalaisia ohjelmistojen valmistajia kohtaan on jatkuvasti lisääntynyt. Tämä koskee sekä mikrotietokoneiden ohjelmia että suurten tietokoneiden kalliita erikoisohjelmistoja. Samalla käyttäjäorganisaatioiden huomio on yhä enemmän siirtynyt tiedonhallintaan ja tietovarantojen hyötykäyttöön. Huomio toisin sanoen keskittyy tietokoneella käsiteltävään informaatioon, erityisesti sen tallennus-, siirto- ja esitysmuotoihin. Se, millä tietokoneilla tai ohjelmilla informaatiota käsitellään, ei ole enää merkityksellistä. Muutos näkyy myös virallisessa ohjauksessa. Valtiovarainministeriö tai valtiokonttori eivät enää päätä esimerkiksi kirjanpito-ohjelman tuotemerkistä. Riittää, että käytetty ohjelma tuottaa valtiokonttorin ja postipankin käyttöön tarkoitetun informaation oikeassa muodossa.

Tähän päättelyketjuun kätkeytyy Javan kumouksellisuuden siemen. Tulevaisuudessa tiedon tarvitsija ei enää lainkaan kiinnitä huomiota käyttämäänsä laitteeseen eikä varsinkaan hanki ja asenna siihen yhtään ohjelmaa.

Käyttäjä selaa erilaisia tietoverkon palveluja ja perehtyy johonkin niistä tarkemmin aina kulloisenkin mielenkiintonsa ja tarpeensa mukaan. Jos tarkempi perehtyminen edellyttää jonkin nimenomaisen ohjelman käyttöä, selain hakee ja käynnistää kyseisen ohjelman. Käyttäjä ei välttämättä edes havaitse käytössä olevien ohjelmien vaihtumista. Ohjelmista ei koskaan makseta erikseen, ne kuuluvat osaksi käytetyn tietopalvelun kokonaisuutta. Palvelun tarjoaja huolehtii ohjelmien hankinnasta, asennuksista ja käyttöön sovittamisesta.

Kun tulevaisuudessa tietokoneohjelmia tekevät, teettävät tai ostavat ensisijaisesti tietopalvelujen tarjoajat, ohjelmat pakostakin muuttavat luonnettaan. Niistä häipyy kokonaan markkinoille hyvin luonteenomainen rihkama. Tietopalveluja tulee olemaan tietoverkoissa tarjolla yllin kyllin. Palvelua ei varmastikaan tulla valitsemaan sillä perusteella, että käytetty ohjelma on aivan erityisen vaikeakäyttöinen ja täynnä virheitä. Ohjelmien on pakko olla käyttötarkoitukseensa hyvin soveltuvia ja virheettömiä.

Tämä tulee viemään kokonaan pohjan pois mm. nykyisen kaltaisten tekstinkäsittelyohjelmien markkinoilta. Useimmin käytetyt ohjelmat tulevat yksinkertaistumaan ja halpenemaan. Niitä ei erillisinä tuotteina myydä enää lainkaan. Osa ohjelmista tulee väistämättä muuttumaan julkisohjelmiksi, jolloin niiden virheettömyys, toimintavarmuus ja jatkuva ylläpito pystytään turvaamaan. Tämä sama kehitys on jo nähtävissä käyttöjärjestelmien osalta. Ammattimaisessa käytössä tavallisin käyttöjärjestelmä on Unix, jonka saa mille tahansa tietokoneelle. Valtaosa Unixin ohjelmista on julkisia ja paras Unix-toteutus (Linux) on kokonaisuudessaan julkisohjelma.

Tekstinkäsittelyn kehityksen nopeuttamiseksi olisi järkevintä, jos Euroopassa tuotettaisiin julkisilla varoilla hyvä Java-kielinen tekstinkäsittelyohjelma, jota levitettäisiin lähdekielisenä veloituksetta maailmanlaajuisesti. Se lopettaisi tarpeettoman rahojen haaskaamisen kaupallisiin ohjelmiin, lopettaisi vaivalloisen ohjelmien päivityskierteen, synnyttäisi laadukkaamman tuotteen ja helpottaisi monin tavoin tavallisten käyttäjien elämää. Tällä operaatiolla Eurooppa säästäisi valtavan summan rahaa muihin tarkoituksiin. Jokin Euroopan ulkopuolinen taho samalla kyllä menettäisi yhtä suuren rahasumman.

Lukija voi mielessään pohtia, onko tekemäni ehdotus järkevä ja onko käytettävissä mitään tapaa, jolla se voitaisiin toteuttaa. Itse en ole sellaista keksinyt. Niinpä odottelen yhteydenottoja vaikkapa sähköpostin välityksellä.

Kirjallisuutta

[1] Advisory Committee for the Co-ordination of Information Systems (ACCIS). *The Internet. An Introductory Guide for United Nations Organizations*. United Nations, Geneva, 1994.

[2] ANSI/NISO. ANSI/NISO Z39.50–1995, *Information Retrieval (Z39.50): Application Service Definition and Protocol Specification*. Z39.50 Maintenance Agency (Library of Congress), July 1995. Official Text for Z39.50–1995.

[3] Eliot Christian. The government information locator service (GILS). *Information Services and Use*, 16:25–42, 1996.

[4] Douglas E. Comer. *Internetworking with TCP/IP. Principles, Protocols, and Architecture*. Prentice-Hall, Englewood Cliffs, N.J., 1988.

[5] Ray Denenberg. Recent developments and future prospects. Z39.50 Seminar at the Royal Library of Belgium. Available from: <ftp://ftp.loc.gov/pub/z3950/articles/kbr.ps>, September 30, 1996.

[6] D. Kaye Gapen. *The Virtual Library: Knowledge, Society, and the Librarian*, pages 1–14. In Saunders [25], 1993.

[7] Charles F. Goldfarb. *The SGML Handbook*. Clarendon Press, Oxford, 1994.

[8] Eric van Herwijnen. *Practical SGML*. Kluwer Academic Publishers, Dordrecht, second edition, 1994.

[9] Mark Hinnebusch. Integrated Library Systems. A Primer on Z39.50. *Academic and Library Computing*, 9(2–10), 1992. [Series of articles].

[10] Kalervo Järvelin. *Tekstitiedonhaku tietokannoista. Johdatus periaatteisiin ja menetelmiin*. Suomen atk-kustannus, Espoo, 1995.

[11] Brewster Kahle, Harry Morris, Franklin Davis, Kevin Tiene, Clare Hart, and Robin Palmer. Wide Area Information Servers: An executive information system for unstructured files. *Electronic Networking: Research, Applications and Policy*, 2(1):59–68, Spring 1992.

[12] Ed Krol. *The Whole Internet. User's Guide & Catalog*. O'Reilly & Associates, Inc., Sevastopol, CA, second edition, 1994. [Saatavana myös suomeksi].

[13] Timo Kuronen. Euroopan tie tietoyhteiskuntaan. Bangemannin raportin esittely. *Tiedotustutkimus*, 18(1):26–36, 1995.

[14] Timo Kuronen. *Ranganathanin lait ja virtuaalikirjasto*. Finnish Information Studies 4, Tampere – Åbo – Oulu, 1996.

[15] *Libraries Workprogramme 1994–1998. Discussion Document.* European Commission DGXIII/E-3, January 1994. Draft. Version 2.

[16] James J. Michael and Mark Hinnebusch. *From A to Z39.50. A Networking Primer*. Mecklermedia, Westport, CT, 1995.

[17] William E. Moen and Charles R. McClure. *The Government Information Locator Service (GILS): Expanding Research and Development on the ANSI/NISO Z39.50 Information Retrieval Standard. Final Report*. NISO Press, Gaithersburg, MD, 1994.

[18] William E. Moen and Charles R. McClure. *An Evaluation of the Federal Government's Implementation of the Government Information Locator Service (GILS). Final Report*. GPO, Washington, DC, 1997.

[19] Olli Mustajärvi. Eduskunnan hankkeet. Teoksessa Sinikka Kangas ja Leena Karjalainen, toim., *SGML-seminaari 12.12.1994 ja 14.12.1994*, s. 19–2. Eduskunnan kirjasto, Helsinki, 1995.

[20] Ilkka Niiniluoto. *Informaatio, tieto ja yhteiskunta. Filosofinen käsiteanalyysi*. Edita, Helsinki, 5., täydennetty painos, 1996.

[21] Katia Obraczka, Peter B. Danzig, and Shih-Hao Li. Internet resource discovery service. *Computer*, 26(9):8–22, September 1993.

[22] Airi Salminen, Merja Lehtovaara, and Katri Kauppinen. Standardization of digital legislative documents. A case study. In *Proceedings of the 29th Hawaii International Conference on System Sciences*, pages 72–81. IEEE Computer Society Press, 1996.

[23] Peter H. Salus. *A Quarter Century of UNIX*. Addison-Wesley, Reading, Mass., 1994.

[24] Peter H. Salus. *Casting the Net. From ARPANET to Internet and Beyond*. Addison-Wesley, Reading, Mass., 1995.

[25] Laverna M. Saunders, editor. *The Virtual Library, Visions and Realities*. Meckler, Westport, CT, 1993.

[26] Michael F. Schwartz. Internet resource discovery at the University of Colorado. *Computer*, 26(9):25–35, September 1993.

[27] Majid Tehranian. *Technologies of Power. Information Machines and Democratic Prospects*. Ablex, Norwood, N.J., 1990.