

Rakenteisten dokumenttien tutkimuksesta

Pekka Kilpeläinen
Kuopion yliopisto
Tietojenkäsittelytieteen ja sovelletun matematiikan laitos

Virkaanastujaisesityelmä 29. huhtikuuta 2002

Kirjoitustaidon vakiintuminen aloitti historiallisen ajan, mistä lähtien ihminen on taltioinut ja viestinyt tietojaan kirjallisessa muodossa erilaisina *dokumentteina*. Perinteisesti dokumenttien tuottajana, vastaanottajana ja tuloksijana on ollut ihminen. Dokumenteista saatava hyöty kasvaa kuitenkin uusiin mittoihin, kun niitä aletaan käsitellä automaattisesti, tieto- ja tietoliikennetekniikan keinoin. Tästä on paljolti kyse suurten odotusten kohteena olevassa sähköisessä kaupankäynnissä eli ”e-bisneksessä”: Kuinka automatisoida tietoverkossa välitettävien, liiketapahtumiin liittyvien dokumenttien kuten tilausten ja laskujen käsittely?

Automaattinen käsittely edellyttää tiedon esittämistä *rakenteisena* eli jossain kontrolloidussa määrämuodossa. Yksinkertaisimmillaan *rakenteisissa dokumenteissa* on kyse dokumentin merkityksellisten osien (kuten kappaleiden, otsakkeiden ja henkilönimien) osoittamisesta sovitulla merkinnöillä, ns. *merkkauksella*.

XML-esitystapa eli *Extensible Markup Language*, suorasanaisesti käännettynä ”laajennettavissa oleva merkkauksikieli”, on nykyään tietotekniikka-alan kuumimpia aiheita. Tämä johtuu muun muassa sen sopivuudesta edellä mainitun sähköisen kaupankäynnin tarpeisiin.

Miltä tällainen rakenteinen XML-dokumentti sitten näyttää? XML-esitystavan perustana on dokumentin merkityksellisten osien, ns. *elementtien* osoittaminen dokumenttiin sisältyvin tunnistein. Kuvassa 1 näemme yksinkertaisen esimerkin laskun sisältöä kuvaavasta XML-dokumentista. Lasku-elementin laajuus on osoitettu vastaavin *alku-* ja *lopputunnistein* (riveillä 1 ja 9). Näiden tunnisteiden ohjaamina dokumentin merkitykselliset osat kuten esimerkiksi vastaanottajan tiedot, nimi ja osoite pystytään helposti tunnistamaan ja eristämään automaattista käsittelyä varten.

XML kiinnitettiin kansainväliseksi teollisuusstandardiksi vuonna 1998; tarkkaan ottaen XML on kansainvälisen Web-konsortion tuolloin julkaisema

```

1. <lasku>
2.   <vastaanottaja>
3.     <nimi>Pekka Kilpeläinen</nimi>
4.     <osoite>Kotikatukuja 123</osoite>
5.   </vastaanottaja>
6.   <tuote>CD-levy</tuote>
7.   <hintaa>20 EUR</hintaa>
8.   ...
9. </lasku>

```

Kuva 1: Esimerkki rakenteisesta XML-dokumentista

suositus [1]. Rakenteisia dokumentteja on tutkittu Suomen yliopistoissa, esimerkiksi Kuopiossa ja Helsingissä, kuitenkin jo toistakymmentä vuotta eli vuosia ennen nykyistä XML-huumaa. Mitä tutkittavaa rakenteisissa dokumenteista sitten on, ja mitä hyötyä niiden tutkimisesta on? Yritän valaista asiaa muutamalla esimerkillä tutkimuksista, joihin itselläni on ollut tilaisuus osallistua.

Helsingin yliopistossa kehitettiin 1990-luvulla merkatun tekstitiedon sisällön etsintään soveltuva malli [2] — ei toki tyhjästä vaan aiempaan kansainväliseen tutkimukseen nojautuen [3, 4, 5]. Kyseisessä mallissa, ns. *tekstialuealgebra*, on kyse tekstitiedon käsittelemisestä yhtenäisten sisältökatkelmien muodostamina joukkoina. Kyseessä on algebra siksi, että mallin operaatiot käsittelevät ja tuottavat ainoastaan tällaisia tekstikatkelmien joukkoja. Ominaisuus on hyödyllinen: sen nojalla mallin ilmaisuja voi yhdistellä vapaasti keskenään, mikä mahdollistaa rajoittamattoman monimutkaisten etsintäehtojen kuvaamisen.

Yksinkertaisena esimerkkinä tekstialuemallin käytöstä tiedonhakuun voidaan tarkastella seuraavaa tilannetta: Käytössä on merkatuista tekstikappaleista muodostuva teksti, josta halutaan löytää sanan ”pop” sisältävät kappaleet (kuva 2). Tiedonhaun lähtökohtana ovat hakutehtävän kuvaamiseen tarvittavat merkkijonot, tässä tapauksessa kappale-elementtien alku- ja loppu-tunnisteet ”<kp1>” ja ”</kp1>” sekä sana ”pop”. Merkkijonojen esiintymät tekstissä ovat mallin käsittelemiä merkkipositiodien rajaamia tekstialueita: alkutunnistetta ”<kp1>” vastaa aluejoukko {[0,4], [23,27]}, lopputunnistetta ”</kp1>” vastaa aluejoukko {[17,22], [39,44]}, ja merkkijonoa ”pop” vastaa aluejoukko {[35,37]}. Näistä voidaan mallin operaattoreilla muodostaa tekstialueiden rajaamia laajempia alueita. Esimerkiksi tässä tapauksessa tekstikappaleita esittävien elementtien alueet saadaan muodostettua ilmauksella

”<kp1>” .. ”</kp1>” ,

Teksti: <kpl>Punk on out.</kpl><kpl>XML on pop.</kpl>
Positiot: 0 4 17 23 27 35 39 44

Kuva 2: Esimerkki tekstialuealgebralla etsittävästä tekstistä

jonka arvo on aluejoukko $\{[0,22], [23,44]\}$. Lopulta muodostettuja aluejoukkoja voi rajoittaa niiden alueiden välisen sisältyvyyden perusteella. Tässä esimerkissä haluttu tulos $\{[23,44]\}$ saadaan aikaan rajoittamalla niihin kappalealueisiin, jotka sisältävät ”pop”-sanan jonkin esiintymän. Tämä voidaan ilmaista seuraavalla lausekkeella:

"<kpl>" .. "</kpl>" containing "pop" .

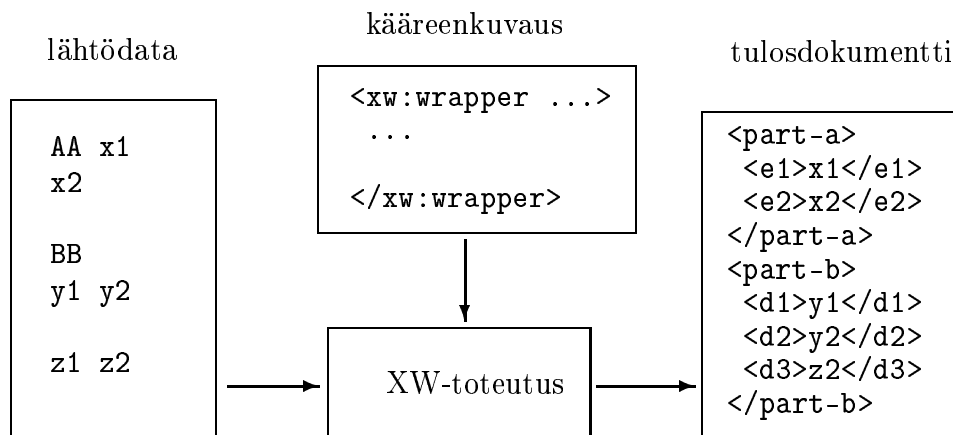
Tekstialuealgebra onnistuttiin toteuttamaan tehokkaaksi etsintäohjelmaksi nimeltä *sgrep* [6], joka saavutti maailmallakin suosiota: sitä on sovellettu mm. web-sivustojen paikallisena hakukoneena [7, Chapter 7], ja se on levinnyt erään Linux-ohjelmistopakettiin [8] mukana ehkä jopa miljooniin tietokoneisiin.¹

Monien toimialojen tiedonsiirrossa ollaan siirtymässä standardoituihin, XML-esitystapaan perustuviin tiedonesitysmuotoihin. Tällaisia on kehitetty mm. terveydenhoitoalan ja sähköisen kaupankäynnin tarpeisiin [9, 10]. XML-muotoisten viestien tuottamiseksi perinteisiä tiedonesitysmuotoja on kyettävä muuntamaan rakenteiseen, eksplisiittisesti merkattuun dokumenttimuotoon. Tällaista muuntamista tehdään tyypillisesti varta vasten kirjoitetuilla tietokoneohjelmilla tai järjestelmien rajapintoihin liitetyillä komennoilla – molempien kehittäminen ja ylläpitäminen on kohtalaisen vaativaa ja työlästä. Muunnosten helpottamiseksi Kuopion yliopistossa on kehitetty TEKE-Sin ja paikallisten yritysten rahoittamassa XML-rajapintojen kehittämistä tutkivassa *XRAKE*-hankkeessa ns. *XML-käärintäkieli* nimeltä *XW* [11].

XW on lyhennys sanoista ”*XML wrapper*” eli XML-kääre. Kyseessä on yksinkertainen ilmaisutapa, jolla kuvataan XML-muotoon muunnettavan lähtötiedon sisältörakenne. Tämän käärekuvauksen perusteella kielen toteutus osaa muuntaa lähtötiedon sitä vastaavaksi XML-dokumentiksi automaattisesti. Kuvassa 3 on esitetty *XW*-kielen perusajatus: Kielen toteutus saa syötetietoinaan XML-muotoon muunnettavan lähtöaineiston sekä *XW*-kielisen kääreenkuvauksen. Kääreenkuvaus koostuu XML-tulosdokumentin mallista

¹Tämä ei välttämättä tarkoita, että miljoonat tai edes tuhannet ihmiset todella käyttäisivät ohjelmaamme. Ohjelmistopaketit voivat sisältää tuhansia ohjelmia, joista normaalkäyttäjät tarvitsee ehkä vain muutamia. Eri puolilta maailmaa tipahtelee joka tapauksessa silloin tällöin kiitoksia ja kommentteja *sgrep*-ohjelman käyttäjiltä.

ja siihen liittyvästä lähtömuodon rakenteen kuvauksesta. Näiden perusteella XW-toteutus tuottaa kuvauksen mukaisia XML-dokumentteja – poimien lähtötiedoista haluttuja osia ja ryhmittelemällä ne kuvauksen mukaisiksi elementeiksi.



Kuva 3: XW-järjestelmän arkkitehtuuri

Rakenteiset dokumenttiformalismit sisältävät myös esitystapoja dokumentteja kontrolloiville *rakennekuvauksille*. XML-spesialistit tuntevat tällaiset rakennekuvaukset esimerkiksi *dokumenttityypinmäärityksinä* [1] tai XML Schema -kielellä esitettyinä *rakennekaavioina* [12]. Rakennekuvauksilla voidaan määrätä, mitä ja millaisia osia dokumenteissa saa tai täytyy olla. Jos esimerkiksi rahaliikenteeseen tai potilaiden hoitoon vaikuttavia viestejä käsitellään automaattisesti, on ilmeisen tärkeää pystyä varmistamaan, että käsiteltävät dokumentit sisältävät täsmälleen vaaditut tiedot täsmälleen oikeanmuotoisina.

Millaisia nämä rakennekuvaukset sitten ovat? Kuvassa 4 näemme katkelman dokumenttityypinmäärityksestä, joka kuvaa kirjojen lukujen yksinkertaistettua rakennetta. Sovellettavat sisältömallit perustuvat sisältöosien peräkkäisyyden, valinnaisuuden ja toisteisuuden kuvaamiseen. Esimerkiksi tässä kuvataan, että luvuilla tulee olla otsikko, ja otsikkoa voi seurata rajoittamaton jono kuva- tai tekstikappale-elementtejä. Kuva-elementeistä määrätään edelleen, että ne sisältävät grafiikka-elementin ja kuvatekstin.

Rakennekuvausten ohjaamina sovellukset voivat käsitellä säädetyn muotoisia dokumentteja luotettavasti. Tietojenkäsittelytieteen kannalta nämä dokumenttien rakennekuvaukset ovat variaatioita mm. ohjelmointikielten ja luonnollisten kielten käsittelyyn sovelletuista *kieliopeista* [13, 14].

Dokumenttien rakennekuvausten vertaileminen on Kuopion yliopistossa

```
<!ELEMENT luku (otsikko, (kuva | tekstikappale)*)>
<!ELEMENT kuva (grafiikka, kuvateksti)>
```

Kuva 4: Esimerkki XML-dokumentin rakennekuvauksesta

uusi, käynnistelyn alla oleva tutkimusaihe. Samoja dokumentteja saatetaan kuvata eri yhteyksissä erilaisilla rakennekuvauksilla. Esimerkiksi dokumentteihin saatetaan organisaation sisällä liittää enemmän tietoa kuin on käytettävissä organisaatioiden väliseen tiedonsiirtoon sovelletussa rakennekuvauksessa. Tällöin tarvitaan keinoja verrata dokumenttien rakennekuvauksia keskenään: Voidaanko varmistua, että organisaation sisäiseen rakennekuvaukseen perustuvat sovellukset pystyvät käsittelemään kaikki saapuvat, ulkoisen rakennekuvauksen mukaiset dokumentit?

Tehtävä ei ole triviaali: Kieliopilliset kuvaukset, kuten dokumenttien rakennekuvaukset, esittävät yleensä *ääretöntä* joukkoa dokumentteja. Lisäksi pinnallisesti hyvinkin eri näköiset kieliopilliset ilmaisut voivat tarkoittaa pohjimmiltaan samaa asiaa. Ongelma vaikuttaisi kuitenkin ratkeavan perinteisen formaalikielten teorian menetelmin: XML-rakennekuvausten vertailu palautuu niitä vastaavien yksinkertaisten laskentamallien, ns. *äärellisten automaattien* vertailemiseen, johon tunnetaan tehokkaita menetelmiä. Toisaalta ongelma ei vaikuta ratkeavan pelkkänä tunnetun tiedon sovelluksena. XML-rakennekuvaukset eivät vastaa perinteisesti tutkittuja kielioppeja *täsmälleen*. Ne sisältävät sekä käsittelyä helpottavia rajoituksia että toisaalta laajennuksia perinteisiin kielioppeihin. Erityisesti XML Schema-kaaviokielen eräiden (järjestämätöntä sisältöä kuvaavien) ilmaisujen tehokas vertaileminen vaikuttaa mielenkiintoiselta haasteelta.

Esittämäni esimerkit toivottavasti valaisevat sitä, että rakenteisiin dokumentteihin liittyy kiinnostavaa tutkittavaa, ja sitä, että tietojenkäsittelijöillä on arvokasta annettavaa rakenteisten dokumenttien käytännön soveltajille. Parhaimmillaan soveltava tietojenkäsittelytiede tuottaa hyvin ymmärrettyyn teoriaan perustuvia menetelmiä, jotka ovat toteutettavissa käytännöllisiä tarpeita palvelevina sovelluksina. Tällaiseen on pyrittävä Kuopion yliopistossa jatkossakin.

Viitteet

- [1] T. Bray, J. Paoli, and C.M. Sperberg-McQueen, editors. *Extensible Markup Language (XML) 1.0*. W3C Recommendation, February 1998.
- [2] J. Jaakkola and P. Kilpeläinen. Nested text-region algebra. Technical Report C-1999-2, University of Helsinki, Department of Computer

Science, January 1999.

- [3] A. Salminen and F. Wm. Tompa. PAT expressions: an algebra for text search. Technical Report OED-92-02, UW Centre for the New Oxford English Dictionary and Text Research, 1992.
- [4] F.J. Burkowski. An algebra for hierarchically organized text-dominated databases. *Information Processing & Management*, 28(3):333-348, 1992.
- [5] C.L.A. Clarke, G.V. Cormack, and F.J. Burkowski. An algebra for structured text search and a framework for its implementation. *The Computer Journal*, 38(1):43-56, 1995.
- [6] J. Jaakkola and P. Kilpeläinen. Using sgrep for querying structured text files. In J. Saarela, editor, *Proceedings of SGML Finland 1996*, pages 56-67. SGML Users' Group Finland, 1996.
- [7] M. Leventhal, D. Lewis, and M. Fuchs. *Designing XML Internet Applications*. Prentice Hall Inc., Englewood Cliffs, New Jersey, 1998.
- [8] Debian GNU/Linux homepage. <http://www.debian.org/>, 2002.
- [9] L. Alschuler, R.H. Dolin, and S. Boyer. Clinical Document Architecture framework; version 1.0 draft. Membership Ballot at the HL7 USA Web site, August 2001.
- [10] Electronic business XML (ebXML) home page. <http://www.ebxml.org>, 2002.
- [11] M. Ek, H. Hakkarainen, P. Kilpeläinen, E. Kuikka, and T. Penttinen. Describing XML wrappers for information integration. In *Proceedings of XML Finland 2001*, pages 38-51, Tampere, Finland, November 2001.
- [12] H.S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn, editors. *XML Schema Part 1: Structures*. W3C Recommendation, May 2001.
- [13] P. Kilpeläinen and D. Wood. SGML and XML document grammars and exceptions. *Information and Computation*, 169:230-251, 2001.
- [14] D. Lee, M. Mani, and M. Murata. Reasoning about XML schema languages using formal language theory. Technical report, IBM Almaden Research Center, November 2000.