Rakenteisten dokumenttien koostamismalli ja koostamisjärjestelmä SAW

Barbara Heikkinen

Helsingin yliopisto
Tietojenkäsittelytieteen laitos
Barbara.Heikkinen@cs.Helsinki.FI

Barbara Heikkinen on työskennellyt vuodesta 1995 lähtien Helsingin yliopiston tietojenkäsittelytieteen laitoksella rakenteisten dokumenttien hallinnan tutkimusryhmässä tutkijana ja tohtorikoulutettavana. Hän on ollut mukana kesällä 1998 päättyneessä Tekes-rahoitteisessa Älykkäät ja rakenteiset dokumentit (Structured and Intelligent Documents, SID) -tutkimushankeessa. Heikkisen keskeisenä tutkimusalueena on dokumenttirakenteiden hallinta SGML-elementtien automaattisen luokittelun avulla sekä uusien dokumenttien koostaminen useiden eri dokumenttien osista.

Oskari Heinonen

Helsingin yliopisto
Tietojenkäsittelytieteen laitos
PL 26 (Teollisuuskatu 23)
00014 Helsingin yliopisto
Puh. (09) 70 851
Fax (09) 7084 4441
Oskari.Heinonen@cs.Helsinki.FI
http://www.cs.helsinki.fi/~oheinone/

Oskari Heinonen on työskennellyt vuodesta 1995 Helsingin yliopiston tietojenkäsittelytieteen laitoksella tohtorikoulutettavana ja rakenteisten dokumenttien hallinnan tutkimusryhmän tutkijana osallistuen SID-hankkeeseen. Hänen tutkimusalueensa on pitkien tekstidokumenttien jakaminen pienemmiksi yhtenäisiksi kokonaisuuksiksi, joita voidaan käyttää uusien dokumenttien koostamisessa.

Jani Jaakkola

Helsingin yliopisto
Tietojenkäsittelytieteen laitos
Jani.Jaakkola@cs.Helsinki.FI

Jani Jaakkola on työskennellyt vuodesta 1995 Helsingin yliopiston tietojenkäsittelytieteen laitoksen rakenteisten dokumenttien hallinnan tutkimusryhmässä ja osallistunut SID-hankkeessa erityisesti rakenteisen tekstin etsintäohjelman *sgrep* ja SGML-dokumenttien muunnoskielen *TranSID* suunnittelu- ja toteutustyöhön. Hän on tietojenkäsittelytieteen perustutkinto-opiskelija.

Pekka Kilpeläinen

Helsingin yliopisto
Tietojenkäsittelytieteen laitos
Pekka.Kilpelainen@cs.Helsinki.FI

Pekka Kilpeläinen valmistui Helsingin yliopiston tietojenkäsittelytieteen laitokselta filosofian tohtoriksi vuonna 1993. Hänen väitöskirjansa käsitteli puuhahmonsovitusalgoritmeja ja niiden soveltamista tiedonhakuun rakenteisista tekstitietokannoista. Vuodesta 1995 vuoteen 1998 hän toimi projektipäällikkönä SID-hankkeessa. Hän on myös ollut keskeisessä roolissa *sgrep*-ohjelman suunnittelussa.

Greger Lindén

Helsingin yliopisto
Tietojenkäsittelytieteen laitos
Greger.Linden@cs.Helsinki.FI

Greger Lindén valmistui Helsingin yliopiston tietojenkäsittelytieteen laitokselta filosofian tohtoriksi vuonna 1997. Hänen väitöskirjansa käsitteli rakenteisten dokumenttien muunnoksia. Hän osallistui tutkijana SID-hankkeeseen. Tällä hetkellä hän

hoitaa tietojenkäsittelytieteen laitoksen yliassistentuuria.

Jyrki Niemi

Helsingin yliopisto
Yleisen kielitieteen laitos
Jyrki.Niemi@Helsinki.FI

Jyrki Niemi on perustutkinto-opiskelija Helsingin yliopistossa, pääaineenaan tietokonelingvistiikka. Hän työskenteli vuonna 1997-98 SID-hankkeessa atk-suunnittelijana ja osallistui erityisesti dokumenttien ryvästysohjelman toteuttamiseen. Tällä hetkellä hän on atk-suunnittelijana Helsingin yliopiston yleisen kielitieteen laitoksella EU:n kieliteknologiaprojektissa MIETTA (Multilingual Information Extraction for Tourism and Travel Assistance).

Kimmo Paasiala

Helsingin yliopisto
Tietojenkäsittelytieteen laitos
Kimmo.Paasiala@cs.Helsinki.FI

Kimmo Paasiala on tietojenkäsittelytieteen perustutkinto-opiskelija Helsingin yliopistossa. Hän työskenteli suunnittelijana SID-hankkeessa vuonna 1997-98 ja osallistui SAW-koostamisjärjestelmän osana olevan ryvästysohjelman toteutukseen. Tällä hetkellä hän tekee pro gradu -työtä ryvästysalgoritmeista.

Tiivistelmä

Tietotekniikan kehittyminen on tehnyt erilaisten tekstidokumenttien levittämisen, tallennuksen ja haun huomattavasti aiempaa helpommaksi. Kirjoittamisen apuvälineet eivät kuitenkaan ole kehittyneet läheskään samaan tahtiin. Toki tekstinkäsittelyjärjestelmät antavat mahdollisuuden joustavaan tekstien kirjoittamiseen, mutta varsinaiseen tekstimateriaalin kokoamisprosessiin niistä on vain niukalti apua. Esittelemme tässä paperissa dokumenttien koostamismallin ja kokeellisen koostamisjärjestelmän. Dokumenttien koostamisella tarkoitamme uuden dokumentin tietokoneavusteista muodostamista useita olemassa olevia tekstilähteitä hyväksi käyttäen. Koostaminen on luonteeltaan interaktiivinen prosessi, jossa kirjoittaja tai koostaja käyttää erilaisia apuvälineitä löytääkseen tarkoitustaan vastaavia lähteitä ja muodostaakseen niistä soveltuvan kokonaisuuden.

1 Johdanto

Dokumentin käsite on viime vuosina muuttunut dramaattisesti. Aikaisemmin dokumentit olivat staattisia, selkeitä kokonaisuuksia ja usein yhden kirjoittajan tuottamia. Useimmiten ne myös päätyivät jossakin vaiheessa sellaisenaan paperille. Tänä päivänä dokumentit ovat dynaamisia ja aktiivisia olioita, joita ei ole ehkä lainkaan olemassa ennen kuin ne luodaan keräämällä katkelmia eri kokoelmista ja yhdistelemällä katkelmista uusia kokonaisuuksia.

SGML ja muut dokumenttistandardit ovat mahdollistaneet dokumenttien tehokkaan luomisen, talletuksen, muotoilun ja kyselyt yhden dokumenttityypin sisällä. Sen sijaan dokumenttien osien tehokas uudelleenkäyttö ja suurten heterogeenisten dokumenttikokoelmien hallinta ovat edelleen avoimia kysymyksiä.

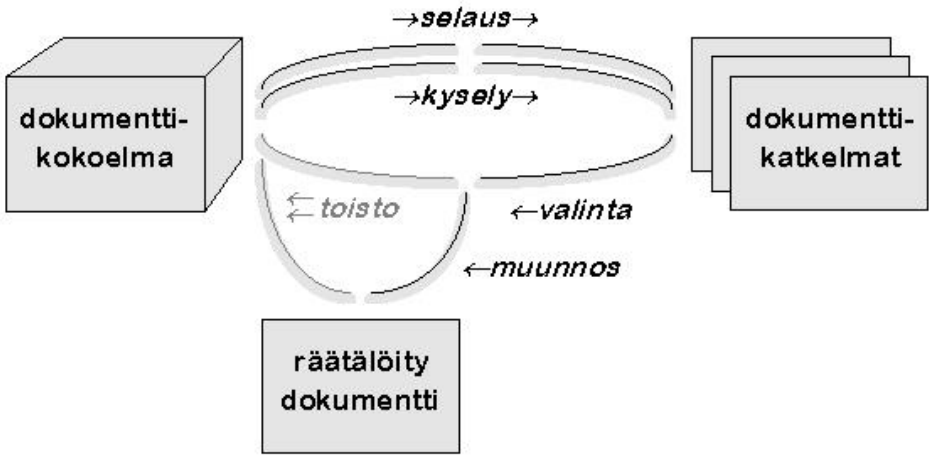
Esittelemme tässä artikkelissa Helsingin yliopiston tietojenkäsittelytieteen laitoksen Älykkäät ja rakenteiset dokumentit -projektin tuloksia. Olemme kehittäneet dokumenttien koostamismallin ja -arkkitehtuurin sekä tutkineet niiden avulla SGML-dokumenttien älykästä uudelleenkäyttöä. Olemme myös toteuttaneet järjestelmän prototyypin ja testanneet lähestymistapaa erilaisilla dokumenteilla, muun muassa lakiaineistolla ja oppikirjoilla.

Tämä teksti jakautuu lukuihin seuraavasti. Luvussa 2 käsitellään yleisesti dokumenttien koostamista ja esitellään koostamismallimme. Luvussa 3 keskitytään dokumenttien esikäsittelyyn, joka on tarpeen ennen dokumenttien lataamista koostamisjärjestelmäämme. Itse koostamisjärjestelmä esitellään luvussa 4, ja luku 5 on lyhyt yhteenveto.

2 Dokumenttien koostaminen

Dokumenttien koostamisella (document assembly) tarkoitetaan uuden dokumentin tietokoneavusteista muodostamista useita olemassa olevia tekstilähteitä hyväksi käyttäen. Kehittämässämme dokumenttien koostamismallissa lähdemme siitä, että koostaminen on vuorovaikutteinen prosessi, jossa käyttäjän, »toimittajan», valinnat ohjaavat automaattisia tiedonhaku- ja koostamisvaiheita. Koostamismalli on esitetty kuvassa 1. Koosteen käyttötarkoitus vaikuttaa siihen, miten paljon vuorovaikutteisuutta tarvitaan. Kun opettaja koostaa oppikirjaa, hän todennäköisesti haluaa ohjata prosessia enemmän kuin esimerkiksi toimittaja, joka haluaa kerätä nopeasti arkistosta tai uutisvirrasta faktoja juttunsa pohjaksi.

Kuva 1: ***Koostamismalli***



Koostamismallimme sisältää relevanttien tekstikatkelmien etsimisen, katkelmien muokkaamisen ja järjestämisen sekä muokattujen katkelmien yhteenliittämisen uudeksi dokumentiksi. Katkelmien etsimiseen käytetään kyselyitä ja selausta. Lisäksi dokumenttikokoelma tai jo valittu osakokoelma voidaan ryvästää eli klusteroida, jotta saadaan yleiskuva siitä millaisia dokumentteja kokoelma sisältää ja päästään nopeammin käsiksi käyttäjää kiinnostaviin aineistoihin.

Muokkaus sisältää muun muassa erilaisten dokumenttirakenteiden yhdenmukaistamisen, koska on pystyttävä käsittelemään eri lähteistä tulevia dokumentteja, joiden dokumenttityypeillä ei välttämättä ole mitään tekemistä keskenään. Rakenteet luokitellaan älykkään algoritmin avulla, joka päättelee dokumenttien ja dokumenttityypin määrittelyjen (DTD) perusteella, minkäluonteisesti mikin SGML-elementti käyttäytyy, ja sijoittaa sen tämän perusteella yhteen geneeriseen eli yleiseen luokkaan. Esimerkiksi jokin elementti voi vaikuttaa otsikolta, koska sen tekstisisältö on lyhyehkö ja se sijaitsee dokumentissa tietyissä paikoissa. Muita yleisiä luokkia ovat muun muassa luvut, aliluvut, kappaleet ja ristiviittaukset.

Koska valitut alkiot voivat olla hyvin erikokoisia, saatetaan tarvita erilaisia täydennysoperaatioita, jotta lopullinen dokumentti olisi validi SGML-dokumentti ja myös sisällöltään mahdollisimman eheä. Esimerkiksi, jos uuteen dokumenttiin tulisi peräkkäin luku ja kappale, joka kuuluu eri lukuun, saatetaan lisätä kappaleeseen myös sen oman luvun otsikko ja vähän johdattelevaa tekstiä. Prototyyppimme pystyy tuottamaan koostamisen lopputuloksena HTML-dokumentteja sekä yleisen meta-DTD:n mukaisia SGML-dokumentteja. Edellä elementtien luokittelun yhteydessä mainitut yleiset luokat ovat meta-DTD:n elementtejä.

3 Dokumenttimateriaalin esikäsittely

Ennen koostamisjärjestelmäämme lataamista dokumenteille on suoritettava muutamia esikäsittelyvaiheita. Näitä ovat elementtien luokittelu yleisiin luokkiin, dokumenttien katkelmointi ja ryvästäminen. Elementtien luokittelusta löytyy lisätietoja lähteestä [AHH+98b]; seuraavassa keskitymme katkelmointiin ja ryvästämiseen.

Katkelmointi

Jotta olemassa olevista lähteistä voidaan koostaa uusia dokumentteja, on lähteet analysoitava ja jaettava sellaisiin osiin, joita yhteenliittämällä voidaan muodostaa järkeviä koosteita. Esiin nousee kysymys siitä, mikä on sovelias alkion koko. Selvää on, että tekstistä ei voi irrottaa yksittäisiä sanoja tai virkkeitä hukkaamatta suurinta osaa merkityksestä. Tekstiyhteys määrää merkitystä.

Kokonaiset kappaleet voivat joskus olla soveliaita osia koostamista varten, mutta todennäköisempää on, että luvut tai aliluvut ovat kappaleita ymmärrettävämpiä kokonaisuuksia. Toisaalta luvut (tai aliluvut) saattavat olla hyvinkin pitkiä ja sisältää erillisiä asiakokonaisuuksia. Tällöin niitä olisi hyödyllistä jakaa lyhyemmiksi osiksi, mikrodokumenteiksi, jotka sisältävät yhden asiakokonaisuuden. Tähän perustuu dokumenttien katkelmointimme.

Katkelma (fragmentti) on peräkkäisten kappaleiden (paragraafien) muodostama jono. Katkelmointiprosessissa eli katkelmien rajakohtien valinnassa hyödynnetään tekstin leksikaalista koheesiota eli tapauksessamme sitä, että samat sanat (termit) toistuvat tekstissä. Menetelmässämme lasketaan kaikille kappaleiden rajakohdille arvo, joka kuvaa leksikaalista koheesiota kyseisessä kohdassa. Katkelmien rajakohdiksi valitaan tämän jälkeen soveliaat leksikaalisen koheesion paikalliset minimikohdat dynaamisena ohjelmointina tunnettua ohjelmointimenetelmää käyttäen. (Lisätietoja tästä aiheesta löytyy lähteestä [Hei98].) Näin pystymme (tarvittaessa) automaattisesti lisäämään rakenteisiin dokumentteihin yhden uuden rakennetason.

Ryvästäminen

Yhtenä osana SAW-koostamisjärjestelmää on dokumenttikatkelmien ryvästäminen eli klusterointi. Tällä tarkoitetaan katkelmajoukon jakamista osajoukkoihin eli *rypäisiin* siten, että kunkin rypään sisällä katkelmat ovat keskenään mahdollisimman samankaltaisia, mutta mahdollisimman erilaisia muiden rypäiden sisältämien katkelmien kanssa. Katkelmien samanlaisuutta mitataan niiden sisältämien yhteisten, perusmuotoon palautettujen sanojen määrällä niin kutsuttua vektoriavaruusmallia käyttäen. Sanoista on karsittu pois kaikkein yleisimmät sanat ja sanat, jotka eivät kuvaa tekstin sisältöä, kuten esimerkiksi partikkelit ja yleisimmät verbit. Jako rypäisiin voi olla joko litteä tai hierarkkinen, jolloin kukin ryväs jaetaan edelleen rypäisiin tiettyyn rajaan asti. Tyypillisesti sopiva rypäiden määrä yhdellä hierarkiatasolla on 5-12, käyttötarkoituksesta riippuen.

Rypäät ovat keino esittää suuria osia katkelmakokoelmasta kohtuullisenkokoisina selailtavina paloina, joiden sisällöt ovat suhteellisen yhtenäisiä. SAW:ssa sovelletaan Cuttingin ja kumppaneiden esittelemää rypäisiin perustuvaa vuorovaikutteista Scatter/Gather-selailumenetelmää (katso [CKP+92] ja [CKP93]). Menetelmä perustuu siihen, että käyttäjä valitsee sopiviksi katsomansa rypäät ja ryvästää niiden sisältämät dokumentit (tapauksessamme dokumenttikatkelmat) uudelleen. Käyttäjä voi toistaa valintaa ja ryvästämistä, kunnes halutunlaisia dokumentteja on jäljellä sopivan pieni määrä tarkempaa käsittelyä varten.

SAW-järjestelmässä käytetään sekä hierarkkista että litteää ryvästämistä. Dokumenttikatkelmakokoelman esikäsittelyvaiheessa sille muodostetaan ryväshierarkia. SAW:n käyttäjä voi selata dokumenttikokoelmaa ryväshierarkian avulla: käyttäjä voi laajentaa valitsemansa rypään, jolloin hän saa näkyviin sen alla olevat rypäät. Käyttöliittymässä kustakin rypäästä esitetään muutama rypään yleisin sana ja rypään kolmen keskeisimmän katkelman otsikot. Katkelman keskeisyyttä rypäässä mitataan sillä, miten samankaltainen se on sanasisällöltään koko rypään kanssa. Litteää dynaamista ryvästämistä SAW-järjestelmä käyttää silloin, kun käyttäjä haluaa ryvästää valitsemansa katkelmajoukon (joka voi sisältää myös rypäitä) ryhmitelläkseen katkelmat niiden sanasisältöjen samankaltaisuuden perusteella.

Ryvästämistä varten SAW-järjestelmään toteutettiin erillinen ryvästysohjelma. Ryvästysohjelma saa syötteenään kustakin ryvästettävästä katkelmasta sen tunnisteen ja sanasisältöinformaation (kukin sana frekvensseineen). Ohjelma tulostaa muodostamansa ryväshierarkian oman DTD:nsä mukaisessa SGML-muodossa. Ryvästystulokseen voi vaikuttaa useilla parametreilla, joilla voi muun muassa määrätä rypäiden koon ja ryvästykseen käytettävien yleisimpien sanojen määrän. Ohjelmassa sovelletaan Cuttingin ja kumppaneiden algoritmeja, ja se osoittautui riittävän tehokkaiksi suurillakin aineistoilla.

4 SAW-koostamisjärjestelmä

Tässä luvussa esittelemme kehittämämme dokumenttien koostamisjärjestelmän prototyypin *SAW* (SID Assembly Workbench) (katso myös [AHH+98a]). Järjestelmän tarkoitus on toimia koealustana kehittäessämme ja kokeillessamme edellä kuvattua dokumenttien koostamismallia.

Järjestelmän idea on lyhyesti seuraava: Käyttäjä kokoaa koostedokumentin iteratiivisesti hakemalla, valitsemalla ja järjestämällä katkelmia dokumenttikokoelmasta. Kun käyttäjä on löytänyt sopivan joukon dokumenttikatkelmia, hän tallettaa ne. Koska kaikilla koosteeseen valituilla katkelmilla ei välttämättä ole sama DTD, järjestelmään on liitetty mahdollisuus yhdistää katkelmat yhdeksi yleisen DTD:n mukaiseksi dokumentiksi jatkokäsittelyä varten.

Käyttöliittymä

SAW:n käyttöliittymä perustuu *näkymän* (view) käsitteeseen. Näkymä esittää joko koko dokumenttikokoelmaa, koostamisprosessin välitulosta tai valmista koostetta. Näkymä on järjestetty joukko dokumenttialkioita, joko dokumentin alkuperäisiä elementtejä tai katkelmoinnin tuloksena syntyneitä fragmentteja eli katkelmia, ryvästämisen tuloksena syntyneitä rypäitä tai aikaisemmin järjestelmään tallennettuja näkymiä.

Käyttöliittymän muodostaa tulosikkuna (esimerkki kuvassa 2), jossa esitetään itse näkymä ja operaatiot näkymän manipuloimiseen. Aloitusnäkymässä esitetään tyypillisesti koko dokumenttikokoelma. Operaation suorittaminen johtaa yleensä uuden näkymän syntymiseen. Operaatioita suorittamalla käyttäjä muokkaa näkymästä vähitellen valmista koostetta, joka lopulta sisältää käyttäjän haluamat dokumenttien osat. Myös välituloksena syntyneitä näkymiä voidaan tallettaa myöhempää käyttöä varten, kuten liitettäväksi myöhemmin osaksi toista näkymää.

Kuva 2: ***Käyttöliittymä ja esimerkkinäkymä***

Käyttöliittymä jakautuu kolmeen loogiseen osaan. Ensimmäisessä osassa on näkymä dokumenttikokoelmaan. Toinen osa koostuu joukosta operaatioita näkymän muokkaamista varten. Näitä operaatioita ovat esimerkiksi alkioiden valitseminen ja järjestäminen, ryvästäminen, avainsana- ja samankaltaisuushaku sekä mahdollisuus suorittaa TranSID-kyselyjä suoraan toteutustason dokumenttitietokantaan. Kolmannessa osassa tarjotaan mahdollisuus näkymän tallettamiseen joko koostamisjärjestelmän palvelimen sisäisesti tai SGML-tiedostoksi levyjärjestelmään.

Näkymä koostuu joukosta alkioita, jotka esitetään taulukkona siten, että kukin taulukon rivi edustaa yhtä alkiota. Jokaista

alkiota edeltää järjestysnumerokenttä, jonka avulla näkymän alkioiden järjestystä voidaan muuttaa. Valintapainikkeilla valitaan, mihin näkymän alkioihin näkymän manipulointioperaatioita kulloinkin sovelletaan. Kaikki alkiot voidaan valita *Select All* -painikkeella, kaikki valinnat voidaan poistaa *Unselect all* -painikkeella, ja kaikki järjestys- ja alkiovalinnat voidaan peruttaa *Reset view* -painikkeella.

Preserve & Order -painikkeella hylätään kaikki valitsemattomat alkiot näkymästä ja järjestetään valitut alkiot uudelleen järjestysnumeroiden mukaiseen järjestykseen. *Zoom in* -painike korvaa valitut alkiot niiden sisällöllä, esimerkiksi lukuelementin joukolla luvun alilukuja. *Zoom out* -painike suorittaa päinvastaisen operaation eli korvaa valitut alkiot alkion sisältävällä dokumenttirakenteen osalla, esimerkiksi kappaleen aliluvulla. *Complete*-painikkeella laajennetaan näkymää hakemalla kokoelmasta valittuja alkioita täydentäviä osia, esimerkiksi otsikoita ja johdantokatkelmia. *Cluster*-painikkeella käynnistetään valittujen alkioiden dynaaminen ryvästysprosessi.

Käyttäjä voi etsiä katkelmia dokumenttikokoelmasta myös avainsanahaulla (*Search*). Avainsanahakua voidaan soveltaa joko koko dokumenttikokoelmaan tai vain näkymän valittuihin alkioihin. Ryvästystä hyödyntäen voidaan etsiä myös valittujen alkioiden kanssa samankaltaisia katkelmia. Haku käynnistetään *Search similar* -painikkeella.

Kun käyttäjä on tyytyväinen muodostamaansa näkymään, hän voi tallettaa sen myöhempää käyttöä varten (*Save view*). Talletettuja näkymiä yhdistelemällä voidaan kooste muodostaa myös pienemmissä erissä.

Valmis kooste voidaan ottaa ulos koostamisjärjestelmästä edelleen muokattavaksi yleiseen meta-DTD-muotoon muunnettuna (*Export as SGML*). Meta-DTD mahdollistaa eri DTD:itten mukaisista alkioista muodostetun koosteen editoinnin SGML-editorilla ilman dokumentin rakenteen hukkumista.

Arkkitehtuuri ja toteutus

Käyttäjä käyttää järjestelmää WWW-selaimen kautta. Itse SAW on toteutettu WWW-palvelimena. WWW-palvelimesta on liittymä tekstitietokantapalvelimeen, jossa suurin osa SAW:n toiminnallisuudesta on toteutettu.

Näkymä on itse asiassa HTML-lomake, joka muodostaa käyttäjälle näkyvän osan SAW:sta. Lomake koostuu vakio-osasta, jossa sijaitsevat painikkeet ja vakiolinkit, sekä dynaamisesta osasta, jossa esitetään näkymän dokumenttialkiot. Lomakkeella näkyvät alkiot liitetään varsinaisiin dokumenttitietokannassa sijaitseviin alkioihin lomakkeen piilokentissä sijaitsevien solmutunnisteiden avulla. Koska näkymän tila on tällä tavalla koodattu lomakkeeseen, voidaan itse tekstitietokantapalvelin pitää pitkälti tilattomana. Näin ollen operaatioita suoritettaessa lähetetään palvelimelle suoritettavan operaation lisäksi myös suorituskohteena olevien alkioiden solmutunnisteet.

Dokumenttitietokanta on toteutettu muunnos- ja kyselykielellä TranSID [JKL97]. TranSID toimii SGML-tekstitietokantapalvelimena, johon on ladattu dokumenttikokoelman SGML-tiedostot. TranSID mallintaa SGML-dokumentit puina, joiden solmuilla on yksikäsittäiset solmutunnisteet. Näitä solmutunnisteita käytetään HTML-dokumentissa dokumenttialkioiden yksilöimiseen. TranSID:ssa on toteutettu erityinen binäärimuotoinen SGML-puiden esitystapa, joka mahdollistaa hyvinkin suurten dokumenttitietokantojen selaamisen SAW:ssa. Palvelimen toiminnallisuus on toteutettu TranSID-kielisillä ohjelmilla. TranSID-palvelin ylläpitää myös tilatietoa silloin kun sen upottaminen HTML-lomakkeen sisään ei ole mahdollista tai käytännöllistä.

5 Yhteenveto

Dokumenttien koostaminen on luonteeltaan interaktiivinen prosessi, jossa kirjoittaja tai koostaja käyttää erilaisia apuvälineitä löytääkseen tarkoitustaan vastaavia lähteitä ja muodostaakseen niistä ehjän kokonaisuuden. Esittelimme tässä paperissa dokumenttien koostamismallin ja kokeellisen dokumenttien koostamisjärjestelmän. Rakenteisten dokumenttien uusiokäyttö ja koostaminen sekä laajojen heterogeenisten dokumenttikokoelmien hallinta ovat haastavia ongelmia, joiden ratkaisua olemme toivoaksemme osaltamme edistäneet.

Kiitokset

SID-hankkeessa tehtyä työtä ovat tukeneet Tekes ja yhteistyökumppanit (Aamulehti, Edita, Opetushallitus, WSOY, Helsinki Media, Lingsoft ja MTV3) osana Elektroninen painoviestintä -ohjelmaa. Barbara Heikkisen väitöskirjatyötä on lisäksi rahoittanut Suomen Kulttuurirahasto ja Oskari Heinosen väitöskirjatyötä Helsingin yliopiston 350-vuotissäätiö.

Lähteet

[AHH+98a] Helena Ahonen, Barbara Heikkinen, Oskari Heinonen, Jani Jaakkola, Pekka Kilpeläinen, and Greger Lindén. Design and implementation of a document assembly workbench. Electronic Publishing, Artistic Imaging, and Digital Typography, Proceedings of the 7th International Conference on Electronic Publishing, EP '98, Saint Malo, France, March/April 1998. Number 1375 in Lecture Notes in Computer Science, Springer-Verlag.

[AHH+98b] Helena Ahonen, Barbara Heikkinen, Oskari Heinonen, Jani Jaakkola ja Mika Klemettinen. Heterogeenisten dokumenttirakenteiden hallinta SGML-elementtien automaattisen luokittelun avulla. SGML/XML Finland '98, Jyväskylä,

lokakuu 1998.

[CKP+92] Douglass R. Cutting, David R. Karger, Jan O. Pedersen, and John W. Tukey. Scatter/Gather: A cluster-based approach to browsing large document collections. Proceedings of the 15th Annual International ACM SIGIR Conference, Copenhagen, Denmark, June 1992.

[CKP93] Douglass R. Cutting, David R. Karger, and Jan O. Pedersen. Constant interaction-time Scatter/Gather browsing of very large document collections. Proceedings of the 16th Annual International ACM SIGIR Conference, Pittsburgh, Pennsylvania, USA, June 1993.

[Hei98] Oskari Heinonen. Optimal multi-paragraph text segmentation by dynamic programming. Proceedings of the COLING-ACL '98 Conference, Montréal, Québec, Canada, August 1998.

[JKL97] Jani Jaakkola, Pekka Kilpeläinen, and Greger Lindén. TranSID: An SGML tree transformation language. Proceedings of the Fifth Symposium on Programming Languages and Software Tools, Jyväskylä, Finland, June 1997. Technical Report C-1997-37, University of Helsinki, Department of Computer Science, Finland.