

Reija Korhonen

**RAKENTEISTEN
LIIKETOIMINTADOKUMENTTIEN
VISUALISOINTI**

XSL:n soveltaminen organisaatioiden välisiin XML-dokumentteihin

Tietojärjestelmätieteen
pro gradu -tutkielma
10.2.2003

Jyväskylän yliopisto
Tietojenkäsittelytieteiden laitos
Jyväskylä

TIIVISTELMÄ

Korhonen, Reija Johanna

Rakenteisten liiketoimintadokumenttien visualisointi. XSL:n soveltaminen organisaatioiden välisiin XML-dokumentteihin/Reija Korhonen

Jyväskylä: Jyväskylän yliopisto, 2003.

97 s.

Pro gradu -tutkielma

Tutkielmassa tarkastellaan tyylikielen avulla tapahtuvaa rakenteisten liiketoimintadokumenttien visuaalista esittämistä ihmiselle. Rakenteisella dokumentilla tarkoitetaan tässä tutkielmassa dokumenttia, jonka rakenneosat on merkitty XML-suosituksen mukaisesti. Liiketoimintadokumentilla tarkoitetaan yrityksen liiketoiminnan prosesseihin kiinteästi liittyvää dokumenttia. Tutkimuksen tavoitteena on selvittää, miten XSL-tyylikielellä kuvatut rakenteisten liiketoimintadokumenttien esitysmääritykset tulisi toteuttaa. Tarkoituksena on keskittyä erityisesti tyylisivujen suunnittelussa ja toteuttamisessa huomioitaviin seikkoihin. Keskeisenä suunnittelun lähtökohtana on ylläpitönäkökulman huomiointi.

Dokumenttien rakenteisuus sallii niiden visuaalisen esittämisen eri tavoin, mutta jokainen erilainen tyylikielellä kuvattu esitys edellyttää omaa tyylisivumääritystä. Esitysmuotojen variaatioiden kasvaessa voidaan päätyä vaikeasti ylläpidettäviin ratkaisuihin. Tutkimusaiheen käsittely perustuu tässä työssä sekä kirjallisuudesta että esimerkkitapauksesta tehtyihin havaintoihin. Tutkielman aluksi on kuvattu yleisiä dokumenttien hallinnan kehittämisessä huomioitavia seikkoja sekä dokumentin roolia liiketoimintaprosessissa. Tämän jälkeen on pohdittu rakenteisten dokumenttien esittämistä tyylikieliä käyttäen. Tutkimustyössä on lisäksi kartoitettu XML-dokumenttien esittämiseen soveltuvia tyylikieliä, joista on tarkemmin perehdytty XSL-kieleen.

Tutkielman keskeisenä tuloksena on tyylisivujen toteutusmalli, joka osoittaa, että ylläpidettävän tyylisivun toteutuksessa on huomioitava monia dokumenttien visualisointiympäristöön liittyviä seikkoja.

AVAINSANAT: liiketoimintadokumentti, visualisointi, tyylisivu, EDI, OVT, XML, XSL.

ABSTRACT

Korhonen, Reija Johanna

A visual approach to structured business information. Applying XSL for interorganisational XML documents/Reija Korhonen

Jyväskylä: University of Jyväskylä, 2003.

97 p.

Master's Thesis

This thesis examines transformation of structured business document information into a visual form for human purposes. In this thesis structured document is understood as a document marked with rules specified in XML specification. Business document is seen as a document deeply tied up with business processes. The aim of this study is to find out how to implement visualisation rules using Extensible Stylesheet Language (XSL). The main focus is on design and implementation of style sheets. The maintainability is a key fact when finding out design solutions.

The explicitly marked structure of a document allows various visualisations of one document, but each different visual appearance requires its own style sheet. There might be difficulties with maintenance when allowing large amount of different visual appearances of a single document. This study is based on observation from literature and an practical case example. At first common facts to be considered when developing electronic document management and the role of a document in the business processes are given. Second, visualisation of structured documents when using style language is discussed. Style languages to be applied for visualising XML documents are also considered. The deepest focus has been on XSL language.

The main outcome of this study is an implementation model for style sheets. The model indicates that many facts about the visualisation environment has to be taken into consideration when implementing maintainable style sheets.

KEYWORDS: business document, visualisation, style sheet, EDI, XML, XSL.

SISÄLLYSLUETTELO

1	JOHDANTO.....	5
2	DOKUMENTTIEN HALLINNAN KEHITTÄMINEN.....	9
2.1	Dokumentti ja sen elinkaari	9
2.2	Rakenteinen dokumentti	12
2.3	Rakenteistaminen	13
2.4	Timanttimalli	15
3	LIIKETOIMINTADOKUMENTTIEN TUOTTAMINEN JA KÄYTTÖ	18
3.1	Liiketoimintaprosessit ja resurssit	18
3.2	Liiketoimintadokumenttien käsittelyn osapuolet	20
3.3	Liiketoimintadokumenttien rakenteistaminen	23
4	RAKENTEISTEN DOKUMENTTIEN ESITTÄMINEN	26
4.1	Rakenteisten liiketoimintadokumenttien esittäminen ihmiselle	26
4.2	Dokumentin vastaanottajan kyky hallita visuaalisuutta	31
4.3	Muunnos- ja tyylikielien käyttö	34
4.4	Muunnos- ja tyylikielien etuja	37
4.5	Muunnos- ja tyylikielien haittoja	38
4.6	Muunnos- ja tyylikieliä	40
5	XSL	44
5.1	XSL-tyylikieli	44
5.2	Esitysprosessi	46
5.3	Puumuunnos	49
5.4	Muotoilu ja esittäminen	51
5.5	XSL-muotoilukielen ominaisuuksia	54
5.6	XSL-muotoilukielen käyttö	57
6	TYYLISIVUJEN TOTEUTUSMALLI	60
6.1	Visualisointiympäristön kartoitus	60
6.2	Tyylisivujen muutostekijöiden arviointi	64
6.3	Tyylisivujen vaatimusmäärittely	67
6.4	Esimerkkitoteutuksen arviointi	74
7	YHTEENVETO	76
	LÄHDELUETTELO	79
	TUTKIELMASSA KÄYTETYT LYHENTEET	88
	LIITE 1: LASKUN XML-MUOTOINEN MALLIDOKUMENTTI	89
	LIITE 2: XSL-TYYLISIVUESIMERKKI	96

1 JOHDANTO

Liiketoimintadokumentit ovat yritysten operatiivisen toiminnan tärkeitä välikappaleita, joita ilman liiketoiminnan toteuttaminen organisaatioiden välillä tai niiden sisällä saattaa olla muun muassa organisaation toiminnan tai kirjanpitolainsäädännön näkökulmasta mahdotonta. Esimerkiksi tuotteita tai palveluita tarjoava yritys tekee tarjouksia, vastaanottaa ostajalta tilauksia sekä toimittaa ostettujen tuotteiden tai palveluiden perusteella ostajalle lähetteitä ja laskuja. Kaikki nämä liiketoimintaprosessit sisältävät joko dokumenttien tuottamista, vastaanottamista tai käsittelyä. Liiketoimintadokumentit tallettavat todisteita liiketoimintatapahtumista ja yhdistämällä useampi liiketoimintadokumentti saadaan aikaan kuvaus liiketoimintaprosessin etenemisestä (Wiggins 2000, 152).

Liiketoimintadokumentit on pitkään välitetty dokumentin tuottajalta vastaanottajalle paperimuodossa postin välityksellä. Vain suuret yritykset ovat pystyneet investoimaan täysin digitaalisiin liiketoimintadokumentteihin käyttämällä organisaatioiden välisen tiedonsiirron (myöhemmin OVT, englanniksi Electronic Document Interchange eli EDI) menettelyjä dokumenttien välittämisessä (Glushko, Tenenbaum & Meltzer 1999, 107; Weiss 1999, 41). Internetin käytön yleistymisen elektronisen liiketoiminnan toimintakanavana sekä W3-järjestön XML-suosituksen (Extensible Markup Language; Bray, Paoli, Sperberg-McQueen & Maler 2000) tarjoama avoin dokumenttien rakenteen kuvaustapa ovat tuoneet digitaaliset liiketoimintadokumentit myös pienempien yritysten ulottuville (Glushko ym. 1999, 107; Banerjee & Kumar 2002, 102; Diaz, Wüstner & Buxmann 2002, 1047).

Siirtyminen paperisten liiketoimintadokumenttien käytöstä digitaalisessa muodossa olevien dokumenttien käyttöön antaa uusia mahdollisuuksia toimintansa kehittämiseen etenkin liiketoimintadokumenttien vastaanottajille. Organisaatiot voivat esimerkiksi tehostaa dokumenttien välitystä, käsittelyä ja kierrätystä ottamalla käyttöön tähän tarkoitukseen kehitettyjä uusia sovelluksia ja palveluita (Banerjee & Kumar 2002). Koska liiketoimintadokumenttien käsittelyä ei voida kuitenkaan täysin automatisoida, on liiketoimintaprosessien sujuvuuden kannalta edelleen tärkeää, miten dokumentit esitetään visuaalisesti ihmiselle.

Dokumentit ovat tiedon siirron ja varastoinnin välineitä dokumentin tuottajan ja vastaanottajan välillä. Dokumentti on looginen kokonaisuus tallennettua tietoa, joka on merkityksellinen tietyssä sosio-organisaationaalisessa kontekstissa (Päivärinta & Peltola 2001, 1). Oleellista dokumentille on sen tallentaminen siten, että dokumentti on tarvittaessa katseltavissa niin usein kuin halutaan. Rakenteiseen dokumenttiin tallennetaan sekä dokumenttien käyttötilanteessa ihmiselle merkityksellinen tietosisältö että jollakin sovitulla merkintätavalla dokumentin loogiset osat, joista dokumentti koostuu (Heeman 1992, 143-144). Yleensä yksittäiset rakenteiset dokumentit sisältävät vain liiketoiminta-alueelle ominaista tietosisältöä sekä jollakin merkintätavalla ilmaistun dokumenttirakenteen, eivät sitä, miten dokumentti tulisi käyttäjälleen esittää.

Dokumenttirakenteen merkintäkielten, kuten SGML (Standard Generalized Markup Language; ISO 8879 1986) ja XML, yleisenä kehitystavoitteena on ollut dokumentin tietosisällön ja sen ulkoisen – visuaalisen tai auralisen – esitysmuodon erottaminen toisistaan. Rakenteisten dokumenttien ulkoisen esitysmuodon kuvaamiseen on kehitetty omia tyylikieliä (Hirvonen 1997). W3-järjestön kehittämä XSL-suositus (Extensible Stylesheet Language; Adler, Berglund, Caruso ym. 2001) on yksi XML-dokumenttien ulkoisen esitysmuodon määrittämiseen tarkoitettuista tyylikielistä.

Dokumenttien tuottamiseen, tallentamiseen ja katselemiseen tarvitaan aina jonkinlaista teknologiaa. Tänä päivänä liiketoimintadokumenttien tuottaminen, käsittely ja varastointi liittyvät kiinteästi liiketoimintatapahtumia suorittaviin ja tallentaviin ohjelmistoihin (Wiggins 2000, 152). Koska XML-suosituksen soveltaminen on varsin nuori alue liiketoimintadokumenttien tuottamiseen ja käsittelyyn tarkoitetuissa sovelluksissa, on myös niiden kyky visualisoida XML-muodossa olevia liiketoimintadokumentteja varsin rajoittunut. Tämän vuoksi XML-suosituksen mukaisesti rakenteistetuista liiketoimintadokumenteista onkin tuotettava eri järjestelmätoimittajien sovelluksiin eri muodossa olevia liiketoimintadokumenttien visuaalisia esityksiä. Vain siten liiketoimintaprosessit voivat sujua ongelmitta.

Liiketoimintadokumenttien rakenteisuus sallii niiden visuaalisen esittämisen eri tavoin esimerkiksi dokumenttien kierrättämiseen, käsittelyyn ja arkistointiin tarkoitettujen sovellusten asettamissa rajoissa. Teknisten rajoitteiden li-

säksi visuaalisen esittämisen eroavaisuus on edelleen myös organisaatiolähtöistä. Etenkin suuret organisaatiot haluavat liiketoimintadokumenttiansa näyttävän visuaalisesti erilaisilta verrattuna toisten organisaatioiden dokumentteihin. Vaikka erilaisten visuaalisten esitysten tuottaminen on rakenteille dokumenteille periaatteessa helppoa, edellyttää jokainen erilainen esitysmuoto omaa tyylikielellä toteutettua tyylisivumäärittystä. Erilaisten esitysmuotovariaatioiden kasvaessa voidaan päätyä vaikeasti ylläpidettäviin ratkaisuihin etenkin ympäristöissä, joissa on pystyttävä tukemaan useiden eri organisaatioiden liiketoimintadokumenttien erilaista visualisointia.

Tämän tutkimuksen tavoitteena on selvittää, miten XSL-tyyliekielellä kuvatut, rakenteisten liiketoimintadokumenttien visualisointiin tarkoitetut tyylisivut tulisi toteuttaa. Toteutustavan kartoituksessa painotetaan ylläpito näkökulmaa. Tutkimus rajoittuu XML-suosituksen mukaisesti rakenteistettuihin liiketoimintadokumentteihin. Liiketoimintadokumenttien lisäksi organisaatioiden toimintaan voi liittyä muitakin rakenteisessa muodossa olevia dokumentteja, kuten esimerkiksi tuotedokumentaatio tai käyttöohjeet. Toisaalta kaikki liiketoimintadokumentit eivät ole rakenteisessa muodossa, vaan niitä tuotetaan esimerkiksi tekstinkäsittelyohjelmalla. Tällaisten dokumenttien visualisoinnin tarkastelu rajataan tämän tutkimuksen ulkopuolelle.

Tutkimus perustuu kirjallisuuteen sekä esimerkkitapauksesta tehtäviin havaintoihin. Esimerkkitapauksena on usean eri organisaation liiketoimintadokumenttien välittämiseen tarkoitettu järjestelmä. Eri tyyppisistä liiketoimintadokumenteista tutkimus rajautuu laskuun. Aiheen käsittelyn havainnollistamiseksi tutkimuksessa käytetään monin paikoin esimerkkinä XML-muodossa olevaa yksittäistä laskua, joka on tutkimuksen liitteenä 1.

Tutkielman seuraavassa luvussa selvitetään tutkimuksen kannalta keskeisiä käsitteitä sekä dokumenttien käsittelyn kehittämisessä huomioitavia tekijöitä. Luvussa esitellään timanttimalli, joka auttaa hahmottamaan tutkimusongelmaan liittyvää kokonaisuutta. Tämän jälkeen luvussa kolme kuvataan rakenteisten liiketoimintadokumenttien tuottamis- ja käsittely-ympäristöä sekä sen teknistä kehittymistä organisaation toiminnan näkökulmasta. Luvussa 4 tarkastellaan rakenteisten dokumenttien visualisointia ihmiskäyttäjän näkökulmasta sekä selvitetään tyylikielien avulla tapahtuvan visualisoinnin periaatteita, hyötyjä ja haittoja. Lisäksi luvussa luodaan lyhyt katsaus XML-doku-

menttien visualisointiin soveltuviin muunnos- ja tyylikieliin. Luvussa 5 syvennyttään XSL-tyylikieleen. Luvun alussa kerrotaan kielen soveltamiseen liittyvästä muunnos- ja esitysprosessista sekä prosesseihin tarvittavista teknisistä komponenteista. Lisäksi luodaan katsaus XSL-kielen muotoiluosan ominaisuuksiin sekä kielen soveltamiseen liittyviin tyylisivun toteutustehtäviin. Luvussa 6 esitellään tutkimuksessa tuotettu tyylisivujen toteutusmalli sekä arvioidaan sitä. Tutkielman päättää luvun 7 lyhyt yhteenveto tutkimuksessa tarkastelluista asioista.

Tutkimuksen keskeisenä tuloksena on tyylisivujen toteutusmalli. Toteutusmalli osoittaa, että ylläpidettävien tyylisivujen toteuttamisessa on huomioitava yksittäisten tyylisivujen lisäksi myös muut visualisointiympäristön ominaisuudet. Yksittäinen ylläpidettävä tyylisivu muodostuu selkeästä, ositetusta rakenteesta. Tutkimuksessa tuotettuja tuloksia voidaan soveltaa sellaiseen liiketoimintadokumenttien käsittelyyn liittyvien järjestelmien ja komponenttien kehityksessä sekä yrityksen liiketoimintadokumenttien rakenteistamisprosessissa. Vaikka tutkimuksessa tarkastellaan erityisesti liiketoimintadokumentteja ja niistäkin esimerkkitapauksena lasku-tyyppistä dokumenttia, on tutkimuksessa tuotettu tyylisivujen toteutusmalli sovellettavissa myös muun tyyppisten rakenteisten dokumenttien visualisointiin tarkoitettujen tyylisivujen toteuttamiseen.

2 DOKUMENTTIEN HALLINNAN KEHITTÄMINEN

Tässä luvussa selvitetään tutkimuksen kannalta keskeisiä käsitteitä sekä kuvataan yleisiä dokumenttien hallinnan kehittämiseen ja rakenteisten dokumenttien käyttöönottoon liittyviä keskeisiä tekijöitä.

2.1 Dokumentti ja sen elinkaari

Dokumentit ovat tiedon siirron ja varastoinnin välineitä dokumentin tuottajan ja vastaanottajan välillä. *Dokumentti* voidaan määritellä loogiseksi kokonaisuudeksi tallennettua tietoa, joka on merkityksellinen tietyssä sosio-organisaationaalisessa kontekstissa (Päivärinta & Peltola 2001, 1). Oleellista dokumentille on sen tallentaminen siten, että dokumentti on tarvittaessa katseltavissa niin usein kuin halutaan. Dokumenttien tuottamiseen, tallentamiseen ja katselemiseen tarvitaan aina jonkinlaista teknologiaa.

Dokumentit ovat organisaation liiketoimintatiedon varastointipaikkoja. Dokumenttikokoelmat muodostavat osan yrityksen toimintaan, tuotteisiin ja palveluihin liittyvästä tietovarannosta (Haimes 1994, 15). Tietovarastoon kuuluu esimerkiksi tuote- ja asiakastietoja sekä muita liiketoiminnan ohjaustietoja, jotka olisivat ilman dokumentteja täysin yrityksen henkilöstöstä riippuvaa tietoa. *Liiketoimintadokumentilla* tarkoitetaan yrityksen liiketoiminnan prosesseihin kiinteästi liittyvää dokumenttia. Liiketoimintadokumentit tallentavat todisteita liiketoimintatapahtumista ja yhdistämällä useampi liiketoimintadokumentti muodostuu tallenne liiketoimintaprosessin etenemisestä (Wiggins 2000, 152). Liiketoimintadokumenttien lisäksi yrityksen toimintaan liittyy myös muunlaisia dokumentteja, kuten esimerkiksi kokousmuistioita, joita tuotetaan muun muassa tekstinkäsittelyohjelmilla.

Tietyt liiketoimintadokumenttien piirteet erottavat ne muun tyyppisistä dokumenteista. Ensinnäkin liiketoimintadokumenttien sisältämä tieto on jo perusluonteeltaan rakenteista, ei lineaarista (Wiggins 2000, 153). Liiketoimintadokumentti koostuu pienistä tietoyksiköistä, kuten lähettäjän ja vastaanottajan tiedoista, tuotteiden nimistä, kappalemääristä ja hintatiedoista. Kokonaisuuden ymmärrettävyyden kannalta on toissijaista, onko esimerkiksi laskussa ensimmäisenä laskun päivä vai laskun numero. Toiseksi liiketoimintado-

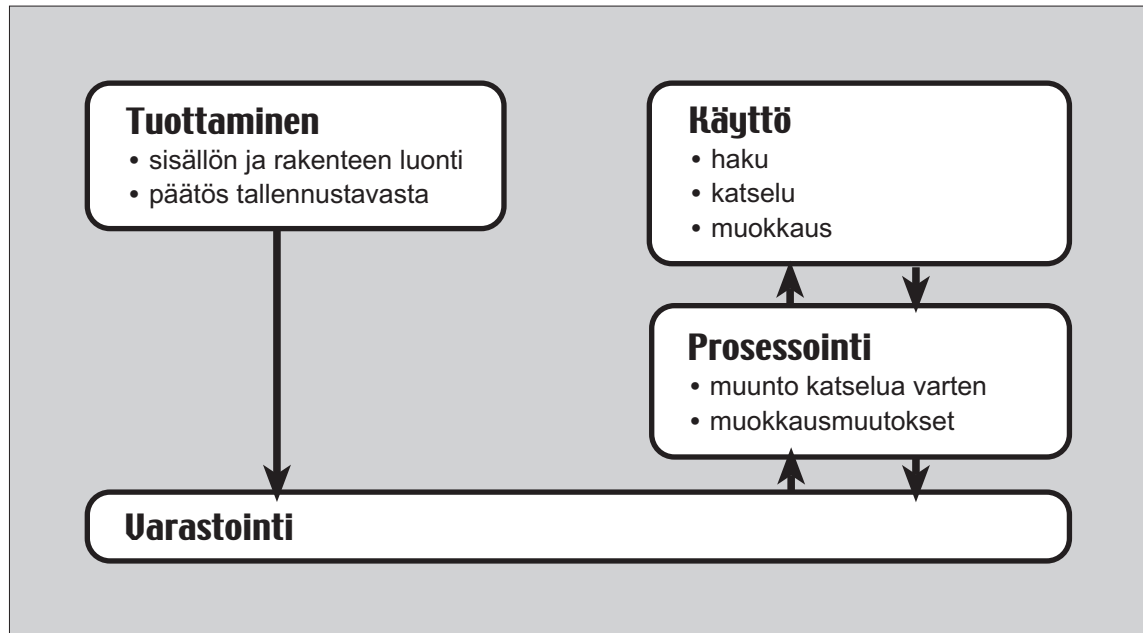
kumenttien tuottaminen ja käsittely liittyvät kiinteästi liiketoimintatapahtumia suorittaviin järjestelmiin (Wiggins 2000, 152).

Dokumentteja voidaan luokitella erilaisiin *dokumenttityyppeihin* dokumenttien sisällön ja tarkoituksen mukaan. Dokumenttien jakaminen tyyppeihin perustuu haluan luokitella ihmisten käyttöön tarkoitettua tietoa erilaisiin, helpommin hallittaviin kokonaisuuksiin. Nämä kokonaisuudet voidaan dokumenttien tapauksessa nimetä dokumenttityypeiksi. Jotta tietty dokumenttityyppi voidaan liittää dokumenttien luokituksen, on dokumenttityypin merkitys oltava samanlainen useammalle kuin yhdelle ihmiselle. (Päivärinta 2000, 28)

Tässä tutkimuksessa tarkasteltu esimerkkitapaus rajautuu lasku-tyyppiseen liiketoimintadokumenttiin. Dokumenttityypistä lasku vallitsee yhtenäinen käsitys useiden ihmisten kesken ja useissa eri organisaatioissa. Joissain tapauksissa lasku voidaan kuitenkin ymmärtää yläkäsitteeksi erilaisille laskutyypeille kuten esimerkiksi veloituslasku, koontilasku tai hyvityslasku. Se, määritetäänkö tällaisessa tapauksessa kukin laskutyyppi omaksi dokumenttityypiksi, riippuu organisaation sosiaalisesta järjestelmästä. Tässä tutkimuksessa lasku ymmärretään laajimmassa merkityksessään.

Kullakin dokumentilla on oma elinkaarensa. Dokumenttien elinkaari ja elinkaaren vaiheet aina dokumentin luomisesta sen tuhoamiseen asti on huomiotavaa kehitettäessä dokumenttien hallintaa (Salminen, Lyytikäinen & Tiitinen 2000, 634). Dokumenttien elinkaaren voidaan nähdä sisältävän esimerkiksi seuraavia vaiheita: luominen, tallennus, haku, katselu, muokkaus ja poisto. Elinkaaret voivat poiketa eri organisaatioissa ja eri dokumenttityypeillä hyvinkin paljon toisistaan.

Dokumentin elinkaaren vaiheet ennen sen hävittämistä voidaan ryhmitellä kuvan 1 tavoin neljään eri toimintoon: tuottamiseen, varastointiin, prosessointiin ja käyttöön (Päivärinta & Tyrväinen 2001, 7). Dokumenttien sisältämä tieto voidaan tuottaa usealla eri tavalla. Tiedon tuottaminen voi tapahtua esimerkiksi manuaalisesti kirjoittamalla tai automatisoidusti esimerkiksi elektronisen kaupankäynnin osaprosessina. Dokumentin tuottamisen lähtökohtana voi olla jo olemassa olevat dokumentit, esimerkiksi yrityksen tuotteiden



Kuva 1. Dokumentin elinkaaren toiminnot (Tyrväinen 2002, 6 mukailten)

yksittäisen myyntitapahtuman dokumentoiva lasku voidaan tuottaa tuotelu-
ettelo- ja tilausdokumenttien tietosisältöjä yhdistämällä ja täydentämällä.

Tuottamisessa dokumenttiin liitetään sisältö ja määritetään sen rakenne eli dokumenttiarkkitehtuuri sekä dokumenttiyksiköt eli osat, joista dokumentti koostuu. Tämän lisäksi tehdään päätös tallennustavasta ja -mediasta sekä sisäisestä esitysmuodosta, jossa dokumentti varastoidaan. Tuottamisen jälkeen dokumentti varastoidaan tallennusmedialle, esimerkiksi paperille, tietokoneen kovalevyille, levykkeelle tai CD-levylle.

Dokumentin käyttö edellyttää dokumentin hakua valitun tallennusmedian dokumenttivarastosta ja prosessointia käyttöä varten. Dokumentti on käytettävissä niin kauan, kun se on varastoituna jollekin tallennusmedialle. Digitaalisessa muodossa varastoitujen dokumenttien käytettävyys edellyttää tämän lisäksi, että dokumentin tallennusmuodon ymmärtäviä sovelluksia on saatavilla. Dokumentin elinkaaren viimeisenä käyttövaiheena on dokumentin poistaminen tallennusmedialta.

Dokumenttia käytettäessä siihen voidaan kohdistaa hakuja ja sitä voidaan katsella ja muokata. Näiden perustehtävien lisäksi eri tyyppisten dokumenttien käyttöön voi liittyä dokumenttityypille ominaisia tehtäviä. Esimerkiksi ostolaskujen käsittelyyn sisältyy hyväksyntä-tehtävä, jossa varmistetaan las-

kun aiheellisuus ja tietosisällön oikeellisuus ja siten hyväksytään lasku maksettavaksi. Toinen laskudokumentille ominainen käyttötehtävä on laskujen arkistointi. Arkistoinnissa dokumentit siirretään lyhytaikaisesta varastointipaikasta pitkäaikaisvarastoon kirjanpitolainsäädännössä säädellyksi ajaksi.

Dokumenttien käyttötarpeet asettavat vaatimuksia prosessoinnille ja tallennukselle, sillä dokumentin käyttö on mahdollista vain prosessoinnin ja tallennustavan asettamissa rajoissa. Esimerkiksi paperinen dokumentti on käytävissä periaatteessa vain paperisena. Nykyteknologia kuitenkin laajentaa dokumenttien käytettävyyttä, sillä monenlaiset dokumenttimuunnokset ovat mahdollisia. Esimerkkejä muunnoksista ovat digitaalisen dokumentin tulostaminen paperille tai paperisen dokumentin skannaaminen digitaalseksi. Digitaalisen dokumentin käyttäjänä voi olla ihminen tai tietokone. Digitaalisessa muodossa olevien dokumenttien käytölle aiheuttavat lisärajoitteita tuottamistavan ja tallennusmedian lisäksi erilaisten päätelaitteiden ja niiden näyttöjen ominaisuudet sekä tiedonsiirtokanavien kaistaleveydet.

2.2 Rakenteinen dokumentti

Rakenteisella dokumentilla tarkoitetaan dokumenttia, josta on havaittavissa dokumentin sisällön lisäksi osat, joista dokumentti koostuu. Jotta rakenne olisi havaittavissa, merkataan dokumentin osat jollain sovitulla merkintätavalla, esimerkiksi erikoismerkeillä tai kirjaimilla tai näiden yhdistelmillä. Dokumentin merkatuista osista muodostuu dokumentin *looginen rakenne*. Looginen rakenne voidaan esittää käsitteellisesti hierarkkisena puuna, jossa hierarkian juuri vastaa koko dokumenttia ja kertoo samalla, minkä tyyppisestä dokumentista on kyse. Puurakenne haarautuu solmukohdissa rakennehierarkian osia vastaavalla tavalla. Rakennehierarkiassa alimmalla tasolla eli lehtisolmuissa on varsinainen dokumentin tietosisältö. (Heeman 1992, 143-144)

XML (Extensible Markup Language; Bray ym. 2000) on yksi dokumenttirakenteen kuvauskielistä. W3-järjestön tuottama XML-suositus määrittää dokumentin rakenteen kuvaamiseen käytettävät merkit. XML on siten metakieli, jolla kuvataan muita kieliä. XML-kielen soveltajan tehtävänä on päättää siitä, mitä nimiä rakenteen osista käytetään. XML-syntaksia noudattavan kielen syntaksi eli rakennetta kuvaavat elementtien nimet ja niiden loogiset suhteet voidaan ilmaista erillisessä DTD-määrittelyssä (Document Type Definition)

tai jollain skeemakielellä toteutettavassa rakennemäärittämisessä. XML-dokumentin tuottaja merkitsee dokumentin rakenteen elementti- ja attribuuttitunnisteilla. Elementit ja attribuutit sisältävät dokumentin tuottajan ja vastaanottajan kannalta varsinaisen merkityksellisen tietosisällön.

XML-suosituksessa määritetään XML-dokumentin loogisen rakenteen lisäksi dokumenttien fyysinen rakenne. Suositus sallii loogisesti yhtenäiselle dokumentille fyysisen modulaarisuuden. Dokumentti voi siten koostua dokumentissa viitatuista *entiteeteistä*, jotka on tallennettu erilleen varsinaisesta dokumentista. Syntaksin ja rakenteen lisäksi XML-suositus kuvaa myös kaksikerroksisen jäsentimestä ja sovelluskerroksesta muodostuvan prosessointiarkkitehtuurin XML-dokumenttien käytölle. (Bray ym. 2000)

Prosessointiarkkitehtuurin ensimmäisen kerroksen muodostaa *jäsennin* (XML processor, XML parser), jonka ensisijainen tehtävä on varmistaa, että prosessoitava dokumentti on *hyvin muodostettua* eli XML-suosituksessa määritetyn syntaksin mukaista. Toinen XML-jäsentimen tehtävä on tarvittaessa *validoida* XML-dokumentti eli tarkastaa dokumentin *oikeellisuus*. Tällä varmistetaan, että XML-dokumentin rakenne noudattaa rakennemäärittämisessä kuvattua loogista rakennetta. Tarkastettuaan XML-dokumentin hyvin muodostuneisuuden – ja tarvittaessa oikeellisuuden – ja todettuaan sen näiltä osin virheettömäksi, jäsennin välittää arkkitehtuurin sovelluskerrokselle XML-dokumentin sisällön ja rakenteen. (Birbeck, Duckett, Gudmundsson ym. 2001, 17) Toteutustasolla jäsennin on ohjelmistokomponentti, joka välittää XML-dokumentin sovelluksen käyttöön joko hierarkkisena oliomallina (DOM; Apparao, Byrne, Champion ym. 1998 ja Le Hors, Le Hégaret, Wood ym. 2000) tai sarjana peräkkäisiä tapahtumia (SAX 2002).

2.3 Rakenteistaminen

Rakenteistamisella tarkoitetaan siirtymistä rakenteisten dokumenttien käyttöön (Lyytikäinen, Päivärinta, Salminen & Tiitinen 1997). Rakenteistamiseen kuuluu paljon muitakin tehtäviä kuin pelkästään dokumenttien sisältämän tiedon kuvaavien merkkauk- ja ositussääntöjen sopiminen (Salminen, Lyytikäinen, Tiitinen & Mustajärvi 2001). Rakenteistamisprosessi voi sisältää yhden tai useamman organisaatiossa käytössä olevan dokumenttityypin rakenteistamisen. Rakenteistamisen on todettu olevan pitkäkestoinen prosessi eri-

tyisesti silloin, kun on kyse useita organisaatioita kattavasta ja useita dokumenttityyppejä sisältävästä rakenteisten dokumenttien käyttöönottoon liittyvästä kehitysprosessista (Salminen ym. 2000, 639).

Rakenteistamisprosessi voidaan vaiheistaa järjestelmäkehityksen tavoin määrittelyyn, suunnitteluun ja toteutukseen. Määrittelyyn kuuluu dokumenttien, niiden käytön ja käyttäjien, käytettävien järjestelmien, toimintatapojen sekä käyttäjien tarpeiden analyysi. Suunnittelu jakaantuu edelleen kolmeen erilliseen osavaiheeseen. Ensinnäkin arvioidaan ja uudelleensuunnitellaan organisaatiossa tarvittavat dokumenttityypit. Toiseksi suunnitellaan uudelleen työkäytännöt sekä kolmanneksi valitaan uutta toimintatapaa tukevat järjestelmät ja suunnitellaan mahdollisesti tarvittava järjestelmien mukautus. (Salminen ym. 2001) Mikäli on kyse useiden organisaatioiden välisten dokumenttien rakenteistamisesta, tulee kunkin organisaation välillä sopia eri dokumenttityypit määrittävien yhteisten rakennemääritysten käytöstä sekä dokumenttien välitystavasta. Erilaisia sellaisenaan käyttöönotettavia XML-suositusta soveltavia dokumenttityyppejä on kehitetty sekä kansainvälisesti että kansallisesti: esimerkiksi lasku-tyyppiselle dokumentille on saatavilla julkisia DTD-määrittäjiä, joita ovat kehittäneet joko yksittäiset organisaatiot tai standardointiyhteisöt.

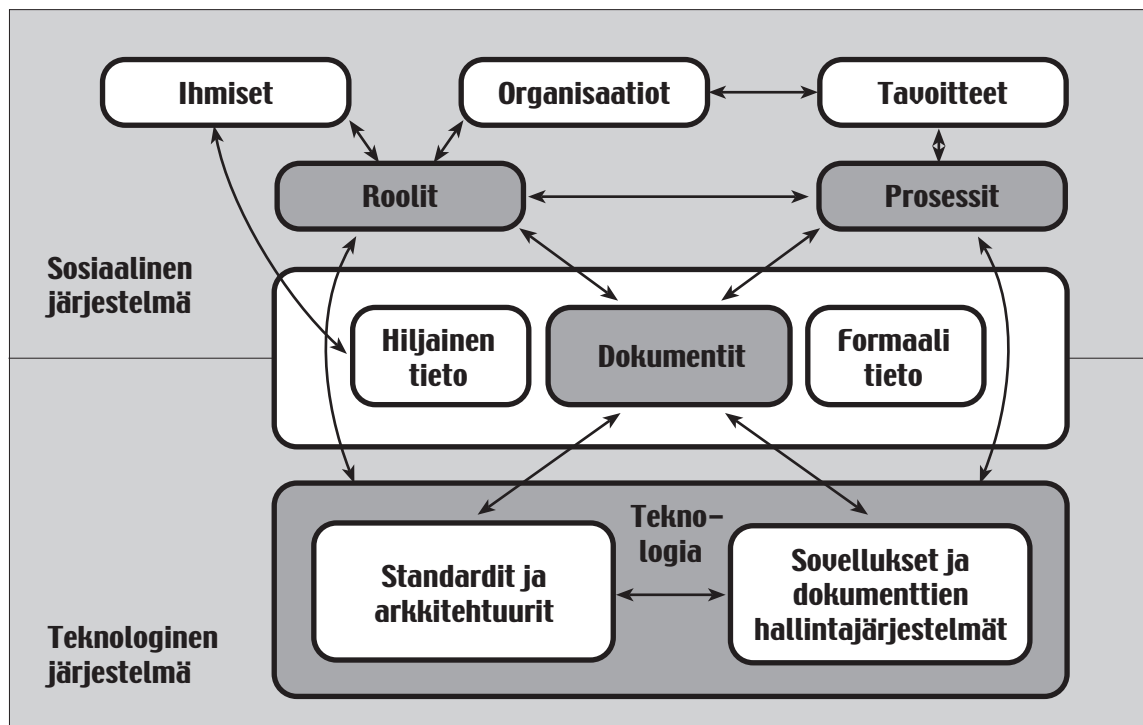
Kun kaikki suunnitteluvaiheeseen liittyvät tehtävät on saatu valmiiksi seuraa toteutusvaihe. Rakenteistamisprosessi päättyy prosessin arviointiin sekä niiden ihmisten koulutukseen, joiden työkäytännöt ovat muuttuneet dokumenttien rakenteistamisen seurauksena. (Salminen ym. 2001)

Rakenteisten dokumenttien visualisoinnin suunnittelu ja toteutus ovat oleellinen osa rakenteisten dokumenttien elinkaaren vaiheiden eri toimintoja tukevien järjestelmien kehitystä. Visualisoinnin suunnittelu saattaa sisältyä myös rakenteistamisprosessissa tehtävään järjestelmän mukauttamiseen esimerkiksi silloin, kun kohdeorganisaatiossa halutaan visualisoida rakenteiset dokumentit organisaation erityistoiveiden mukaisesti. Erityinen tarve visualisoinnin suunnittelulle rakenteistamisprosessissa on silloin, kun rakenteisten dokumenttien tuottamiseen tai käsittelyyn valittu järjestelmä ei pysty suoraan tukemaan esimerkiksi tyylikielien avulla tapahtuvaa rakenteisessa muodossa olevien dokumenttien visualisointia.

2.4 Timanttimalli

Rakenteisten dokumenttien visualisoinnin kehittäminen on osa dokumentti- muodossa olevan tiedon hallinnan kehittämistä. Kokonaiskuva käsillä ole- vasta ongelmakentästä voidaan muodostaa timanttimallin (Salminen 2000, 301; Päivärinta 2000, 18) avulla. Timanttimalli eriyttää neljä dokumenttien hallinnan kehittämisessä huomioitavaa osa-alueetta, jotka erillisyydestään huolimatta ovat vahvasti sidoksissa toisiinsa. Nämä neljä tekijää ovat ensin- näkin organisationaaliset prosessit, toiseksi organisatorisissa rooleissa toimi- vat ihmiset, kolmanneksi organisaatiossa käytössä oleva teknologia ja sen tar- joamat mahdollisuudet sekä neljänneksi itse dokumentit. Dokumenttien hal- lintaa voidaan lähestyä timantin kunkin ulottuvuuden näkökulmasta. On kuitenkin huomioitava, että millä tahansa alueella toteutettu kehitysmuutos säteilee aina vaikutuksia myös kolmelle muulle osa-alueelle.

Mallin ytimen muodostamaa neljää osatekijää voidaan selventää lisäämällä malliin tarkennuksia (Tyrväinen 2002, 16). Kuvassa 2 on esitetty tarkennettu timanttimalli, jossa mallin neljä ydinkomponenttia - roolit, prosessit, doku- mentit ja teknologia - erottuvat tumman harmaalla pohjavärillä valkopohjai- sista, tarkennettuun malliin tehdyistä lisäyksistä.



Kuva 2. Tarkennettu timanttimalli (Tyrväinen 2002, 16 mukailten)

Tarkennus havainnollistaa, miten roolit muodostuvat ihmisten ja organisaatioiden tehtävistä ja tarpeista. Samoin havainnollistuu prosessien suhde organisaation tavoitteisiin. Tarkennetussa mallissa dokumenttien rinnalle on kuvattu myös muuta organisaatioon varastoitunutta tietoa. Muu tieto on ihmisiin ja kirjaamattomiin toimintatapoihin varastoitunutta hiljaista tietoa. Tämän lisäksi tietoa on varastoitunut formaalissa muodossa organisaatiossa käytössä oleviin tietokantoihin. Hiljainen tieto, dokumentit ja formaali tietokantatieto muodostavat yhdessä organisaation tietovarannon. Teknologia on tarkennuksessa jaoteltu kahteen erilliseen kehittämisen kohteeseen: ensinnäkin standardeihin ja arkkitehtuureihin sekä toiseksi teknisiin sovelluksiin ja dokumenttien hallintajärjestelmiin.

Timanttimalli jakaa dokumenttimuodossa olevan tiedon hallinnan ja sen kehittämisen kahteen erilliseen osajärjestelmään: sosiaaliseen ja teknologiseen. Dokumentit liittyvät sekä sosiaaliseen että teknologiseen järjestelmään ja tämän vuoksi liiketoimintaan liittyvien dokumenttien hallinnan kehittämisessä onkin otettava huomioon muutostarpeet ja muutosten vaikutukset molemmissa osajärjestelmissä. Timanttimalli määrittää sosiaaliseen järjestelmään kuuluvaksi liiketoimintaprosessit sekä liiketoimintaprosesseihin liittyvissä rooleissa toimivat ihmiset. Teknologinen järjestelmä muodostuu käytössä olevasta teknisestä infrastruktuurista kuten laitteista, sovellusohjelmista, tietojärjestelmistä ja tietoliikenneverkosta sekä teknisistä dokumenttistandardeista ja dokumenttiarkkitehtuureista.

Sosiaalisen järjestelmän näkökulmasta liiketoimintadokumenttien hallintaa voidaan kehittää prosessi- ja roolilähtöisesti. Prosessilähtöisessä kehittämisessä keskitytään ensisijaisesti liiketoimintaprosessien mallintamiseen. Saatujen mallien perusteella asetetaan vaatimukset dokumenteille, käytettävälle teknologialle sekä rooleissa toimiville ihmisille. Roolilähtöinen kehittäminen on vahvasti sidoksissa prosessilähtöiseen kehittämiseen. Siinä keskitytään kartoittamaan niitä dokumentteihin liittyviä tietotarpeita, joita liiketoimintaprosessien toteuttamiseen liittyvissä rooleissa toimivat ihmiset asettavat. Tällöin keskeistä on eri rooleissa käsiteltävien dokumenttien sisältämän tiedon käytettävyys. (Tyrväinen 2002, 9)

Teknologisesta näkökulmasta kehittämisen kohteina ovat tekniset dokumentti-arkkitehtuurit ja -standardit sekä käytettävät sovellusohjelmat ja doku-

menttien hallintajärjestelmät. Teknologiaan liittyvät myös dokumenttien välittämiseen liittyvät tiedonsiirtokanavat, käyttöoikeudet ja tietoturvakysymykset. (Tyrväinen 2002, 9) Teknologia tarjoaa uusia innovaatioita ja teknologiapainotteisuus mahdollistaakin kaikkein tehokkaimman tavan kehittää liiketoimintadokumenttien käsittelyä. Investointipäätöksissä on kuitenkin kustannus-hyöty-suhteen lisäksi arvioitava investointien vaikutukset dokumenttien käsittelyyn liittyvään sosiaaliseen järjestelmään paitsi organisaation sisällä myös eri sidosryhmien ja organisaatioiden välillä.

Organisaatiot ovat hyvin eri tasoilla teknologian hyödyntämisessä liiketoimintadokumenttien käsittelyssä. Myös teknisesti samalla kehitystasolla olevissa organisaatioissa voi olla käytössä hyvin erilaista – esimerkiksi eri standardeja tukevaa tai toimittajakohtaisiin ratkaisuihin perustuvaa – dokumenttien käsittelyyn tarkoitettua teknologiaa.

Organisaatioiden välillä niin kuin myös organisaatioiden sisällä on huomioitava eroavaisuudet myös sosiaalisissa järjestelmissä. Kunkin organisaatioyksikön sosiaalinen järjestelmä vaikuttaa dokumentteihin sekä niiden käsittelyyn liittyvään teknologiaan. Vaikutus on myös päinvastainen eli käytössä oleva teknologia muokkaa sosiaalista järjestelmää. Organisaatioiden välillä integroitava tieto voi siten olla semanttisesti hyvinkin heterogeenista ja käytössä voi olla monenlaisia tiedon jäsentämistapoja (Stonebraker & Hellerstein 2001, 554). Tämän vuoksi organisaatioiden välisen liiketoimintadokumenttien käsittelyn yhteen sovittaminen ei ole triviaali tehtävä. Kuten organisaatioiden sisäistä dokumenttien hallintaa kehitettäessä, on myös organisaatioiden välisen dokumenttien käsittelyn kehittämisessä huomioitava kaikki timanttimallissa kuvatut osa-alueet.

Timanttimalli ei sellaisenaan anna vastauksia liiketoimintadokumenttien käsittelyyn tarkoitettujen järjestelmien ominaisuuksien tai komponenttien kehittämiseen. Se auttaa kuitenkin havainnollistamaan esimerkiksi yksittäiseen arkkitehtuuriin tai standardiin perustuvien komponenttien kehittämisessä huomioitavia seikkoja. Seuraavassa luvussa tarkastellaan rakenteisten liiketoimintadokumenttien visualisointia sosiaalisen järjestelmän eli prosessien ja roolien näkökulmasta. Tämän jälkeen tutkielman loppuosassa tutkimusaiheen tarkastelunäkökulma on teknologinen, mutta samalla sosiaalisen järjestelmän huomioiva.

3 LIIKETOIMINTADOKUMENTTIEN TUOTTAMINEN JA KÄYTTÖ

Tässä luvussa luodaan katsaus rakenteisten liiketoimintadokumenttien hallinnan kokonaisuuteen, jotta tutkimuksen kohdeympäristöstä voidaan muodostaa selkeä kokonaiskuva. Tarkastelunäkökulma on sosiaalinen. Aluksi selvitetään, mitä ovat liiketoimintaprosessit ja resurssit sekä miten dokumentti liittyy niihin. Tämän jälkeen kartoitetaan liiketoimintaprosessiin liittyvien dokumenttien käsittelyn osapuolia ja niiden välisiä suhteita sekä liiketoimintadokumenttien tuottamiseen, välittämiseen ja vastaanottamiseen liittyvää teknologiaa. Luvun lopuksi tarkastellaan rakenteisten liiketoimintadokumenttien tuottamiseen liittyvän teknologian muutoksia selvittämällä, mitä liiketoimintadokumenttien rakenteistaminen tarkoittaa ja mitä liiketoimintadokumenttien rakenteisuus merkitsee niiden visuaaliselle esittämiselle.

3.1 Liiketoimintaprosessit ja resurssit

Liiketoimintaprosessi on joukko tietyssä järjestyksessä suoritettavia tehtäviä, jotka tähtäävät yrityksen liiketoimintatavoitteiden saavuttamiseen. Tavoitteena on usein tarjota asiakkaalle tuote tai palvelu halutulla tehokkuudella, jonka mittareina voidaan käyttää esimerkiksi kuluja, pitkäikäisyyttä, palvelun tasoa ja laatua. Liiketoimintaprosessi tapahtuu organisaatiorakenteessa, joka muodostuu organisaation yksiköistä, ihmisistä ja rooleista sekä osaamisesta. (Tsalgatidou 1999, 60)

Liiketoimintaan liittyvä tieto on varastoituneena *liiketoiminnan resursseihin*, jotka muodostuvat dokumenteista, järjestelmistä ja toimijoista. Toimijat ovat ihmisiä ja organisaatioita, jotka käyttävät dokumentteja ja järjestelmiä liiketoiminnan prosesseja suorittaessaan. (Salminen ym. 2000, 625) Ihmiset toimivat työssään aina jossakin roolissa, joka liittyy suoritettavaan liiketoimintaprosessiin. Esimerkiksi myyntiprosessiin sisältyviä työrooleja ovat myyjä ja ostaja.

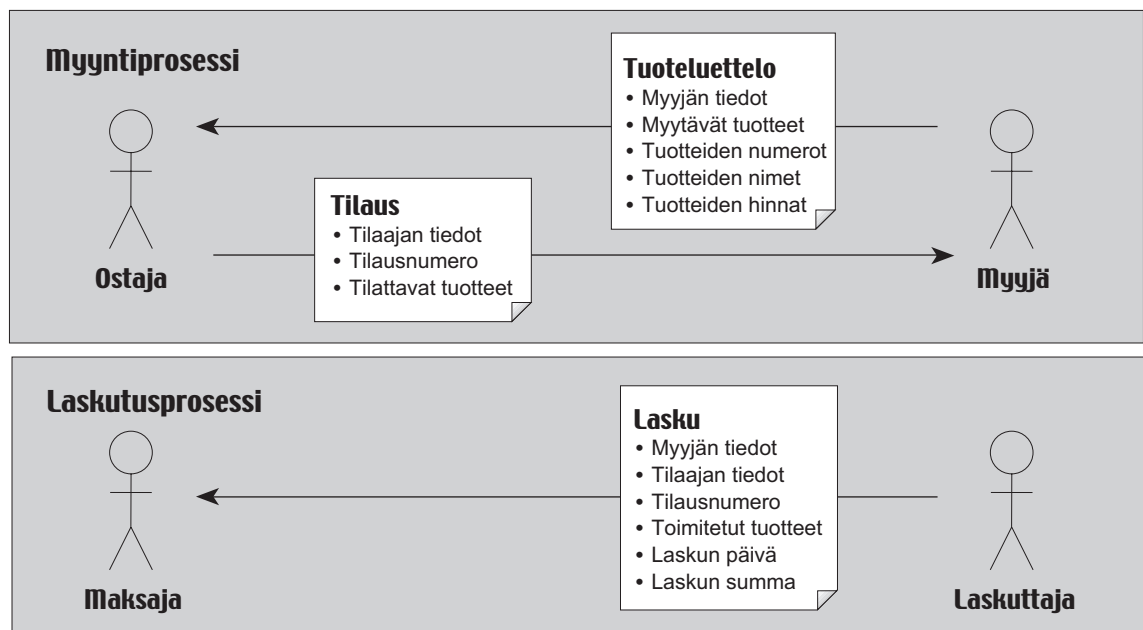
Liiketoimintaprosesseihin liittyy erilaisia tietotarpeita, jotka täytetään vaihtamalla tietoja prosessin sisäisten ja ulkoisten roolien välillä. Liiketoimintaprosessissa liikkuu monenlaista tietoa, josta muodostuu joko pelkästään organisaation sisäisiä tai myös organisaation rajat rikkovia tietovirtoja. Osa liiketoi-

mintaprosessissa liikkuvasta tiedosta on dokumentoitava ja arkistoitava kirjanpitolainsäädännössä säädellyllä tavalla (FINLEX 2003).

Kuva 3 esittää kahden yksittäisen liiketoimintaprosessin tehtäviin liittyvää tietojen vaihtoa prosessiin osallistuvien roolien välillä. Kuvassa alempana esitetty laskutusprosessi on yleensä ajallisesti seurausta kuvassa ylempänä esitetystä myyntiprosessista. Yrityksestä riippuen näiden prosessien väliin saateetaan sisällyttää myös muita prosesseja. Kumpaankin prosessiin liittyy useita eri tehtäviä, joista kuvaan on valittu kolme dokumenttien välittämiseen ja käsitteilyyn liittyvää tehtävää: tuoteluettelon toimittaminen, tilauksen tekeminen ja laskuttaminen. Tehtävät muodostavat osapuolien välisiä tietovirtoja, jotka on kuvattu roolien välisillä nuolilla. Tieto liikkuu paitsi dokumenteissa mahdollisesti myös sanallisena informaationa.

Kuvasta ilmenevät liiketoimintaprosessiin liittyvät resurssit, joita ovat rooleissaan toimivat ihmiset ja dokumentit. Kuvan ulkopuolelle jäävät järjestelmät eli tekninen infrastruktuuri, jonka avulla dokumentit tuotetaan ja välitetään liiketoimintaprosessin osapuolten välillä.

Esimerkkiprosesseissa käsitellään kolmen tyyppisiä dokumentteja: tuoteluettelo, tilausta ja laskua. Eri dokumenttityypit sisältävät osittain samaa liike-



Kuva 3. Esimerkki kahdesta liiketoimintaprosessista ja niihin liittyvistä resursseista (roolit, dokumentit ja dokumenttien liiketoimintatieto).

toimintatietoa, kuten myyjän ja tilaajan tiedot, tuotetiedot ja tilausmäärät sekä hintatiedot. Esimerkkitapauksessa laskutusprosessi seuraa ajallisesti myyntiprosessia ja laskun liiketoimintatiedot ovat yhdistelmä myyntiprosessissa käsiteltyjä tuoteluettelon ja tilauksen tietoja, joita on summattu ja täydennetty laskutukseen liittyvillä tiedoilla.

Toisiaan vastaavissa liiketoimintaprosesseissa voi olla suuriakin yritysten välisiä eroavaisuuksia prosesseihin liittyvien roolien, prosesseissa tarvittavien dokumenttien ja dokumenttien sisältämien tietojen määrän suhteen. Lisäksi eroavaisuutta voi olla prosessin automatisointitason ja käytössä olevan teknologian suhteen. Dokumentit ja dokumenttien käsittelyyn liittyvät sovellukset voivat olla osa useaa eri liiketoimintaprosessia, esimerkiksi silloin, kun organisaatiossa on käytössä asiakkuudenhallintajärjestelmä, tuotetiedonhallintajärjestelmä tai toiminnanohjausjärjestelmä.

3.2 Liiketoimintadokumenttien käsittelyn osapuolet

Organisaatioiden väliset liiketoimintadokumenttien käsittelyn osapuolet ovat yksinkertaisimmillaan dokumentin tuottava ja dokumentin vastaanottava organisaatio. Näiden välillä vallitsee kuvan 4 mukainen monesta moneen suhde. Yksittäinen organisaatio tuottaa dokumentteja ja toimittaa ne usealle eri organisaatiolle. Vastaavasti yksittäinen organisaatio vastaanottaa dokumentteja monelta eri organisaatiolta.

Liiketoimintadokumenttien välittäminen on tapahtunut perinteisesti dokumentin tuottajalta vastaanottajalle paperimuodossa joko kädestä käteen tai postin välityksellä. Dokumenttien välittäminen myös digitaalisessa muodossa liiketoiminnan osapuolten välillä tietoverkkojen pitkin on ollut mahdollista jo pitkään esimerkiksi OVT-menettelyjä käyttäen.



Kuva 4. Liiketoimintadokumenttien käsittelyn osapuolet

Kun dokumentti on välitetty tuottavasta organisaatiosta vastaanottavalle organisaatiolle, jatkuu sen elinkaari kummassakin organisaatiossa. Esimerkiksi lasku jatkaa elinkaartansa sen tuottaneessa organisaatiossa myyntilaskuna ja vastaanottaneessa organisaatiossa ostolaskuna. Molemmissa organisaatioissa dokumentin välittämisen jälkeinen käyttötarve liittyy laskun osalta ensinnäkin kirjanpidolliseen käsittelyyn ja mahdollisesti myös reklamointi- ja perintätilanteisiin. Kun kirjanpidollinen käsittely on saatu valmiiksi, laskut arkistoidaan säädetyllä tavalla ennen niiden elinkaaren päättymistä lopulliseen tuhoamiseen. Myös viranomaisille tehtävät liiketoiminnan raportoinnit edellyttävät dokumenttien käyttöä, sillä esimerkiksi veroraportit perustuvat liiketoimintadokumenteista saataville tiedoille.

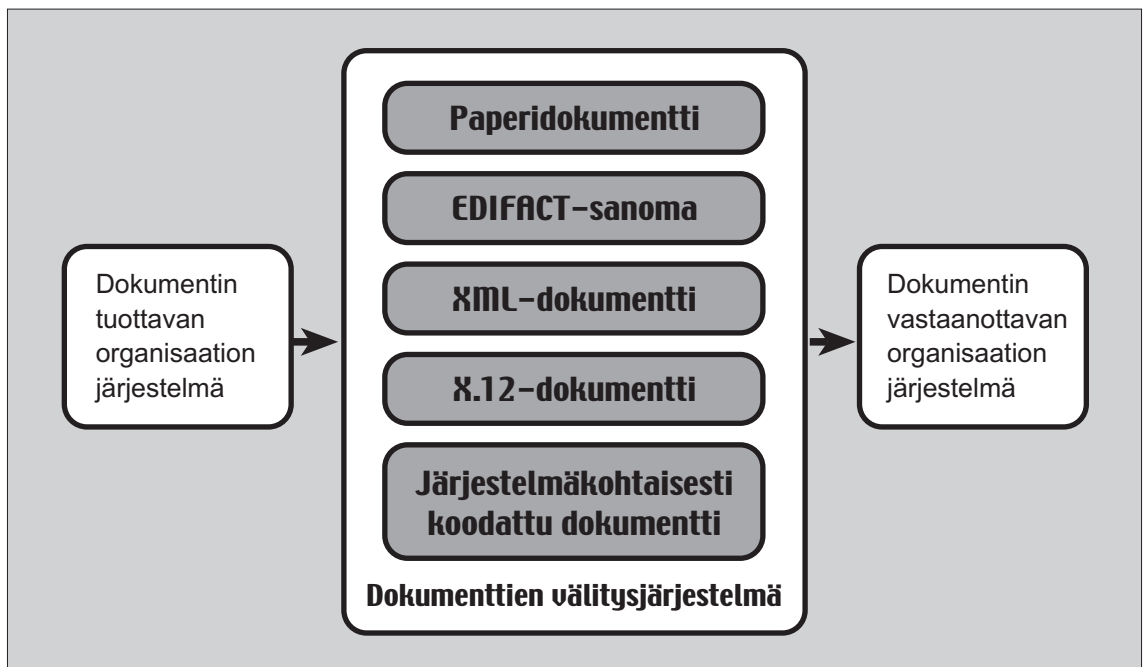
Liiketoimintadokumenttien tuottaminen ja käsittely liittyvät kiinteästi liiketoimintatapahtumia suorittaviin järjestelmiin (Wiggins 2000, 152). Esimerkiksi laskudokumentit tuotetaan yleensä suoraan laskutus- tai myyntireskontrajärjestelmästä ja kirjataan vastaanottavassa organisaatiossa dokumenttien kierrätys-, ostoreskontra- tai laskureskontrajärjestelmään. Jotta saumaton digitaalisten dokumenttien käsittely liiketoiminnan osapuolien välillä olisi mahdollista, tulisi organisaatioiden olla dokumenttien käsittelyssä samalla teknologisella tasolla ja noudattaa samoja dokumenttitiedon kuvaavia sääntöjä.

Käytännössä dokumenttien tuottamis-, välitys- ja vastaanottotavat ovat hyvin heterogeeniset, sillä organisaatioissa on käytössä monenlaisia standardeja, dokumenttiarkkitehtuureja, sovelluksia ja järjestelmiä. Ja vaikka kehityksen yleisenä tavoitteena onkin yhä useamman dokumentin tuottaminen ja käsittely pelkästään digitaalisessa muodossa, on edelleen pidettävä mielessä myös tarve dokumenttien esittämiseen analogisena paperimuodossa: kaikilla liiketoiminnan osapuolilla ei välttämättä ole valmiuksia vastaanottaa digitaalisia dokumentteja. Paperittomasta toimistosta on puhuttu jo parikymmentä vuotta, mutta silti totuttu tapa käsitellä dokumentteja paperimuodossa ja ylläpitää paperisia arkistoja jatkuu edelleen myös digitaalisuuden aikakaudella (Liu & Stork 2000, 94-96).

Vaikka liiketoimintadokumentit tuotetaankin tänä päivänä jo pääosin aina digitaalisina, tulee pahimmassa mahdollisessa tilanteessa dokumentin tuottavan organisaation toimittaa osalle vastaanottajista dokumentit paperisina,

osalle OVT-sanomina esimerkiksi EDIFACT- tai XML-muodossa, osalle X.12-dokumentteina ja osalle vastaanottavan järjestelmän ymmärtämässä järjestelmä- tai toimittajakohteisessa, ei-standardissa muodossa. Heterogeenisessä teknologiaympäristössä tällaista toimintamallia yksinkertaistavat sekä dokumentin tuottajan että vastaanottajan kannalta liiketoimintadokumenttien välityspalveluja tarjoavat organisaatiot (Anilinker 2002, BasWare 2002, Elisa 2002, ELMA 2002, TietoEnator 2002, Suomen Posti 2002). Välityspalvelujen tarjoajat vastaanottavat kuvan 5 mukaisesti organisaation järjestelmän tuottamat sovitun tyyppiset liiketoimintadokumentit sovitussa muodossa, muuntavat ne vastaanottajan haluamaan muotoon ja välittävät ne edelleen vastaanottajalle vastaanottajan haluamalla tavalla.

Rakenteisten dokumenttien käsittelylle on keskeistä monenlaiset muunnokset (Kuikka 1996; Lindén 1997). Dokumenttien välityspalveluja tarjoavien organisaatioiden järjestelmät muuntavat tarvittaessa dokumentit XML-, X.12- tai EDIFACT-määrityksen mukaisesta muodosta tai järjestelmäkohtaisesti koodatusta muodosta toiseen, vastaanottajan haluamaan muotoon (Diaz ym. 2002, 1047). Myös muunnokset paperidokumentista digitaaliseksi tai digitaalisesta dokumentista paperiseksi ovat mahdollisia.



Kuva 5. Liiketoimintadokumenttien välityspalvelua tarjoavien organisaatioiden järjestelmät välittävät dokumentit tuottajalta vastaanottajalle ja muuntavat eri muodoissa tuotetut dokumentit vastaanottajan haluamaan muotoon.

Vaikka välityspalvelujen tukemat dokumenttityypit ja muunnospalvelut ovat vielä varsin rajoittuneita, laajentaa palvelun käyttö digitaalisen dokumenttien välityksen ja käsittelyn organisaatioverkostoa huomattavasti. Välityspalveluiden etuna dokumentin tuottajalle on se, että dokumentit voidaan toimittaa välitysjärjestelmään yhdessä, dokumentin tuottavan järjestelmän mukaisessa muodossa riippumatta siitä, missä muodossa vastaanottaja dokumentit haluaa. Dokumentin vastaanottajan etuna on se, että liiketoimintadokumentit voidaan vastaanottaa digitaalisessa, dokumentteja vastaanottavan järjestelmän tukemassa muodossa. Täten digitaalisessa muodossa olevien liiketoimintadokumenttien käsittelyn piiriin saadaan yhä enemmän organisaatioita.

Suomessa dokumenttien välityspalveluja tarjoavat yritykset ovat solmineet keskenään yhteistyösopimuksia puhelinoperaattoreiden tai kansallisten postilaitosten tavoin. Vaikka vastaanottajana olisikin toisen dokumenttien välityspalvelun sopimusasiakas, on dokumenttien välittyminen digitaalisena tuottajalta vastaanottajalle mahdollista. Välityspalveluja tarjoavien organisaatioiden välisessä yhdysliikenteessä käytetään sovittua yhtenäistä dokumenttirakenteen kuvaavaa esitysmuotoa. (Verkkolaskukonsortio, 2001)

3.3 Liiketoimintadokumenttien rakenteistaminen

Jotta liiketoimintadokumenttien vastaanottavan osapuolen olisi mahdollista automatisoida dokumenttien käsittelyä, tulee liiketoimintadokumenttien olla digitaalisessa muodossa ja rakenteistettuna käsittelyn osapuolten välisen sopimuksen mukaisesti. Dokumentit voivat toki olla digitaalisina myös muussa kuin rakenteisessa muodossa, esimerkiksi tekstinkäsittelyohjelmien tallennusmuodossa tai paperidokumenteista skannattuina kuvina. Rakenteisuus tuo kuitenkin mukanaan paremmat mahdollisuudet automatisoida dokumenttien käsittelyä pelkkään digitaalisuuteen verrattuna.

Liiketoimintadokumenttien rakenteistamista on perusteltu toiminnan tehostumisella ja siitä aiheutuvalla kustannussäästöllä varsinkin silloin, kun organisaation käsittelemien dokumenttien määrä on suuri. Toimintaa tehostaa jo dokumenttien välittäminen digitaalisina tietoverkkoja pitkin, sillä se on monesti nopeampi vaihtoehto postin välityksellä tapahtuvalle dokumenttien toimittamiselle. Organisaatioiden sisällä toimintaa tehostaa se, että dokumenttien käsittelyä voidaan automatisoida, kun dokumentit ovat formaalisissa muo-

dossa. Etenkin suurissa ja maantieteellisesti erillään toimivissa yrityksissä dokumentit etenevät liiketoimintaprosesseissa nopeammin, mikäli paperidokumenttien sisäisen kierrättämisen sijaan käytetään kierrätysjärjestelmää, jossa dokumentteja käsitellään ja katsellaan digitaalisessa muodossa yhteisestä tietovarastosta.

Koska liiketoimintadokumenttien tuottaminen ja käsittely liittyvät kiinteästi liiketoimintatapahtumia suorittaviin järjestelmiin (Wiggins 2000, 152), on dokumenttien rakenteistamisprosessissa siten kyse joko liiketoimintatapahtumia suorittavien järjestelmien mukauttamisesta tai uusien, rakenteisten dokumenttien käsittelytapaa tukevien järjestelmien käyttöönotosta. Rakenteistaminen ei välttämättä muuta järjestelmien sisäistä liiketoimintatiedon esitysmuotoa. Sen sijaan rakenteistamisessa on ennemminkin kyse järjestelmien sisäisen liiketoimintatiedon esitysmuodon muuntamisesta järjestelmän rajapinnassa organisaatioiden väliseen rakenteiseen liiketoimintatiedon esitysmuotoon. Sovittua rakennetta käytetään siten järjestelmien välisessä dokumenttien välityksessä ja tiedonsiirrossa, mutta ei välttämättä niiden sisällä.

Liiketoimintadokumenttien rakenteistamisen aikajänne eri organisaatioiden välillä jo pelkästään laskujen kohdalla voi olla hyvin pitkä. Kokemuksia siirtymisestä organisaatioiden väliseen rakenteisten dokumenttien välittämiseen on saatu OVT-menettelyjen käyttöönotoista jo vuosien ajan: vielä tänäkään päivänä kaikki yritykset eivät ole investoineet OVT-järjestelmiin. OVT:n laajamittaisen yleistymisen esteiksi on mainittu muun muassa kompleksisuus ja soveltamisen vaatima tietotaidon puute sekä korkeat implementointikustannukset (Salminen 1994, 639). Lisäksi OVT:n haittapuolina on pidetty suljettua siirtokanavaa ja perinteisen OVT:n syntaksista johtuvaa tarvetta integrointiratkaisun räätälöimiseen jokaisen liiketoimintaprosessin osapuolen välillä (Glushko ym. 1999, 107). Kaikki nämä tekijät ovat rajoittaneet OVT-ratkaisut vain suurten organisaatioiden ja näiden tärkeimpien liiketoimintakumppanien ulottuville. Internetin käytön yleistymisen elektronisen liiketoiminnan toimintakanavana sekä XML-suosituksen tarjoaman avoimen dokumenttien rakenteen kuvaustavan on sanottu tuovan täysin digitaalisen liiketoimintadokumenttien käsittely-ympäristön myös perinteisen OVT:n käyttöönotosta kustannussyistä luopuneiden pienempien yritysten ulottuville (Glushko ym. 1999, 107; Banerjee & Kumar 2002, 102; Diaz ym. 2002, 1047).

Siirtyminen elektroniseen liiketoimintaan jollain liiketoimintaprosessin alueella, kuten esimerkiksi laskujen käsittelyssä, edellyttää yritysten välistä liiketoimintatiedon integraatiota monella eri tasolla (Stonebraker & Hellerstein 2001, 552). On huomioitava, että organisaatioiden välillä integroitava tieto on semanttisesti heterogeenista ja että käytössä on monia tiedon luokittelutapoja ja skeemoja (Stonebraker & Hellerstein 2001, 554). Organisaatiot käyttävät tietosisällöstään eri nimityksiä. Yksinkertaisena esimerkkinä tuotteen myyneen yrityksen viitteestä voidaan käyttää suomen kielessä nimitystä "Myyjän viite", "Viitteemme" tai myyntiorganisaatiota korostaen "Yritys X:n viite". Myös saman tietosisällön ymmärrys voi olla erilainen eri organisaatioiden välillä. Esimerkiksi seitsemän päivän huomautusaika voi tarkoittaa seitsemää kalenteripäivää viikonloput mukaan lukien tai seitsemää arkipäivää viikonloput pois lukien.

Toinen huomioitava asia siirryttäessä elektroniseen liiketoimintaan on organisaatioissa käytössä olevien järjestelmien väliset eroavaisuudet. Koska dokumenttien rakenteisuuden soveltaminen on varsin nuori alue muissa liiketoimintadokumenttien tuottamiseen ja käsittelyyn tarkoitetuissa järjestelmissä kuin OVT:hen kehitetyissä järjestelmissä, on järjestelmien kyky tuottaa tai vastaanottaa rakenteisen liiketoimintadokumentin tietosisältöä rajoittunutta.

Esimerkiksi laskujen kierrätys- tai ostoreskontrajärjestelmissä on saatettu rajoittaa järjestelmän käyttöliittymässä esitettäväksi yksittäisen laskun tiedoista vain laskun perustiedot kuten laskuttajaan, laskun tunnistamiseen ja maksamiseen sekä veroerittelyyn liittyvät tiedot. Laskun käsittelyyn oleellisesti liittyvät laskun erittelytiedot kyllä tallennetaan järjestelmään organisaatioiden välisessä tiedonsiirrossa käytettävässä muodossa, mutta näiden tietojen visuaalista esittämistä ei tueta järjestelmän käyttöliittymässä. Toisaalta vastaanotavassa organisaatioissa voi olla käytössä järjestelmä, johon ei olla vielä toteutettu rajapintaa rakenteisten dokumenttien automaattiselle sisäänluvulle. Tästä huolimatta organisaatio haluaa siirtyä vastaanottamaan laskunsa esimerkiksi Internetissä toimivan välityspalvelun välityksellä. Yllä kuvatuissa tilanteissa korostuu rakenteisessa muodossa olevan laskun visuaalinen esittäminen siten, että visualisointi tukee laskun käyttötilanteessa käyttäjän toimintatapaa ja tietotarpeita. Seuraavassa luvussa perehdytään tarkemmin niihin seikkoihin, jotka liittyvät rakenteisten liiketoimintadokumenttien visuaaliseen esittämiseen ihmiselle erityisesti silloin, kun visualisointiin sovelletaan tyylikieliä.

4 RAKENTEISTEN DOKUMENTTIEN ESITTÄMINEN

Lähtökohtana XML-muodossa olevan liiketoimintadokumentin esittämisen onnistumiselle voidaan pitää timanttimallin rooli- ja prosessinäkökulmasta tarkasteltuna sitä, että ihmiskäyttäjä saa halutussa liiketoimintaprosessin vaiheessa dokumentin visuaalisesta esityksestä tarvitsemansa informaation. Tämän lisäksi esityksessä nähdyn tietosisällön tulee vastata esitettävänä olevan XML-dokumentin tietosisältöä. Teknisestä näkökulmasta esittäminen on onnistunut silloin, kun päätelaitteen sovellusohjelma, jossa dokumenttia katsellaan, pystyy tulkitsemaan esityksen.

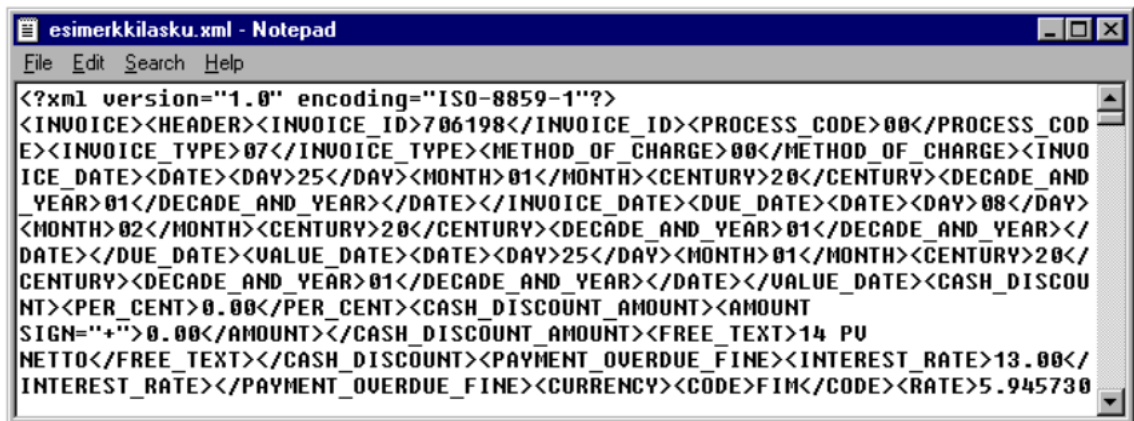
Tämän luvun aluksi selvitetään esimerkkien avulla rakenteisten liiketoimintadokumenttien visuaalista esittämistä ihmiselle. Tämän jälkeen luodaan lyhyt katsaus siihen, miten dokumentin visuaalisen ulkoasun konrolli siirtyy rakenteisuuden myötä myös dokumentin vastaanottajan hallintaan. Luvun loppupuolella kuvataan rakenteisten dokumenttien esittämiseen tarkoitettujen tyylikielien käytön periaatteita sekä näiden kielien käytön tuomia etuja ja haittoja. Luvun lopuksi esitellään muutamia XML-dokumenttien esittämisen määrittämiseen soveltuvia muunnos- ja tyylikieliä.

4.1 Rakenteisten liiketoimintadokumenttien esittäminen ihmiselle

Liiketoimintadokumentteja tuottavat ja käsittelevät pääsääntöisesti vain liike-toimintaprosessiin liittyvää tehtävää suorittavat henkilöt (Wiggins 2000, 152). Yksittäinen liiketoimintadokumentti tarjoaa käyttöhetkellään käyttäjälleen rajapinnan organisaation liiketoimintaan liittyvään tietokokonaisuuteen. Liiketoimintaprosessien etenemisen ja työtehtävien tehokkaan sujumisen kannalta onkin keskeistä, että dokumentit ja niiden sisältämä tieto ovat riittävän helposti tiedon tarvitsijan saatavilla ja ymmärrettävissä. Yksittäisen rakenteisen liiketoimintadokumentin luettavuus ja ymmärrettävyys ihmiselle liittyy kiinteästi siihen, miten dokumentti esitetään visuaalisesti.

XML-suosituksen mukaisissa liiketoimintadokumenteissa elementtien nimet kuvaavat yleensä dokumentin sisältämän liiketoimintatiedon semantiikkaa, eivät sitä, miten elementtien sisältämä tietosisältö tulisi visuaalisesti muotoilla. XML-dokumentit ovat sellaisenaankin käyttäjän tulkittavissa, mikäli dokumenttirakenteen merkintään käytetään liiketoiminta-alueella vakiintunut-

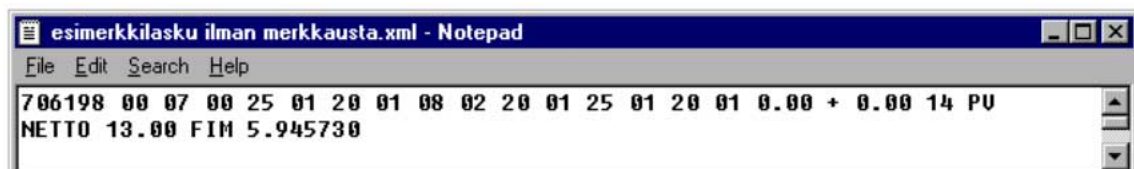
ta terminologiaa ja mikäli kuhunkin rakenneosaan käytetty terminologia vastaa semanttisesti rakenteen sisällä olevaa merkkietoa. Tulkinta on varsin suoraviivaista esimerkiksi silloin, kun laskun numero merkataan elementillä, jonka nimi on INVOICE_ID ja kun elementin arvona on todella laskun tunnistetieto, ei esimerkiksi tuotteen tilaajan laskulle määrittämä viitetieto. Kuva 6 esittää katkelmaa merkkijononäkymästä XML-dokumenttiin, joka on avattu yksinkertaiseen Notepad-tekstinkäsittelyohjelmaan. Merkkijononäkymässä elementit ja niiden sisältö muodostavat yhdessä informatiivisen kokonaisuuden, mutta dokumentin sisältämä tietosisältö ei ole ihmiselle kovinkaan havainnollista, sillä se hukkuu merkkauksen sekaan.



```
<?xml version="1.0" encoding="ISO-8859-1"?>
<INVOICE><HEADER><INVOICE_ID>706198</INVOICE_ID><PROCESS_CODE>00</PROCESS_CODE>
<INVOICE_TYPE>07</INVOICE_TYPE><METHOD_OF_CHARGE>00</METHOD_OF_CHARGE><INVOICE_DATE><DATE><DAY>25</DAY><MONTH>01</MONTH><CENTURY>20</CENTURY><DECADE_AND_YEAR>01</DECADE_AND_YEAR></DATE></INVOICE_DATE><DUE_DATE><DATE><DAY>08</DAY><MONTH>02</MONTH><CENTURY>20</CENTURY><DECADE_AND_YEAR>01</DECADE_AND_YEAR></DATE></DUE_DATE><VALUE_DATE><DATE><DAY>25</DAY><MONTH>01</MONTH><CENTURY>20</CENTURY><DECADE_AND_YEAR>01</DECADE_AND_YEAR></DATE></VALUE_DATE><CASH_DISCOUNT><PER_CENT>0.00</PER_CENT><CASH_DISCOUNT_AMOUNT><AMOUNT_SIGN="+">0.00</AMOUNT></CASH_DISCOUNT_AMOUNT><FREE_TEXT>14 PV NETTO</FREE_TEXT></CASH_DISCOUNT><PAYMENT_OVERDUE_FINE><INTEREST_RATE>13.00</INTEREST_RATE></PAYMENT_OVERDUE_FINE><CURRENCY><CODE>FIM</CODE><RATE>5.945730
```

Kuva 6. XML-dokumentin sisältö esitettynä merkkijononäkymänä yksinkertaisessa tekstinkäsittelyohjelmassa.

Jotta XML-dokumenttien sisältämä tieto olisi selkeämmin käyttäjän havaittavissa ja tulkittavissa, on dokumentin esittämisessä syytä kätkeä dokumenttirakenteen merkkkaus ja näyttää dokumentin katselijalle vain merkkietoa josain visuaalisessa olomuodossa. Kuvan 7 näkymässä on poistettu merkkkaus ja näkyvissä on pelkästään dokumentin sisältämä merkkietoa tyhjä-merkein



```
706198 00 07 00 25 01 20 01 08 02 20 01 25 01 20 01 0.00 + 0.00 14 PV
NETTO 13.00 FIM 5.945730
```

Kuva 7. Kuvassa 6 näkyvän XML-dokumentin sisältö esitettynä ilman merkkausta.

muotoiltuna. Ilman merkkausta pelkkä merkkitiety ei ole kuitenkaan yhtä informatiivista kuin yhdessä merkkauksen kanssa.

Kuvan 7 perusteella voidaan havaita, että merkkauksen poistaminenkaan ei paranna dokumentin käytettävyyttä, sillä samalla kun merkkaus poistetaan, katoaa osa dokumentin tietosisällön semantiikasta. Numeerisista arvoista on hankala päätellä tietosisällön merkitystä: vaikka osa sisällöstä havaittaisiinkin päivämäärä-arvoiksi, on niistä lähes mahdotonta päätellä, mikä on esimerkiksi laskun päivä, mikä arvopäivä ja mikä eräpäivä.

Kuvan 6 ja 7 visuaalisiin esityksiin verrattuna XML-dokumentin käytettävyyttä ihmiselle voidaan parantaa huomattavasti, kun dokumentin tietosisältö esitetään loogisiin kokonaisuuksiin ryhmiteltynä joko elementtien ja attribuuttien nimien mukaan otsikoiden tai käyttäjän kielen mukaisesti otsikoiden kuten kuvassa 8. Kuvan vasemmassa reunassa XML-dokumentin katkelma on esitetty XML Spy -nimisessä XML-dokumenttien käsittelyohjelmassa

The image shows two side-by-side windows. The left window is XML Spy displaying an XML invoice document. The right window is Microsoft Internet Explorer displaying the HTML rendering of the same invoice.

XML Spy - [mallilasku.xml]

INVOICE (1)
 (1) HEADER
 (1) INVOICE_ID 706198
 (1) PROCESS_CODE 00
 (1) INVOICE_TYPE 07
 (1) METHOD_OF_CHARGE 00
 (1) INVOICE_DATE
 DATE
 DAY 25
 MONTH 01
 CENTURY 20
 DECADE_AND_YEAR 01
 DUE_DATE
 DATE
 DAY 08
 MONTH 02
 CENTURY 20
 DECADE_AND_YEAR 01
 VALUE_DATE
 DATE
 DAY 25
 MONTH 01
 CENTURY 20
 DECADE_AND_YEAR 01
 CASH_DISCOUNT
 PER_CENT 0.00
 CASH_DISCOUNT_AMOUNT
 AMOUNT
 SIGN
 FREE_TEXT 14 PV NETTO
 PAYMENT_OVERDUE_FINE

Lasku 706198 - Microsoft Internet Explorer

Malliyritys Oyj LASKU FAKTURA
 Päivämäärä-Datumum 25.01.2001 Laskunro-Fakturas num 706198 Asiakasnro-Kundnum 859522
 Viiteenno-Vår referens Käsitteljä-Handläggare Majja Malli
 Viiteenno-Er referens MARI MALLI

Laskutusosoite-Faktureringsadress ESIMERKKI OYJ MIKA MALLI PL 1 ESIMERKKITIE 71 00001 MALLILA
 Toimitusosoite-Leveransadress ESIMERKKI OYJ MIKA MALLI PL 1 ESIMERKKITIE 71 00001 MALLILA

Maksaja-Betalare ESIMERKKI OYJ MIKA MALLI PL 1 ESIMERKKITIE 71 00001 MALLILA
 Maksuehdot-Betalningsvillkor 14 PV NETTO 0.00% 0.00 FIM
 Eräpäivä-Förfallodag 08.02.2001 Viivästyskorke-Dröjsmåttaranta 13.00 %

Pankkiyhteys-Bankförbindelse		Sis. ALV	ALV 0
PANKKI	123456-12345	inkl. MOMS	MOMS 0
Eräpäivä-Förfallodag	08.02.2001	Laskun rivit yhteensä-Totalt	601.80 FIM
Viitenumero-Referensnummer	859522706198204	Vakuutus-Försäkring	0.00 FIM
		ALV-MOMS	132.41 FIM
		Pyöristys-Avrundning	0.01 FIM
		YHTEENSÄ-TOTALT	734.20 FIM

ALV-MOMS Peruste-Grund 201.03 % Summa 132.41

Kuva 8. XML-dokumentin katkelma esitettynä tietosisältönsä osalta vasemmassa XML-dokumenttien käsittelyohjelmassa ja oikealla HTML-selaimessa, kun dokumentti on ensin muunnettu HTML-kielelle ja tietosisällöt on otsikoitu muunnoksessa uudelleen.

sa yhden ohjelmaan sisään rakennetun esitysmäärityksen mukaisesti. Tietosisällöt on otsikoitu elementti- ja attribuuttitunnisteita käyttäen. Kuvan oikeassa reunassa sama XML-dokumentti on muunnettu HTML-dokumentiksi. Muunnoksessa dokumentin tietosisältö on otsikoitu käyttäjän kielen mukaan, tässä tapauksessa sekä suomen että ruotsinkielisin otsikoin. Lisäksi dokumenttiin on lisätty laskuttajan logo.

Kuvat 6, 7 ja 8 havainnollistavat, miten samasta XML-dokumentista voidaan tuottaa erilaisia visuaalisia esityksiä käytettävyyttä asteittain parantaen. Kuvan 8 HTML-dokumentin esitysmuoto on määritetty XSL-kielillä tyylisivulla. Kun XML-dokumentti visualisoidaan käyttöä varten kuvan 8 osoittamalla tavalla, liitetään tyylisivussa kuvattu ulkoinen esitysmuoto prosessointivaiheessa dokumentin sisältöön joka kerran, kun dokumenttia käytetään. Prosessoinnissa voidaan samalla suodattaa dokumentin sisällöstä joitain tietoja pois tai koostaa dokumentin visuaalinen esitys useasta erilleen varastoidusta osasta. Esimerkiksi kuvassa 8 HTML-muotoisesta esityksestä on suodatettu PROCESS_CODE ja METHOD_OF_CHARGE -elementtien sisältö pois ja koostettu esitys sekä XML-dokumentista että binäärimuodossa olevasta logotiedostosta. Kuten kuvasta 8 havaitaan, voidaan esityksiin liittää dokumenttisisällön ja ulkoisten entiteettien lisäksi erilaisia visuaalisia elementtejä, kuten esimerkiksi värejä ja viivoja sekä esittää tekstisisältö erilaisilla kirjasimilla.

Lyytikäinen ja Tiitinen (2001, 26-27) ovat koonneet rakenteisten dokumenttien käytettävyyteen liittyviä suunnitteluohjeita. Heidän mukaansa dokumenttien esittämisessä tulee huomioida ensinnäkin tuki kaikille käyttäjäryhmille sekä dokumenttien katselemiseen soveltuville selaimille. Päätelaitteet ja käyttöympäristöt ovat muuttumassa yhä heterogeenisemmiksi, sillä Internetin käyttö on laajenemassa perinteisestä PC:n WWW-selainympäristöstä mobiilipäätelaitteisiin ja myös Web-televisioon. (Lyytikäinen & Tiitinen 2001, 26) Vaikka erilaiset päätelaitteet tekevät vasta tuloaan liiketoimintadokumenttien käytön alueelle, on käyttöympäristön heterogeenistuminen syytä ottaa jo tämän hetken toteutuksissa huomioon. Etenkin mobiilipäätelaitteet laajentavat dokumenttien käytettävyyttä, sillä dokumentit ovat käytettävissä juuri siellä, missä niitä tarvitaan ja juuri silloin, kun niitä tarvitaan (Lamming, Eldridge, Flynn, Jones & Pendlebury 2000, 331). Dokumenttien käyttöympäristön heterogeenistuminen asettaa dokumenttien käytön näkökulmasta vaatimuksia ja rajoitteita sekä dokumenttien esitysmuodolle että dokumentissa esitettävän

liiketoimintatiedon määrälle (van Ossenbruggen, Geurts, Cornelissen, Hardman & Rutledge 2001, 479).

Teknisen ympäristön lisäksi rakenteisten dokumenttien käytettävyyteen liitetyissä suunnitteluohjeissa kiinnitetään huomiota myös sosiaalisiin tekijöihin. Laajan käyttäjä- ja selaintuen lisäksi itse dokumenttien sisältöön tulee tarjota erilaisia näkymiä ja esitysmuotoja. Tällä tavoin voidaan huomioida ihmisten erilaisia kykyjä ja rajoitteita sekä organisationaalisia tekijöitä. Rakenteinen teksti tarjoaa itsessään parempia mahdollisuuksia toteuttaa erilaisia näkymiä dokumentteihin erilaisille käyttäjäryhmille jopa niin, että joissain tapauksissa käyttäjät voisivat itse määrittellä näkymiä. (Lyytikäinen & Tiitinen 2001, 27)

Vaikka erilaisten visuaalisten esitysten tuottaminen on rakenteisille dokumenteille periaatteessa helppoa, edellyttää jokainen erilainen näkymä tai esitysmuoto joko omaa tyylikielellä toteutettua tyyllisivumäärittystä tai muulla tavoin toteutettua esitysmuunnosta. Dokumenttien käytettävyyden suunnitteluperiaatteiden huomioimisella voi olla myös negatiivisia vaikutuksia. Dokumenttien sisältöön tarjottavien erilaisten näkymien ja esitysmuotojen seurauksena saattavat saman dokumentin käsittelijät kahdessa eri organisaatiossa nähdä erilaisen esityksen samasta liiketoimintadokumentista. Myös yksittäinen käyttäjä voi nähdä erilaisen esityksen samasta dokumentista eri käyttötilanteissa esimerkiksi katsellessaan dokumenttia PC:n näyttöpäätteellä ja mobiilipäätelaitteen näytöllä. Tällöin käyttäjä ei välttämättä tunnista dokumenttia samaksi eri käyttötilanteissa.

Dokumenttien käytettävyyden suunnitteluperiaatteiden toteutumista liiketoimintadokumenttien alueella rajoittaa se, että organisaatioissa käytössä olevat liiketoimintadokumenttien tuottamiseen ja käsittelyyn tarkoitetut tekniset järjestelmät ovat vasta kehitysvaiheensa alussa rakenteisten dokumenttien monimuotoisen esittämisen tukemisessa. Käytännössä tämän hetkisistä liiketoimintadokumenttien käsittelyyn kehitetyistä järjestelmistä vain osa pystyy tukemaan suoraan XML-muodossa olevien dokumenttien esittämistä esimerkiksi XSL-kielisten tyyllisivujen avulla. Sen sijaan joidenkin järjestelmien tuki rajoittuu ainoastaan XML-muodossa olevasta dokumentista muunnetun ja tallennetun erillisen kuvatiedoston visualisointiin. Kuvatiedostojen tallennusmuotojenkin tuki saattaa olla rajoittunutta siten, että järjestelmän käyttö-

liittymässä on mahdollista katsella vain HTML- (Hypertext Markup Language; Berners-Lee & Connolly 1995), TIFF- (Tagged Image File Format; TIFF 1992) tai PDF-muodossa (Portable Document Format; Meehan, Taft, Chernicoff & Rose 2000) olevia kuvadokumentteja. Kaikkein tiukimmin rajoittuneita ovat järjestelmät, jotka eivät kykene millään tavoin visualisoimaan XML-dokumentteja eivätkä mihinkään muotoon tallennettuja kuvadokumentteja.

Tavoiteltavana lopputuloksena liiketoimintadokumenttien käsittelyyn tarkoitettujen järjestelmien tulisi kyetä eri tallennusmuodoissa (mm. XML, HTML, PDF, TIFF) olevien liiketoimintadokumenttien visuaalisten esitysten tukemiseen. Tähän tavoitteeseen tähtää tutkimustyö, jota tehdään sellaisten selainohjelmien kehittämiseksi, jotka pystyvät näyttämään eri tallennusmuotoihin koodattuja dokumentteja (Phelps & Wilensky 2001). Tällöin sovellukseen voitaisiin sisällyttää yksi, eri tallennusmuotoja tukeva selainohjelma sen sijaan, että niihin on nyt sisällytettävä oma selainohjelma kutakin tuettua eri tallennusmuotoa varten.

4.2 Dokumentin vastaanottajan kyky hallita visuaalisuutta

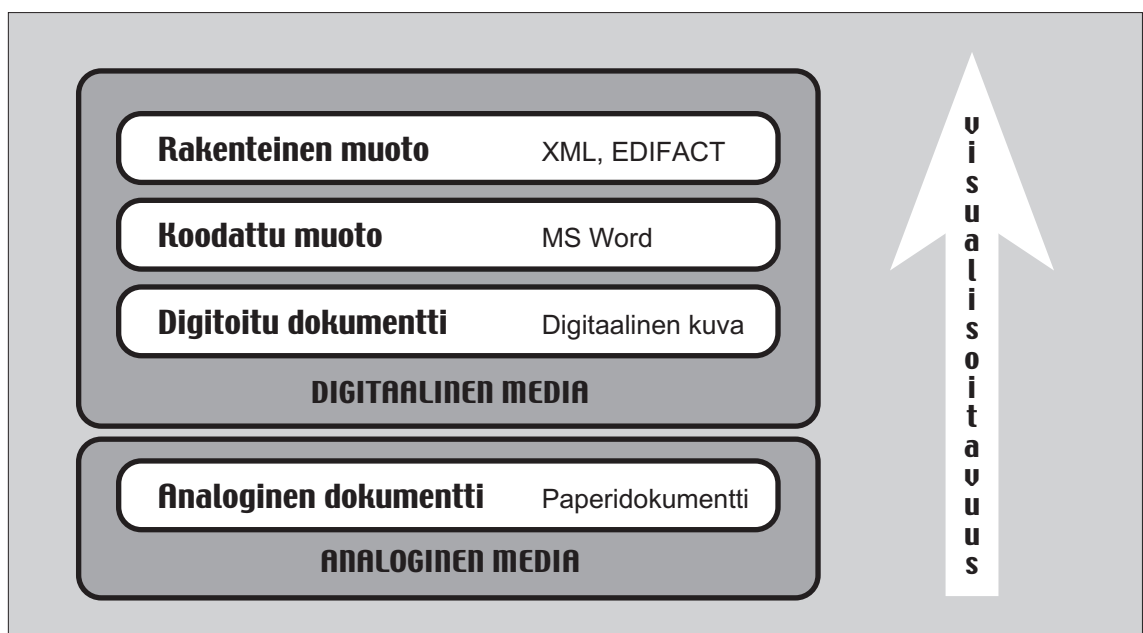
Perinteisesti paperidokumentteja käsiteltäessä dokumentin tuottajan organisaatio on määrittänyt, miltä dokumentin tulee visuaalisesti näyttää. Tämän vuoksi myös liiketoimintadokumenteista on muodostunut etenkin suurissa organisaatioissa osa yrityksen imagon rakennusvälineistöä. Markkinointi- ja viestintätahot ovat usein pyrkineet päättämään, miltä organisaatiosta ulos lähtevät dokumentit näyttävät.

Kun dokumentteja tuottavassa organisaatiossa siirrytään tuottamaan rakenteisia dokumentteja, katoaa markkinointi- ja viestintäosastojen täydellinen kontrolli dokumenttien visuaalisen esittämisen suhteen. Tämä johtuu siitä, että rakenteiseen dokumenttiin sisällytetään yleensä vain dokumentin varsinainen tietosisältö, kuten liitteen 1 mallidokumentissa. Sen sijaan rakenteinen dokumentti ei yleensä sisällä tietoa siitä, miten dokumentti tulisi katselijalleen esittää. Koska rakenteisen dokumentin visuaalinen olomuoto kiinnittyy vasta dokumentin katseluhetkellä, siirtyy visualisoinnin kontrolli osittain myös dokumentin vastaanottajalle. Rakenteisuuden myötä menetetään siten dokumentin tuottajan täydellinen kyky hallita dokumenttien visuaalista esittämistä joka kerran, kun dokumenttia katsellaan.

Seuraavassa luodaan lyhyt silmäys siihen, miten visuaalisuuden kontrolli siirtyy dokumentin formaalisuustason kasvaessa vähitellen myös dokumentin vastaanottajalle. Kuvassa 9 esitetään mukaelma Tyrväisen ja Päivärinnan (1998, 105) tekemästä dokumenttien formaalisuusasteeseen perustuvasta luokittelusta, joka avulla voidaan havainnollistaa dokumenttien visualisoitavuutta. *Visualisoitavuudella* tarkoitetaan tässä mahdollisuutta esittää dokumentti eri tavoin eri käyttötilanteissa. Kuvassa 9 dokumenttitiedon formaalisuusluokitus on esitetty siten, että vähiten formaalein, analoginen dokumenttiluokka on alimpana ja formaalein dokumenttiluokka, rakenteinen dokumentti, ylimpänä.

Koska dokumenttien edellytetään olevan tallennettuja tietokokonaisuuksia, tulee niiden olla vähintään analogisessa muodossa, esimerkiksi paperilla. Paperinen dokumentti näyttää aina siltä, miltä dokumentin tuottaja haluaa sen näyttävän, sillä ulkoasu kiinnittyy dokumenttiin aina tuottamisvaiheessa tuottamistavasta riippumatta. Ulkoasu pysyy muuttumattomana koko dokumentin elinkaaren ajan.

Digitaalisista dokumenteista vähiten formaalein on digitoitu dokumentti, esimerkiksi faksi tai analogisesta digitaaliseksi skannattu kuva, kuten skannattu



Kuva 9. Dokumenttien formaalisuusluokitus (Tyrväinen 2002, 11 ja Tyrväinen & Päivärinta 1998, 105 mukailten)

paperilasku. Digitoitua dokumenttia formaalimpi on koodattu dokumentti. Koodaus voi olla joko jonkin standardin mukainen tai valmistajakohtainen, esimerkiksi tekstinkäsittelyohjelman tallennusmuotoon. Digitoidussa dokumentissa ja koodatussa muodossa sekä sovellusohjelman ymmärtämä esitysmuoto että visuaalinen ulkoasu on koodattuna dokumenttiin. Kun dokumenttia katsellaan sen tallennusmuodon ymmärtävässä sovellusohjelmassa, näyttää se ainakin periaatteessa samalta, miltä dokumentin tuottaja haluaa dokumentin näyttävän. Häiriötekijöitä voi kuitenkin esiintyä. Näihin formaalisuusluokkiin kuuluvien dokumenttien ulkoista esitysmuotoa on kuitenkin mahdollista muokata dokumentin elinkaaren aikana niiden tallennusmuotoa ymmärtävien sovellusohjelmien toiminnallisuuden rajoissa.

Rakenteisia muotoja ovat OVT-järjestelmien EDIFACT- tai XML-muotoisten dokumenttien lisäksi esimerkiksi WWW-kauppapaikan tilauslomakkeet. Tässä tutkielmassa keskitytään kuitenkin käsittelemään pelkästään XML-dokumentteja. Kuten jo aiemmin todettiin, ei yksittäinen rakenteinen XML-dokumentti kerro yleensä tietosisältönsä tai rakenteen merkkauksen osalta mitään dokumentin visuaalisesta esittämistavasta, vaan sisältää ainoastaan liiketoimintaan liittyvää informaatiota. Visuaalisen ulkoasun määrittäminen tuotetaan eri prosessissa kuin dokumentin tietosisältö. Tätä tarkoitusta varten on kehitetty muunnos- ja tyylikieliä (Hirvonen 1997, 26). Muunnos- ja tyylikielillä toteutettua dokumentin tietosisällöstä erillään olevaa ulkoasun määrittämistä kutsutaan *tyylisivuksi* (stylesheet). Tyylisivun määrittämään ulkoasuun yhdistetään dokumentin tietosisältö, kun dokumenttia prosessoidaan käyttöä varten.

Tieto ulkoasun muotoiluun käytettävästä tyylisivusta voidaan liittää rakenteiseen XML-dokumenttiin jo dokumentin tuottamisvaiheessa (Clark 1999a). Tämä edellyttää kuitenkin sitä, että tyylisivu on varastoitava ja välitettävä yhdessä dokumentin kanssa. Tyylisivun tulisi olla myös saatavilla yhtä kauan kuin itse dokumentin. Näin ei kuitenkaan aina ole. Lisäksi dokumentin tuottajan määrittämä viittaus tyylisivuun on teknisen suosituksen näkökulmasta vain ehdotelma siitä, mitä tyylisivua dokumentin esittämisessä tulisi soveltaa. Tyylisivun viittaussuositus (Clark 1999a) jättää dokumentin käyttäjän ja esittämiseen käytettävän sovellusohjelman päätettäväksi, hyödynnetäänkö dokumentin esittämisessä ehdotettua tyylisivua vai ei. Suositus siis antaa dokumenttien vastaanottajalle mahdollisuuden visualisoida kukin vastaanottamansa rakenteinen dokumentti haluamallaan tavalla.

Dokumentin visuaalisen esittämisen kontrollointi katoaa käytännössä dokumentin tuottajan ulottuvilta esimerkiksi silloin, kun rakenteinen dokumentti välitetään toiseen organisaatioon kuuluvalle vastaanottajalle, joka katselee dokumenttia tuottamisympäristöstä poikkeavassa teknologiaympäristössä, esimerkiksi eri toimittajan järjestelmässä. Rakenteisten dokumenttien käyttöön siirtyvien organisaatioiden tulisikin tiedostaa ja hyväksyä se tosiasia, että dokumentti voidaan esittää elinkaarensa eri vaiheissa visuaalisesti toisin, kun dokumentin tuottanut organisaatio sen haluaisi esitettävän. Mikäli näin ei haluta tapahtuvan, on siitä sovittava dokumentin tuottajan ja vastaanottajan välillä.

4.3 Muunnos- ja tyylikielien käyttö

Rakenteisten dokumenttien esittämisen prosessi muodostuu käsitteellisellä tasolla kahdesta osasta: muuntamisesta ja muotoilusta. Muunnoksella ja muotoilulla rakenteisen dokumentin tietosisällöstä saadaan luotua haluttu esitys. Tarve vaihtaa rakenteisen dokumentin tietosisältö useampaan kuin yhteen järjestykseen edellyttää joustavaa muunnosta. Tarve esittää tietosisältöä useammassa kuin yhdessä ilmenemismuodossa, kuten visuaalisena tai auralisena ilmentymänä, vaatii muotoilua. (Holman 2002)

Rakenteisten dokumenttien esittämissääntöjen kuvaamiseen voidaan käyttää muunnosten osalta muunnoskieliä ja muotoilun osalta tyylikieliä. Muunnoskielien käsite kuitenkin sivuutetaan usein ja puhutaan pelkästään tyylikielistä, vaikka tarkoitetaan muotoilun lisäksi myös muunnoksen määrittämistä. Tämän vuoksi myös käsite tyylisivu voi olla hieman harhaanjohtava, sillä tyylisivu voi sisältää myös muunnosmäärittäjiä. Tässä tutkimuksessa halutaan kielistä puhuttaessa pitää erillään muunnoksen ja muotoilun käsitteet, sillä jako selventää rakenteisen dokumentin esitysprosessin kaksivaiheisuutta. Tämän vuoksi *muunnoskielillä* tarkoitetaan tässä tutkimuksessa rakenteisten dokumenttien sisältämän dokumenttirakenteen ja tietosisällön muuntamiseen tarkoitettuja kieliä. *Tyylikielet* ymmärretään tässä tutkimuksessa suppeassa merkityksessä ja niillä tarkoitetaan siten pelkästään rakenteisten dokumenttien sisältämän tiedon ulkoisen esitysmuodon määrittämiseen tarkoitettuja kieliä. Ulkoisella esitysmuodolla voidaan ymmärtää joko visuaalista tai auralista esittämistä, joista tässä tutkimuksessa rajoitutaan vain visuaaliseen.

Muunnoskieliä käytetään esityksen kohteena olevan dokumentin loogisen rakenteen muuntamiseen halutun esityksen mukaiseksi loogiseksi rakenteeksi. Esityksen looginen rakenne syntyy esityksen määrittävän organisaation tavasta jäsentää esityksen kohteena olevaa tietoa. Liiketoiminta-alueella käytettäväksi sovittujen dokumenttityyppien looginen rakenne voi poiketa hyvinkin paljon siitä rakenteesta, mikä halutaan esittää visuaalisesti. Esimerkiksi XML-dokumentin loogisessa rakenteessa saattaa olla ensimmäisenä dokumentin tuottaneen organisaation yhteystiedot, mutta dokumenttia visualisoidessa halutaankin esittää ensimmäiseksi dokumentin vastaanottajan tiedot.

Muunnoskielillä voidaan tietosisällön keskinäisen esitysjärjestyksen lisäksi muuntaa dokumentin alkuperäistä tietosisältöä. Tietosisällön muunnolla tarkoitetaan lisättyä, vaihdettua, poistettua tai toistettua tietosisältöä verrattuna muunnoksen kohteena olevan dokumentin tietosisältöön. Lisätty tietosisältö voi olla staattista vakiotekstiä, kuten esimerkiksi otsikkotekstejä. Lisätty tietosisältö voi olla myös alkuperäisessä dokumentissa olevien tietosisältöjen perusteella tuotettua sisältöä, kuten esimerkiksi arvonlisäveron määrä laskettuna verottoman hinnan ja arvonlisäveroprosentin perusteella. Vaihdetulla tietosisällöllä tarkoitetaan tietosisällön muuntamista toiseksi. Esimerkiksi tiettyssä rakenteessa oleva koodiarvo "01" voidaan muuntaa visuaalisessa esityksessä muotoon "HYVITYSLASKU". Tietosisällön poistolla tarkoitetaan sitä, että kaikkia alkuperäisessä dokumentissa olevia tietosisältöosia ei määritetä esitettäväksi. Dokumentissa saattaa olla esimerkiksi sovellusten väliseen kommunikointiin tarkoitettua tietoa, jota ei ole tarkoitus näyttää normaalilanteissa ihmiselle. Tietosisällön poisto voi myös aiheutua tahattomasti siitä, että dokumentin rakennesääntömääritykseen on lisätty jokin uusi rakenne, mutta muunnosmäärityksestä on jäänyt muutosta vastaava ylläpito toteuttamatta. Toistetulla sisällöllä tarkoitetaan muunnoksen kohteena olevan dokumentin osien esittämistä useammin kuin yhden kerran. Esimerkiksi laskuttajan nimi saatetaan haluta esittää visuaalisesti sekä dokumentin yläreunassa että alareunassa.

Muunnoskieliä voidaan käyttää myös muulloin kuin XML-dokumentin visuaalista tai auralista esittämistä määritettäessä. Muunnoskielillä voidaan määrittää esimerkiksi pelkästään sisältömuunnoksia kahden eri dokumenttityyppiä olevien XML-dokumenttien välillä tai XML-dokumentista rakenteettomaksi tekstidokumentiksi.

Silloin, kun muunnoskieliä käytetään dokumentin esittämisen määrittämiseen, sisältävät muunnoskielillä tehdyt määrytykset myös jollakin tyylikielillä ilmaistun dokumentin esityksen määrytyksen. Muunnoskielellä määritetään siten, mitkä XML-dokumentin rakenteiden tietosisällöt esitetään ja missä järjestyksessä. Tyylikielimäärytykset kuvaavat muunnoksen lopputuloksen esitystavan. Muunnosprosessin lopputuloksena syntyvän dokumentin tai dokumenttia vastaavan muistikuvauksen tietosisältö voi poiketa tietomääränsä tai järjestyksen suhteen paljonkin alkuperäisestä muunnoksen kohteena olevasta XML-dokumentista.

Tyylikielien kehittyminen liittyy yleiseen dokumenttirakenteen merkkauksen ja erityisesti SGML-standardin (ISO 8879 1986) kehittämiseen. SGML-standardin kehitystyön keskeisenä ajatuksena oli dokumentin rakenteen ja tietosisällön erottaminen sen ulkoisesta esittämisestä. SGML-standardin kehitystyön yhteydessä kehitettiin myös ensimmäinen tyylikieli DSSSL (Document Style Semantics and Specification Language; ISO 10179 1996), joka oli tarkoitettu SGML-dokumenttien esittämisen määrittämiseen. Vaihtoehtona tyylikielien käytölle on sovelluskohtainen koodaus, jolla kuvataan kunkin dokumentin ulkoinen erityismuoto. Tällöin dokumentin esittämisen rajoitteena on valmistaja- tai toimittajakohtainen sidonnaisuus koodausta tulkittaessa.

Tyylikieliä käytettäessä on huomioitava, että tyylikielet ovat vain säännöstöjä rakenteisten dokumenttien esittämisen määrittämiseen. Tyylimäärytykset ovat tyylikielien sovelluksia ja lopullisen dokumentin esittämisen aikaansaamiseen tarvitaan aina sovellusohjelma, joka osaa tulkita tyylikielen mukaisia määrytyksiä ja muotoilla tyylikielellä kuvatun esityksen tyylikielen säännöstössä sovitulla tavalla.

Tyylimäärytykset tallennetaan usein dokumentista erillisiin tyylisivuihin, mutta tyylimäärytyksiä voidaan myös sisällyttää dokumenttiin, jonka esittämisen muotoiluun ne on tarkoitettu. Tyylimäärytysten tallentamisessa erilleen dokumentista tyylisivuiksi on se etu, että yhtä tyylisivua voidaan käyttää usean samoja rakennesääntöjä noudattavan dokumentin esittämiseen. Toisaalta yhden dokumentin esittämiseen voidaan käyttää useita erilaisia tyylisivuja, jolloin dokumentti voidaan esittää eri tavoin eri käyttäjäryhmille tai eri käyttötilanteissa.

4.4 Muunnos- ja tyylikielien etuja

Muunnos- ja tyylikielien käytöstä dokumenttien käsittelyssä on seuraavia etuja:

- tiedon uudelleenkäyttö
- mahdollisuus erilaisiin tiedon esitysmuotoihin ja sitä kautta lisääntyvä dokumenttien käytettävyys
- dokumenttien ulkoisen esitysmuodon määrittysten standardointimahdollisuus ja ylläpidon helpottuminen
- laitteisto- ja ohjelmistoriippumattomuus
- dokumentin esityksen myöhäinen sidonta.

Seuraavassa käydään lyhyesti läpi nämä hyötynäkökohdat.

Tiedon uudelleenkäytöllä tarkoitetaan sitä, että saman dokumentin osia käytetään eri tarkoituksiin (Grosso & Walsh 2000). Tyypillinen esimerkki tiedon uudelleenkäytöstä on dokumentin lukujen otsikoiden perusteella luotava sisällysluettelo. Lukujen otsikot tallennetaan vain yhteen kertaan, mutta otsikotekstit esitetään sekä lukujen alussa että sisällysluettelossa.

Erilaisilla esitysmuodoilla tarkoitetaan sitä, että muunnos- ja tyylikieliä käytettäessä dokumenttiin voidaan liittää useita tyylimäärittäjiä ja esittää dokumentti eri tavoin erilaisille käyttäjille tai erilaisissa käyttötilanteissa (Hirvonen 1997, 11-12). Rakenteisuuden sekä tyyli- ja muunnoskielien käytön yleisenä etuna on dokumenttien käyttöalueen laajeneminen. Koska visuaalinen esitysmuoto kiinnitetään dokumenttiin joka kerran, kun dokumenttia prosessoidaan, voidaan dokumentit muuntaa niitä käytettäessä erilaisille päätelaitteelle ja käyttöympäristölle ja käyttäjille sopivaan ulkoasuun (Lie & Saarela 1999, 96).

Ulkoisen esitysmuodon määrittämisen standardoinnilla tarkoitetaan sitä, että esimerkiksi organisaatioissa voidaan muodostaa kontrolloidusti dokumenttien ulkoisen esitysmuodon määrittävät tyylisivut ja käyttää näitä mahdollisuuksien mukaan aina, kun dokumenttia esitetään (Grosso & Walsh 2000). Tätä kautta saavutetaan myös ylläpidon helpottuminen, sillä esittämisen liittyvät muutokset tarvitsee toteuttaa vain tyylisivuihin, ei jokaiseen dokumenttiin erikseen.

Laitteisto- ja ohjelmistoriippumattomuus saavutetaan ainakin periaatteessa standardoituja muunnos- ja tyylikieliä käytettäessä. Laitteisto- ja ohjelmistoriippumattomuus on hyödyllinen ominaisuus etenkin silloin, kun samoja dokumentteja esitetään eri organisaatioissa ja eri laitteisto- ja ohjelmistokokoonpanoilla (Hirvonen 1997, 12). Tämä etu saattaa kuitenkin käytännössä jäädä saavuttamatta, mikäli sovellukset tukevat vain osittain muunnos- ja tyylistandardia.

Dokumentin esittämisen myöhäisellä sitomisella tarkoitetaan sitä, että tyylikielillä määritetty dokumentin esitys muotoillaan vasta dokumenttia käyttävän päätelaitteen sovellusohjelmassa ja päätelaitteen ominaisuuksien ehdoilla. Samalla, kun ulkoasun viimeistelyyn sovelletaan esitysympäristön reunaehtoja, myös vähennetään dokumentin prosessointiin käytettävän palvelimen kuormitusta, sillä osa esitysprosessoinnista siirretään päätelaitteen sovellusohjelman tehtäväksi.

4.5 Muunnos- ja tyylikielien haittoja

Muunnos- ja tyylikielien käytöstä dokumenttien käsittelyssä on monien etujen lisäksi myös seuraavia haitallisia seurauksia:

- dokumentin visuaaliseen esittämiseen tarvitaan aina dokumentin lisäksi myös tyylisivu
- dokumentin käyttöön liittyy muunnosprosessi
- dokumentin käyttäjällä tulee olla tyylikielen syntaksin ymmärtävä sovellus
- tyylikielien syntaksin mukaisen esittämisen tulkinta tapahtuu vasta päätelaitteen sovelluksessa
- tyylisivuihin liittyy tekijänoikeudellisia ja taloudellisia kysymyksiä ja ongelmia

Seuraavassa käydään lyhyesti läpi nämä haittatekijät.

Tyyli- ja muunnoskielien käytön haittana on ensinnäkin se, että dokumentin visuaaliseen esittämiseen tarvitaan aina itse dokumentin lisäksi myös esittämissäännöt sisältävä tyylisivu. Tämä saattaa lisätä siirrettävän tiedon määrää, sillä varsinaisen dokumentin lisäksi on siirrettävä myös tyylisivu. Tyylisivua ei kuitenkaan tarvitse välttämättä siirtää joka kerran dokumenttien lähettämi-

sen yhteydessä, mikäli vastaanottavassa järjestelmässä on ratkaistu tyyllisivujen varastointi- ja ylläpitomenettely. Tällöin on dokumentin siirron yhteydessä siirrettävä vain uudet tai muuttuneet tyyllisivut tai järjestettävä tyyllisivujen päivitys jollain muulla tavoin.

Toinen häiritsevä tekijä on se, että rakenteisen dokumentin käyttöön liittyy aina muunnosprosessi. Jos dokumentin esittämiseen tarkoitettua tyylimääritystä ei ole saatavilla käyttötilanteessa, ei käyttäjälle pystytä tarjoamaan halutun mukaista esitystä dokumentista. Huonoimmassa tapauksessa käyttäjä näkee dokumentin rakenteen kuvaukseen tarkoitettua merkkauksen yhdessä dokumentin tietosisällön kanssa. Jos tyyllisivu on saatavilla, mutta siinä määritettyjä esittämissääntöjä on muutettu sen jälkeen, kun käyttäjä on viimeksi katsellut dokumenttia, näkee käyttäjä dokumentin erilaisessa visuaalisessa esitysmuodossa kuin edellisen kerran sitä katsellessaan. Tämä voi aiheuttaa hämmennystä dokumentin käyttäjälle, sillä dokumenttia ei välttämättä tunnusteta samaksi, vaikka se tietosisällöllisesti olisikin sama kuin edellisen kerran sitä katseltaessa. Vastaava ongelma voi ilmetä myös silloin, kun samaa dokumenttia katsella eri päätelaitteissa erilaisissa esitysmuodoissa.

Kolmas häiritsevä tekijä on se, että dokumentin käyttäjällä tulee olla aina tyylikielen syntaksin ymmärtävä sovellus. Mikäli sovellus ei tue tyylikieltä, ei tyylikielen mukaista esitysmääritystä voida hyödyntää. Tyylikielten ja niiden versioiden lisääntyessä rakenteisten dokumenttien esittämiseen tarkoitettujen sovellusohjelmien tulee tukea yhä useampia tyylikieliä ja niiden eri versioita.

Neljäntenä häiritsevä tekijä voidaan pitää sitä, että tyylikielten syntaksin tulkinta tapahtuu vasta päätelaitteen sovelluksessa. Koska esityksen kontrolli on viime kädessä sovellusohjelmalla, jossa dokumenttia käytetään, ei dokumentin ulkoisesta esitysmuodosta voida koskaan olla täysin varmoja.

Viides häiritsevä tekijä, joka ilmenee nimen omaan organisaatioiden välisessä rakenteisten dokumenttien välityksessä, liittyy tyyllisivujen tekijänoikeudellisiin ja taloudellisiin kysymyksiin. Laajan ja kompleksin dokumenttiarkkitehtuurin sisältävien dokumenttien tyylimääritysten tekemiseen käytetty työ määrä voidaan katsoa taloudellisesti niin arvokkaaksi, että tyylimääritykset halutaan pitää vain organisaation sisäisenä omaisuutena. Dokumentteja välitettäessä siirretään vain dokumentti, muttei tyyllisivua. Tästä on seurauksena

moninkertainen työ, sillä kukin organisaatio joutuu toteuttamaan omat tyyli-määrittymät samalle dokumenttirakenteelle. Liiketoimintaan liittyvissä ongelmatilanteissa, esimerkiksi reklamoitaessa laskusta tai saatavien perinnässä, dokumenttien käyttäjillä on tällöin erilainen näkymä dokumentin sisältämään tietoon, mikä hankaloittaa dokumenttien käsittelijöiden työtä. Toisaalta tällaiseen tilanteeseen saatetaan joutua myös tietoisesti, mikäli dokumentin vastaanottava organisaatio määrittää tyylisivun, jonka mukaan esimerkiksi kaikki yritykseen tulevat tietyn rakenteen mukaiset dokumentit visualisoidaan.

Tyyli- ja muunnoskielien käyttö tuo mukanaan monia edellisessä kohdassa mainittuja etuja. Samalla aiheutuu myös monenlaisia haittatekijöitä verrattuna sovellusalueilla aiemmin käytössä olleeseen toimintatapaan. Haittatekijät on tiedostettava ja hyväksyttävä ennen kuin tyyli- ja muunnoskielien käyttöä lähdetään soveltamaan kullakin sovellusalueella.

4.6 Muunnos- ja tyylikieliä

Edellä mainittiin DSSSL SGML-dokumenteille kehitettynä tyylikielenä. Muita rakenteisten dokumenttien tyylikieliä ovat CSS (Cascading Style Sheet Language; Lie & Bos 1996) sekä tämän tutkimuksen soveltamisen kohteena oleva XSL. Muunnoskieliä ovat muun muassa DSSSL, Omnimark (Omnimark 2000), Balise (Balise 2000) sekä XSL-kielen muunnoskieli XSLT (XSL Transformations; Clark 1999b).

Seuraavassa esitellään XML-dokumenttien esittämiseen soveltuvia muunnos- ja tyylikieliä sekä arvioidaan niiden soveltuvuutta käytettäväksi tutkimuksen tapausympäristössä. Lähtökohtana tyylikielivaihtoehtojen kartoittamiselle oli se, että niiden tulee olla joko standardeja tai ei-kaupallisten järjestöjen suosituksia. Edellä kuvatut ehdot täyttävät sekä ISO:n standardoima DSSSL että W3-järjestön tyylikielisuositukset CSS ja XSL. Näistä CSS on puhdas tyylikieli, kun taas DSSSL sekä XSLT- ja XSL-kielistä yhdessä muodostuva XSL sisältävät sekä muunnos- että tyylikielet. Toisena vaatimuksena oli se, että kieliä ymmärtävien tulkintamoduulien tulee olla valmiita itsenäisiä komponentteja, jotka ovat kutsuttavissa joko Java-ohjelmista tai ovat liitettyinä yleisimpiin HTML-selaimiin. Toinen vaatimus tiputtaa DSSSL:n pois, mutta se esitellään tässä lyhykäisesti, koska kieli on perustana sekä CSS:lle että XSL:lle.

DSSSL

DSSSL on kansainvälisen standardointijärjestön ISO:n vuonna 1996 standardoima tyylikieli, joka on kehitetty SGML-dokumenttien esitysmäärittysten tuottamiseen. Tyylikielen lisäksi DSSSL sisältää myös kysely- ja muunnoskielistandardit. (Hirvonen 1997, 26) Koska XML on SGML:n osasovellus, löytyy DSSSL:stä siten ominaisuudet myös XML-dokumentin esityksen määrittämisen kaikkiin tarpeisiin. DSSSL:n muotoiluosa on standardoitu kokoelma sääntöjä, joilla voidaan kuvata sivutetun esityksen rakenne ja ominaisuudet. DSSSL ei määritä sitä, millainen dokumentin esitysprosessin tulisi olla.

DSSSL:n etuja XML-dokumenttien esittämisen määrittämisessä ovat täysi ohjelmointiympäristö sekä mahdollisuus tuottaa RTF-esitysmuotoa (Rich Text Format; RTF 1999). Kielen käytön rajoitteena on kuitenkin hyvin vähäinen ohjelmistotuki (Pawson 2001). Koska DSSSL on tyylikielistä vanhin, on luonnollista, että sen kehityksessä tehtyä työtä on hyödynnetty muita tyylikieliä kehitettäessä. DSSSL:ssä muun muassa esiteltiin ensi kerran ajatus dokumentin esityksen muotoiluun käytettävistä muotoiluolioista (formatting objects) ja muotoiluominaisuuksista (formatting properties), joihin myös CSS- ja XSL-kielten mukainen esityksen muotoilu perustuu (Adler ym. 2001, 6).

DSSSL:n syntaksi ei ole XML-suosituksen mukainen, joten DSSSL:n käyttö edellyttää tyyliääritysten toteuttajalta uuden syntaksin opettelua. Seuraavassa on esimerkki DSSSL-syntaksista, jossa määritetään laskun tunniste -elementin (INVOICE_ID) ulkoisen esitysmuodon kirjasintyyli, pistekoko ja marginaalit.

```
(element INVOICE_ID
  (make paragraph
    font-family-name: "sans-serif"
    font-weight: normal
    font-size: 10pt
    line-spacing: 12pt
    space-before: 2pt
    space-after: 5pt))
```

DSSSL:n käyttöä tapausympäristössä rajoittaa sitä tukevien sovellusten puute sekä XML-suosituksesta poikkeava syntaksi.

CSS

CSS on alkujaan HTML-dokumenttien esittämisen muotoiluun kehitetty tyylikieli, josta on julkaistu kaksi eri versiota: taso 1 (CSS1; Lie & Bos 1996) ja taso 2 (CSS2; Bos, Lie, Lilley & Jacobs 1998). Kielestä on kehitteillä myös kolmannen tason versio CSS3. Tasoilla 2 ja 3 on otettu huomioon myös XML-dokumenttien muotoilutarpeet. CSS-kieli sisältää joukon muotoiluominaisuuksia, joita voidaan liittää rakenteisiin dokumentteihin. Muotoilumääritykset kuvataan muotoiluominaisuuksina ja niiden arvoina. CSS-kieltä tukeva selain tulkitsee dokumenttia esitettäessä CSS-kielellä määritetyt ominaisuudet ja niiden arvot ja muotoilee dokumentin esityksen niiden perusteella. HTML-selaimet tukevat laajasti CSS:n ensimmäistä tasoa, mutta tason 2 tuki on rajoittunut, vaikka version julkistamisesta on jo vuosia. CSS-kieltä käytettäessä sekä dokumentin tuottaja että käyttäjä voivat liittää dokumenttiin muotoilumäärityksiä. (Bos 2002a, Bos 2002b)

CSS:n syntaksi muistuttaa DSSSL:n syntaksia, eikä siten noudata XML-syntaksia. Seuraavassa on esimerkki CSS-määrityksestä. Esimerkissä laskun tunnistelementin (INVOICE_ID) ulkoiseen esitysmuotoon liitetään samat ominaisuudet (kirjasintyyli, pistekoko ja marginaalit) kuin edellisessä DSSSL-esimerkissä.

```
INVOICE_ID {  
    font-family: sans-serif;  
    font-weight: normal;  
    font-size: 10pt;  
    line-height: 12pt;  
    margin-top: 2pt;  
    margin-bottom: 5pt;}
```

CSS-kielen käyttöä tapausympäristössä rajoittaa se, että CSS on pelkkä muotoilukieli, jolla voidaan vain liittää erilaisia tyyliominaisuuksia esitettävän dokumentin elementteihin. CSS-kielestä puuttuu tiedon haku- ja muokkausominaisuudet, joita tarvitaan usein tietokokonaisuuksien muuntamiseksi ihmisille esitettävään muotoon. CSS-kieltä käytettäessä oletetaan, että mahdollisesti tarvittava muunnos suoritetaan ensin jollain toisella kielellä tai ohjelmalla ja CSS-kieli vain muotoilee dokumentin tietosisällön esitykseksi. Muunnoskielenä voidaan käyttää esimerkiksi XSLT-kieltä. (Lie & Bos 1998)

CSS-kielen etuna on se, että esityksen leveys- ja korkeusarvot voidaan määrittää suhteellisina, joten dokumentin käyttäjä voi muuttaa niitä dynaamisesti dokumentin käytön aikana.

CSS-kielen taso 2 soveltuu käytettäväksi tapausympäristössä silloin, kun XML-dokumentit muunnetaan HTML-kielen mukaiseen syntaksiin. CSS-kielen taso 2:n käyttöä rajoittaa kuitenkin HTML-selainten rajoittunut tuki sekä XML-suosituksesta poikkeava syntaksi. CSS-kielen taso 2:n käytössä on tämän vuoksi pitäydyttävä useimpien HTML-selainten tukemissa ominaisuuksissa. CSS-kielen käytöllä voidaan merkittävästi vähentää typografisen ulkoasun määrittämiseen liittyvää työtä, kun esimerkiksi käytettävät kirjasintyyli- ja pistekoot määritellään vain yhteen kertaan CSS-tyylisivussa. Samoin CSS-kielen taso 2:ta käyttämällä voidaan hallita tietyillä HTML-selaimilla paperille tulostettujen dokumenttien sivujen ylä- ja alatunnisteiden tulostumista.

Tutkimuksen tapausympäristössä laajimmin sovellettavaksi muunnos- ja tyylikieleksi soveltuu XSL, sillä sitä tukevia sovelluksia on laajasti ja myös makсутta saatavilla erilaisiin sovelluskehitysympäristöihin. Toinen XSL:n etu on sen sisältämän XSLT:n käyttö itsenäisiin muunnostehtäviin, jolloin tyylisivuja voidaan toteuttaa kielen sallimissa rajoissa myös esimerkiksi paperitulostuksen ja HTML-syntaksin mukaisen esittämisen määrittämiseen. Seuraavassa luvussa tutustutaan XSL-kieleen, jotta kieltä osataan soveltaa rakenteisten liiketoimintadokumenttien esittämiseen tarvittaviin muunnoksiin ja tyylimäärittäisiin.

5 XSL

Tässä luvussa perehdytään tarkemmin sekä XSL-kieleen että XSL-kielen spesifikaation määrittämään esitysprosessiin tyyllisivujen toteuttajan näkökulmasta. XSL-kielen ja esitysprosessin tunteminen on edellytyksenä sille, että tyyllisivujen toteutuksessa voidaan ottaa ylläpito näkökulma huomioon. Luku perustuu muilta kuin erikseen mainituilta viitteiltään Adlerin ym. (2001) toimittamaan W3-järjestön suositukseen.

5.1 XSL-tyylikieli

XSL on W3-järjestön lokakuussa 2001 spesifioima tyylikieli. Kieli on kehitetty nimenomaan XML-dokumenttien esittämisen määrittämiseen, mutta sitä voidaan soveltaa myös muiden kuin XML-suosituksen mukaan rakenteistettujen dokumenttien esittämiseen. XSL-suositus jakaantuu kahteen erilliseen kieleen: muunnoskieleen XSLT ja muotoilukieleen XSL. XSL-kieltä sovellettaessa kolmannen tärkeän komponentin muodostaa XPath-kieli (XML Path Language; Clark & DeRose 1999), jota käytetään pääasiassa XML-dokumentin osien osoittamiseen.

Koska XSL-lyhenteellä voidaan viitata toisaalta muunnos- ja muotoilukielen muodostamaan kokonaisuuteen tai pelkästään muunnoskieleen, käytetään XSL:n muotoilukielestä sekaannusten välttämiseksi usein lyhennettä XSLFO tai XSL-FO (Cooper, Santacruz Valencia, Donnelly & Sergeant 2000; Holman 2002). Tässä tutkielmassa käytetään jatkossa nimitystä *XSL-tyylikieli* tai lyhyemmin *XSL* tarkoitettaessa XSLT-muunnoskielen ja XSL-muotoilukielen muodostamaa kokonaisuutta. XSL-tyylikielen muunnoskielestä XSLT käytetään nimitystä *XSLT-kieli* ja muotoilukielestä *XSL-muotoilukieli*.

XSLT-kieli on tarkoitettu käytettäväksi XML-dokumentin muuntamiseen alkuperäisestä rakenteestaan poikkeavaksi XML-dokumentiksi. XSLT-kieltä voidaan käyttää myös muunnoksiin, joissa XML-dokumentin esittäminen määritetään jollakin muulla tavoin kuin XSL-muotoilukielellä. XSLT-kieli on kuitenkin rajoittunut muunnoskieli. Se ei sovellu kaikenlaisiin muunnoksiin, sillä kieli on suunniteltu käytettäväksi XML-dokumentin muuntamiseen nimen omaan XSL-muotoilukielelle.

XSLT-kielen ja XSL-muotoilukielen syntaksi on XML-suosituksen mukainen. XPath-kieltä käytetään XSLT-kielen attribuuttien arvona. XSLT- ja XSL-muotoilukieli erotetaan tyyllisivussa toisistaan käyttämällä nimiavaruuksia XML Namespaces -suosituksen (Bray, Hollander & Layman 1999) mukaisesti. Nimiavaruuksien käyttö edellyttää, että esitysprosessissa käytettävät muunnos- ja muotoilukomponentit tunnistavat nimiavaruudet. XSLT-kielen nimiavaruudesta käytetään usein lyhennettä xsl. XSL-muotoilukielen nimiavaruuden lyhenteenä on usein fo.

Seuraavassa syntaksiesimerkissä määritetään XSL-muotoilukielellä lohkoalue ja sille tulostettavan tekstin ominaisuudet vastaavasti kuin edellisen luvun DSSSL- ja CSS-esimerkeissä. Lohkoalueen sisältö valitaan esimerkin kolmanneksi viimeisellä rivillä XSLT-kielen "value-of" -määritteellä, jonka select-attribuutissa viitataan XPath-kielillä esityksen kohteena olevan XML-dokumentin INVOICE_ID-elementtiin.

```
<xsl:template match="INVOICE/HEADER">
  <fo:block
    font-family="sans-serif"
    font-wight="normal"
    font-size="10pt"
    line-height="12pt"
    margin-top="2pt"
    margin-bottom="5pt">
    <xsl:value-of select="./INVOICE_ID"/>
  </fo:block>
</xsl:template>
```

XSL-muotoilukielen kehittämisen tavoitteena on ollut kieli, joka kattaa vähintään samat muotoilumääritteet, jotka löytyvät DSSSL:stä tai CSS2:sta. Kielen kehittämisessä on pyritty hyödyntämään mahdollisimman paljon DSSSL:n ja CSS2:n kehittämisessä tehtyä työtä. XSL-muotoilukielessä esityksen rakenteen määrittävät muotoiluoliot ja -ominaisuudet pohjautuvat siten sekä DSSSL- että CSS2-kieleen. Pyrkimyksenä on kuitenkin ollut muotoilun määrittämisen yksinkertaistaminen DSSSL:ään verrattuna.

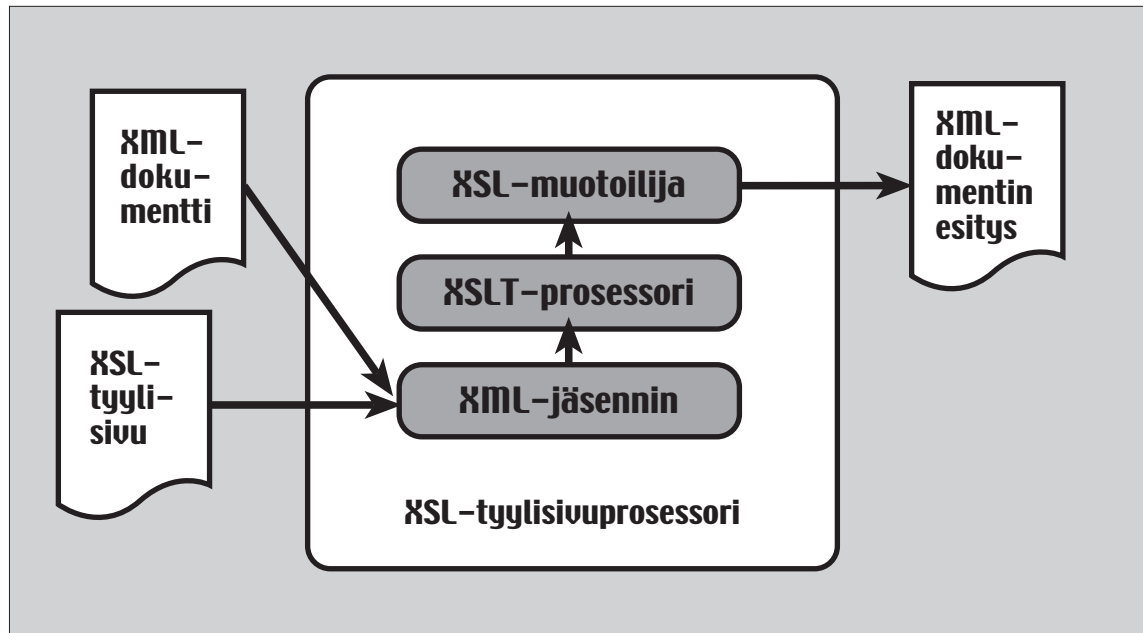
XSL-muotoilukielen muotoiluominaisuuksista päätettäessä on pyritty läheiseen yhteyteen CSS2:n kanssa. XSL-muotoilukielen muotoiluominaisuudet eivät kuitenkaan vastaa täysin CSS2:n muotoiluominaisuuksia. Osa muotoiluominaisuuksista on kyllä kopioitu CSS2:sta sellaisenaan, mutta osa ominaisuuksista eroaa CSS2:een verrattuna. Eroavaisuutta ilmenee ensinnäkin siten, että vastaavan CSS2:n muotoiluominaisuuden arvojoukkoa on laajennettu. Toinen eroavaisuus on se, että muotoiluominaisuuksia on pilkottu pienempiin osiin. Joitakin muotoiluominaisuuksia on sekä pilkottu pienempiin osiin että laajennettu niiden arvojoukkoa. XSL-muotoilukieleen on kehitetty myös täysin uusia muotoiluominaisuuksia. XSL-muotoilukielen kehityksessä on hyödynnetty DSSSL:n kehitystyössä jo kertaalleen tehtyä työtä myös siten, että XSL-muotoilukielellä kuvatun dokumentin tulkintamalli perustuu DSSSL-kieleen.

XSL-muotoilukieltä on laajennettu DSSSL:n ja CSS2:n pohjalta muun muassa kehittämällä sivutus- ja sivun asettelumallit. XSL-muotoilukieli voidaan luokitella kuuluvaksi sivunkuvauskieliin, joita ovat muun muassa PostScript (Reid 1990) ja PDF (Meehan ym. 2000). Yhteistä PostScript:ille, PDF:lle ja XSL-muotoilukielelle on se, että ne kuvaavat dokumentin esityksen semantiikan sivutetussa muodossa tavalla, joka on riippumaton sovellusohjelmasta, käyttöjärjestelmästä ja laitteistosta, jolla dokumentti on luotu tai jossa dokumenttia katsellaan tai tulostetaan.

Sivunkuvauskielten ympäristöriippumattomuus perustuu esitettävän dokumentin sivujen ja niiden sisältöjen kuvaamiseen käsitteellisinä graafisina olioina täysresoluutioisten pikselitaulukoiden sijasta. PostScriptin ja PDF:n käsitteellisenä mallina on Adoben visualisointimalli (imaging model; Meehan ym. 2000). XSL-muotoilukielessä käsitteellisen muodon *aluemallin* (area model) mukainen *aluepuu*. XSL-muotoilukieli eroaa PostScriptistä ja PDF:stä syntaksinsa ja kehittäjäriippumattomuutensa lisäksi myös siten, että dokumentin esittämiseen käytettäväksi mediaksi on ajateltu visuaalisen median lisäksi auralista mediaa, esimerkiksi puhesyntetisaattoria.

5.2 Esitysprosessi

XSL-suositus määrittää XSL-kielen lisäksi XML-dokumenttien esitysprosessin, joka koostuu toteutustasolla kuvan 10 mukaisesti useista eri komponent-



Kuva 10. XSL-tyylisivuprosessorin komponentit.

teista. Lähtökohtana esitysprosessille on XML-dokumentti, jonka tietosisältö halutaan muuntaa esitykseksi. Dokumentin ei tarvitse olla minkään dokumenttityypin määrityksen mukainen, vaan esitysprosessille riittää, että dokumentti on hyvin muodostettu eli XML-suosituksen mukainen. Toiseksi tarvitaan XSL-tyylisivu, joka kuvaa esitysprosessin lopputuloksena syntyvän esityksen ulkoisen esitysmuodon ja sisältää viittaukset niihin XML-dokumentin osiin, jotka halutaan sisällyttää esitykseen. Tyylisivu voi sisältää lisäksi myös viittauksia ulkoisiin entiteetteihin kuten kuviin tai muihin XML-dokumentteihin sekä vakiotekstiä tai visuaalisia elementtejä, joka liitetään XML-dokumentista haettavaan sisältöön. Koska XSL-tyylisivu noudattaa XML-syntaksia tulee myös sen olla hyvin muodostettu.

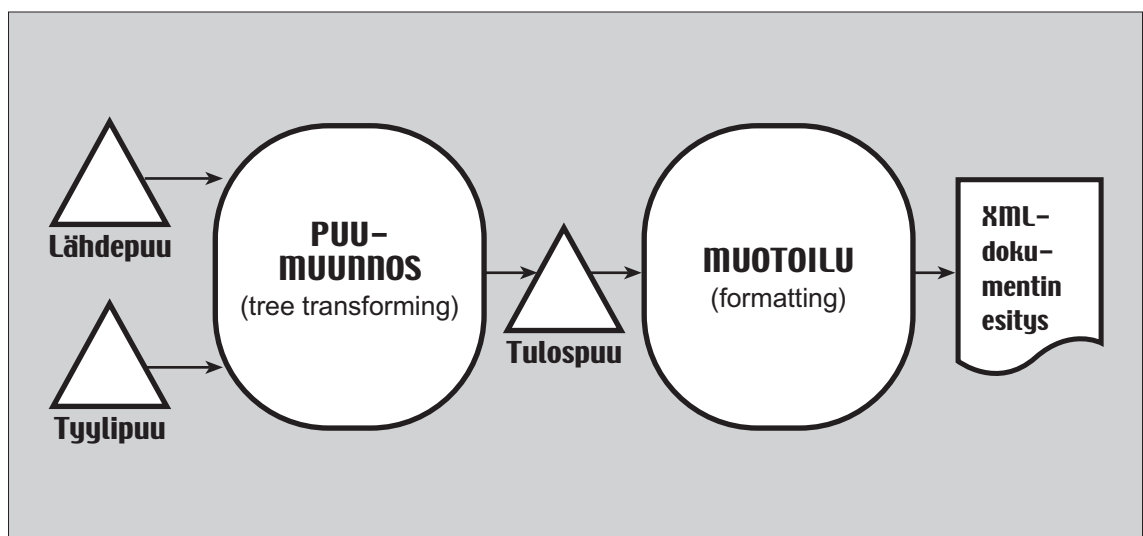
Koko esitysprosessin ydin on *XSL-tyylisivuprosessori* (XSL stylesheet processor), joka suorittaa XML-dokumentille tyylisivun mukaisen esitysmuunnoksen. Seuraavissa kappaleissa kuvattu XSL-tyylisivuprosessorin toiminnan kuvaus perustuu lähteisiin Adlerin ym. (2001, 1-5) ja Kay (2001, 45-47, 68) ellei toisin mainita.

Esitysprosessi voidaan kuvata yksinkertaisesti siten, että XSL-tyylisivuprosessori saa syötteenä XML-dokumentin sekä XSL-tyylisivun ja muodostaa tulosteeksi tyylisivussa määritetyn esityksen. Jos XSL-tyylisivuprosessoria tar-

kastellaan tarkemmalla tasolla kuin pelkkänä mustana laatikkona havaitaan, että se sisältää ensinnäkin XML-jäsentimen, jonka avulla prosessori lukee muistiin sekä XML-dokumentin että XSL-tyylisivun. Toiseksi se sisältää *XSLT-prosessorin*, joka toteuttaa muunnoksen. Kolmas tyylisivuprosessorin sisältämä komponentti on muotoiluvaiheen toteuttava *XSL-muotoilija*.

XML-dokumentti ja XSL-tyylisivu luetaan esitysprosessin aluksi XML-jäsentimen toteutuksesta riippuen joko SAX- tai DOM-rajapintaa tai niiden yhdistelmää käyttäen. Kun jonkin kieliopin mukainen dokumentti on jäsennetty muistikuvaukseksi, sitä kutsutaan *jäsennyspuuksi* (Salminen 1992, 21). Jäsennyspuu on tietorakenne, joka sisältää tarvittavan tiedon jäsennetyn dokumentin loogisesta rakenteesta. XSL-suositus nimeää kussakin esitysprosessin vaiheessa muistissa olevat jäsennyspuut eri nimillä. Muistiin luettua XML-dokumenttia kutsutaan *lähdepuuksi* ja XSL-tyylisivua *tyylipuuksi*.

XSL-tyylisivuprosessori suorittaa XSLT-prosessorin ja XSL-muotoilijan avulla XSL-suosituksen mukaisesti esitysprosessin kaksi päävaihetta: muunnosvaiheen eli puumuunnoksen (tree transforming) ja muotoilun (formatting). Puumuunnoksen lopputuloksena syntyy tulospuu, joka on muotoiluvaiheen syöte. Muotoilun lopputuloksena XML-dokumentista nähdään esitys jossain esitysovelluksessa. Päävaiheet ja niiden jakautuminen osavaiheisiin sekä vaiheiden syötet ja tulosteet on esitetty kuvassa 11.



Kuva 11. XML-dokumentin esitysprosessin päävaiheet (Adler ym. 2001, 1-5 mukailten).

5.3 Puumuunnos

XSL-suosituksen mukaan XSL-tyylisivu määrittää, kuinka tiettyyn dokumenttityypiluokkaan kuuluva yksittäinen XML-dokumentti muunnetaan puumuunnoksessa XSL-muotoilukielen muotoilusanaston mukaiseksi XML-dokumentiksi. Puumuunnoksen lopputuloksena syntyvää XML-dokumenttia kutsutaan XSL-suosituksessa tulospuuksi. *Tulospuu* kuvaa muunnoksen kohteena olevan XML-dokumentin esityksen loogisen rakenteen. Tulospuu sisältää kaiken muotoiluvaiheelle tarpeellisen informaation.

Muunnoksen määrittäminen tapahtuu siten, että tyylisivuun luodaan XSLT-kieltä käyttäen *mallipohjia* (template), jotka täytetään puumuunnoksen aikana muunnoksen kohteena olevan XML-dokumentin tietosisällöllä. Tyylisivu sisältää siten mallipohjan yhdestä tai useammasta tulospuun esitysrakenteesta. Tyylisivun toteuttaja määrittää sekä mallipohjat että sen, kuinka tulospuu muodostetaan tyylisivun sisältämien mallipohjien perusteella. Mallipohjien määrä riippuu tyylisivun toteuttajan tavasta jäsentää esityksen rakennetta. Tyylisivun toteuttajan vastuulla on se, että muunnosprosessin lopputuloksena syntyvä tulospuu noudattaa XSL-muotoilukielen kielioppia. Tyylisivuja toteutettaessa on huomioitava, että mallipohjat sidotaan yhteen tulospuun jäsentämisjärjestyksen perusteella. Jotta haluttu lopputulos saadaan aikaan, on tyylisivua luontevaa toteuttaa XSL-muotoilukielen jäsentämisjärjestyksen mukaisesti.

Puumuunnoksessa XSLT-proessori lukee lähdepuun solmuja ja etsii tyyli-
puusta mallipohjia, jotka täsmäävät (match) prosessoinnin kohteena olevaan solmuun. Täsmäys tapahtuu XPath-kielillä ilmaistujen hahmojen (pattern) perusteella. Hahmo muodostuu yhdestä tai useammasta ehdosta. Jos prosessoinnin kohteena oleva solmu täyttää ehdon, se täsmää hahmoon ja mallipohjan suoritus alkaa. Kohdassa 5.1 olleessa syntaksiesimerkissä on yksi mallipohja, joka täsmää XML-dokumentin puurakenteessa HEADER-elementtiin, joka on INVOICE-elementin välittömänä lapsielementtinä.

Kussakin mallipohjassa määritetään lähdepuun jonkin rakenneosan tietosisällön esitystapa halutun esityskielen syntaksilla. Tässä tutkielmassa tyyli-
sivuun sisällytettävää esityskieltä kutsutaan *kohdekieleksi*. Vaikka XSL-suosituksen mukaan kohdekielenä on aina XSL-muotoilukieli, voidaan kohdekielenä

käyttää myös jotakin muuta esityksen kuvauskieltä. Kohdekieli voi olla esimerkiksi HTML tai XHTML (eXtensible Hypertext Markup Language; Pemberton, Austin, Axelsson ym. 2000). Kohdekieli riippuu aina siitä, millaisessa käyttöympäristössä ja millaisella sovellusohjelmalla XML-dokumentti halutaan esittää. Jos kohdekielenä on XSL-muotoilukieli, liittyy tyylisivun toteuttaja mallipohjiin XSLT-kielen muunnosmäärittelyjen lomaan XSL-muotoilukielellä kuvatun tyylimäärittelyksen.

Muunnoksen kohteena olevan XML-dokumentin tietosisältö liitetään mallipohjiin *ilmaisuuilla* (expressions). Ilmaisuuilla osoitetaan XML-dokumentin rakenteessa sijaitsevaan tietosisältöön. XSL:ssä ilmaisukielenä käytetään XPath-kieltä. XPath kielellä on mahdollista osoittaa mihin tahansa XML-dokumentin rakenteeseen riippumatta siitä, mitä rakenneosaa muunnoksessa ollaan prosessoimassa.

Jos kohdekielenä käytetään XSL-muotoilukieltä, sisältää puumuunnoksen lopputuloksena syntyvä tulospuu muotoiluolioilla ja niihin liitetyillä muotoiluominaisuuksilla kuvatun muotoilusemantiikan. Tulospuu koostuu muotoiluolioita vastaavista elementtisolmuista ja muotoiluominaisuuksia vastaavista attribuuttisolmuista sekä niiden arvoista. Tulospuu voidaan sarjallistaa ja kirjoittaa tiedostoon dokumentiksi, jonka syntaksi on XSL-muotoilukielen syntaksia. Tässä tutkielmassa tiedostoon kirjoitettua tulospuuta, joka sisältää ainoastaan XSL-muotoilukielen syntaksia, kutsutaan *XSL-dokumentiksi*. Tiedostoon kirjoitettu XSL-dokumentti voi näyttää esimerkiksi tältä:

```
<?xml version="1.0"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="sisasivu"
      page-width="50mm" page-height="50mm">
      <fo:region-body region-name="mallisivunRunko"/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="sisasivu">
    <fo:flow flow-name="mallisivunRunko">
      <fo:block font-family="sans-serif" font-size="8pt">
        Esimerkkidokumentti, jossa yksi lohko
```

```

        </fo:block>
    </fo:flow>
</fo:page-sequence>
</fo:root>

```

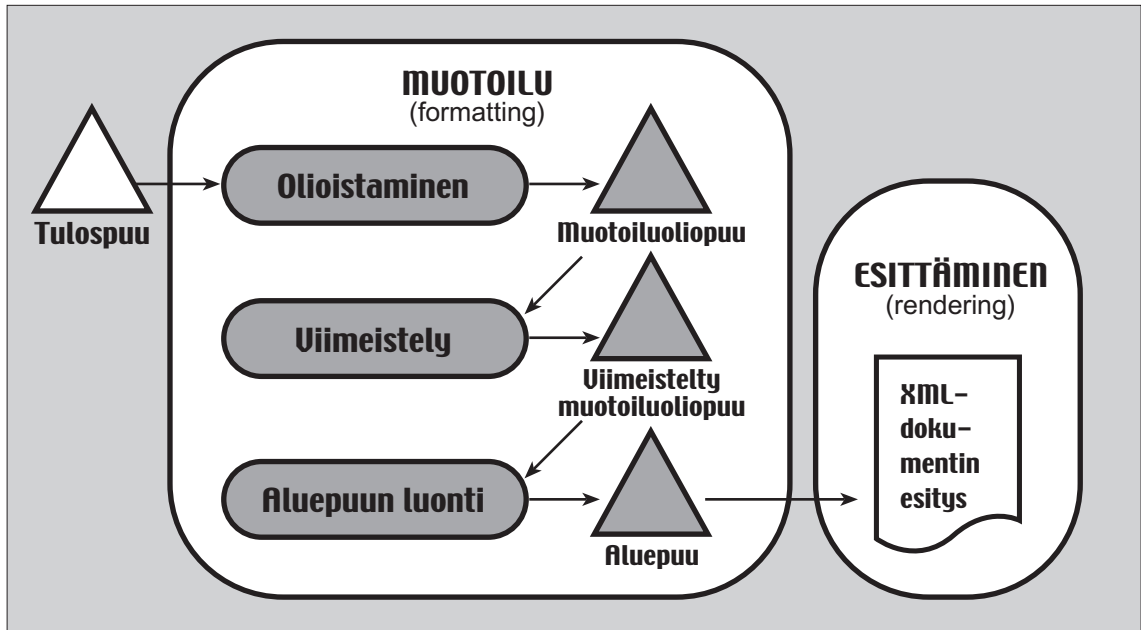
5.4 Muotoilu ja esittäminen

Muotoiluvaiheessa *XSL-muotoilija* (formatter) tulkitsee ja muotoilee (formatting) tulospuun ja esittää (render) sen käytettävälle esitysovellukselle ominaisella tavalla. Esittäminen tapahtuu *esitysovelluksessa* joko näytölle tai paperille tulostettuna tai puhesyntetisaattorin tulkitsemana. Muotoiluvaiheen lopuksi XML-dokumentti siis esitetään käyttäjälle jossain ulkoisessa esitysmuodossa. Tutkimuksen aikaisten XSL-muotoilijoiden ominaisuuksien perusteella esitysovellus voi olla esimerkiksi XML-selain (X-Smiles 2002, FOA 2002), PDF-selain (Adobe Acrobat Reader 5 2002) tai RTF-muotoa ymmärtävä sovellus. Esitysovellus voi olla myös paperi, mikäli tulostusprosessi tukee XSL-muotoilijan tuottamaa sivunkuvauskieltä.

XSL-suosituksen mukaan muotoiluvaiheen tehtävänä on muodostaa aluemallinen mukainen, geometrisia alueita sisältävä aluepuu. Muotoiluvaihe jakaantuu edelleen kolmeen eri osavaiheeseen: olioistamiseen (objectify), viimeistelyyn (refinement) ja muotoiluvaiheen lopputuloksen, aluepuun, muodostamiseen (area tree generating). Jokaisessa vaiheessa tuloksena on edellisestä vaiheesta eroava puu. Kuva 12 havainnollistaa muotoiluvaihetta, sen osavaiheita sekä muotoiluvaiheen lopputuloksena syntyvää esitystä.

Muotoilun ensimmäisessä osavaiheessa, *olioistamisessa*, tulospuun (elementti- ja attribuuttipuu) elementit muunnetaan muotoiluoliosolmuiksi. Attribuutit muunnetaan muotoiluoliosolmujen ominaisuuksiksi. Olioistamisessa myös jokaisesta tulospuun yksittäisestä merkistä muodostetaan oma solmunsa sekä luodaan yhdistelmäominaisuudet. Tuloksena syntyy *muotoiluoliopuu* (formatting object tree).

Muotoilun toisessa vaiheessa, *viimeistelyssä* (refinement), muotoiluoliopuu muunnetaan *viimeistellyksi muotoiluoliopuuksi* (refined formatting object tree). Viimeistelyssä ominaisuudet muunnetaan *piirteiksi* (traits), jotka määrittävät rajoitteita muotoilun lopputulokseen eli esitykseen. Monet piirteet vastaavat



Kuva 12. XML-dokumentin muotoiluvaihe ja esittäminen (Adler ym. 2001, 1-5 mukailten).

yksi yhteen tiettyä muotoiluominaisuutta, mutta osa piirteistä vaatii monimutkaisempaa käsittelyä. Viimeistely sisältää seuraavia toimintoja: ensinnäkin kaikki ominaisuudet muunnetaan yksittäisiksi ominaisuuksiksi, toiseksi suhteelliset arvot muunnetaan absoluuttisiksi, kolmanneksi ratkaistaan tyhjämerkkien ja rivinvaihtojen käsittely, neljänneksi toteutetaan ominaisuuksien perintä sekä viidenneksi poistetaan päällekkäiset ominaisuusmääritykset.

Muotoilun kolmannessa vaiheessa *muodostetaan aluepuu* kuhunkin muotoiluoliioon liitetyn muotoilusemantiikan perusteella. Jokaiseen muotoiluolioluokkaan hyväksyttävät piirteet kontrolloivat alueiden muodostamista. Vaikka kaikki yli 200 muotoiluominaisuutta voidaankin määrittää jokaiselle muotoiluoliolle ja jokaiselle muotoiluolioluokalle, niin vain osaa muotoiluominaisuuksista voidaan käyttää tietyn luokan olioiden piirteiden määrittämiseen. Aluepuussa geometriset alueet kuvataan ylimmällä hierarkiatasolla sivuina, jotka muodostavat yhden tai useamman sivun mittaisia jaksoja. Yksittäinen sivu sisältää geometrisia alueita, joilla kullakin on oma paikkansa sivulla sekä tieto siitä, mitä tietoa tällä alueella esitetään. Alueella voi olla myös tausta, sisennykset tai reunukset. Alueet voivat olla rinnakkaisia tai sisäkkäisiä. Esimerkiksi sivulla voi olla kaksi rinnakkaista palstaa, jotka muodostuvat riveistä, joilla on kirjaimia. Kirjain on siis rivialueen sisällä, joka on osa palstaa, joka puolestaan esitetään jollain sivulla.

Esittämisvaiheessa (render) esityksen abstrakti kuvaus eli geometrisista alueista koostuvia, sivuja sisältävä aluepuu muunnetaan esitykseksi käytettävälle esitysovellukselle ominaisella tavalla. On huomioitava, että eri esitysovellusten tuottamien esitysten välillä on eroavaisuuksia. Esitysovellus saa syötteenä suosituksen mukaisesti aluepuun, joka kuvaa sivuja sekä tekstin ja graafisten objektien sijoittelujen näille sivuille. Esitysovelluksen vastuulla on aluepuun tulkinta esitykseksi ja esityksen muodostus esitysovelluksen tukemaan muotoon. Esitysovellukset toimivat muun muassa kirjasinten muodostamisen osalta eri tavoin, minkä vuoksi eri esitysovellukset saattavat tuottaa eri kokoisia kirjasimia, vaikka kirjasinten koko olisikin määritetty kiinteäksi. Tämän vuoksi dokumentista samalla tyyllisivulla muunnettu esitys voi näyttää varsin erilaiselta eri esitysovelluksissa katseltuna. (FOP 2002)

Vaikka XSL-suositus kuvaa käsitteellisesti yhtenäisen XSL-tyylisivuprosessori-ohjelmiston, voivat toteutuksissa XSLT-prosessori ja XSL-muotoilija olla täysin erillisiä ohjelmia, jotka voivat myös sijaita fyysisesti eri paikoissa. Tällöin tulospuu voidaan sarjallistaa ja kirjoittaa tiedostoon XSL-dokumentiksi. Tiedostoon kirjoitettu XSL-dokumentti voidaan välittää myöhemmin XSL-muotoilijalle esittämistä varten. Puumuunnos voidaan esimerkiksi tehdä palvelimelle sijoitetulla XSLT-prosessorilla ja XSL-muotoilija voidaan liittää selaimen osaksi. Tällöin selaimen osana on oltava myös jäsenin, jotta XML-syntaksin mukaisen XSL-dokumentin luku on mahdollista. XSL-suositus sallii tulospuun tuottamisen myös muulla tavoin kuin puumuunnoksessa. Tulospuu voidaan tuottaa esimerkiksi käsin tekstieditorilla, mutta se on erittäin työläs vaihtoehto.

Esitysprosessin kokonaisuuden kannalta oleellista toteutusratkaisuissa on ainostaan se, että prosessin eri osien väliset rajapinnat ovat keskenään yhteensopivia. Toteutuksissa pitää varmistua siitä, että esittämisen toteuttavalle ohjelmistolle välitetään vain sellaisessa esitysmuodossa olevia dokumentteja, joiden muotoiluun ja tulkintaan esitysohjelmisto pystyy. Esityksen epäonnistumisen varalta välitettävästä dokumentista tulisi löytyä viittaus alkuperäiseen XML-dokumenttiin.

Tutkimuksen tekohetkellä lokakuussa 2002 XSL-muotoilukieltä ymmärtävillä hieman yli kymmenellä sovelluksella on mahdollista katsella XSL-dokumentteja näytöllä XML-selaimessa, muuntaa ne johonkin toiseen esitysmuotoon,

kuten PDF- RTF-, TeX- muotoon (TeX Users Group 2002) tai tulostaa muotoiltu dokumentti suoraan paperille PCL-, PDF- tai PostScript-kieltä käyttäen. Myös muuntaminen joihinkin julkaisuohjelmiin, kuten QuarkXPressiin ja FrameMakeriin on mahdollista. Suurin osa toteutuksista on avoimen lähdekoodin sovelluksia. (XMLSoftware 2002)

5.5 XSL-muotoilukielen ominaisuuksia

XSL-muotoilukielessä dokumentin esityksen käsitelmallin muodostava aluepuu kuvataan puun solmuja vastaavilla muotoiluolioilla ja niihin liitettyillä muotoiluominaisuuksilla. Tulospuun sisältämät *muotoiluoliot* määrittävät yhdessä järjestyksen, jossa esitettävä tietosisältö sijoitetaan esityksen aikana sivulle. Muotoilun yksityiskohdat, kuten mitat ja kirjasimet, määritellään *muotoiluominaisuuksilla*. (Harold 1999) XSL-tyylisivussa XSL-muotoilukielen elementit vastaavat muotoiluolioita ja attribuutit muotoiluominaisuuksia. Elementtien sisällöt vastaavat muotoiluolioiden sisältöjä ja attribuuttien arvot muotoiluominaisuuksien arvoja. Seuraavassa tarkastellaan XSL-muotoilukielen ominaisuuksista muotoiluolioita, muotoiluominaisuuksia sekä esityksen sivutusta.

MUOTOILUOLIOT

XSL-muotoilukielen ensimmäinen suositusversio 1.0 sisältää 56 erilaista muotoiluoliota. Muotoiluoliot jakaantuvat esitysprosessin aikaisen aluepuun luonnin näkökulmasta kolmeen eri tyyppiin. Ensimmäisen tyyppin muodostavat oliot, jotka muodostavat yhden tai useamman alueen. Toinen oliotyyppi koostuu olioista, jotka palauttavat joko yhden tai useamman alueen, mutta eivät muodosta niitä. Ensimmäisen ja toisen tyyppin olioita kutsutaan vuoto-olioiksi, sillä ne luovat esitykseen muodostuvien sivujen kokonaisvuon esitykseen liitettävän tietosisällön määrän ja esitystavan perusteella. Kolmas oliotyyppi muodostuu alueiden muodostamisessa käytettävistä asettelu- ja apu-olioista.

Tyylisivujen toteuttajan kannalta on kuitenkin selkeintä tarkastella muotoiluolioita sivunkuvauksen näkökulmasta. Tällöin tarkastelu kohdistuu dokumenttikokonaisuuden, yleisten esitysasetusten sekä yksittäisen sivun yleisten muotoiluasetusten määrittämisessä käytettäviin muotoiluoloihin. Tämän li-

säksi tulee tarkastella sivukokonaisuuden sisällä tarvittavia olioita. Näitä ovat lohko-, taulukko- ja luettelo-oliot sekä niiden sisäiset rivioliot.

Jokaiseen XML-dokumentin esitykseen on määritettävä dokumenttikokonaisuuden, dokumentin sivujen yleisen asettelun sekä sivujen jaksotuksen kuvaavat oliot. Näillä määritetään dokumentin sivumääritykset kullekin erilaiselle sivutyypille sekä erilaisten sivutyyppeiden jaksotus. Eri sivutyyppejä voivat olla esimerkiksi kansisivu, oikean ja vasemman puoleinen sisäsivu sekä takasivu. Jaksotuksella tarkoitetaan eri sivutyyppeiden keskinäistä esiintymisjärjestystä. Kullekin sivutyypille määritetään yleiset asetellut, kuten ylä-, ala- ja sivumarginaalien sekä varsinaisen tekstin sisältävän sivun runko-osan mitat.

XSL-kielen soveltajan näkökulmasta toisen tärkeän muotoiluolioluokan muodostavat sivuille sijoitettavat lohkotason oliot eli yksittäiset lohkot sekä lohkosäiliöt. Lohkot ovat esimerkiksi otsikoita, kappaleita ja alaotsikoita. Lohkotason olioiksi ymmärretään myös taulukot sekä luetteloriveistä muodostuvat luettelolohkot. Taulukoiden ja luettelolohkojen määrittäminen muotoiluolioita käyttäen vastaa HTML-kielen taulukoiden ja luetteloiden määrittämistä. Sekä HTML- että XSL-muotoilukielessä taulukolle voidaan määrittää otsikko, rivi ja solu, kun taas luettelolohko koostuu luettelorivistä, luettelomerkistä ja luettelotekstistä. Vain syntaksi kielten välillä on hieman erilainen. Lohkosäiliöt ovat useampia lohkotason olioita sisältäviä ja nämä lohkotason oliot kokonaisuudeksi kokoavia säiliöitä.

Kolmas tyylisivun toteuttajan näkökulmasta tärkeä muotoiluolioluokka on lohkotyyppisten olioiden sisälle, yksittäisille riveille määritettävät oliot. Näitä ovat esimerkiksi riveille tulostuvat merkit, sivunumerot, linkit, ulkoiset kuvat sekä muut ulkoiset viittaukset.

MUOTOILUOMINAISUUDET

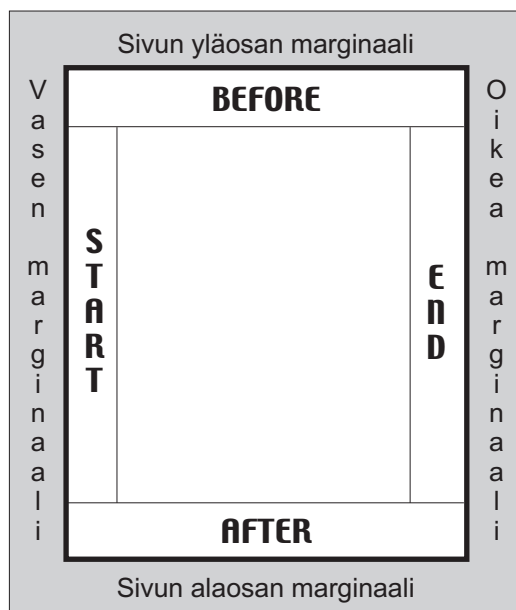
Muotoiluominaisuuksia XSL-muotoilukielessä on yli 200. Ne ovat hyvin eri tyyppisiä ja soveltuvat eri muotoiluolioiden ominaisuuksien määrittämiseen. Muotoiluominaisuuksia on muun muassa kehysten, lohkojen täytteen, taustan, kirjasinten, tavutuksen, marginaalien ja rivityksen määrittämiseen. Muotoiluominaisuuksilla voidaan myös määrittää eri tyyppisten alueiden koot,

yksittäisten merkkien ominaisuudet sekä lohkotason olioiden absoluuttinen tai suhteellinen sijainti. Osa muotoiluominaisuuksista on tarkoitettu typograafisten asettelujen määrittämiseen, kuten esimerkiksi leski- ja orporivien kontrollointiin. Oman muotoiluominaisuusluokkansa muodostavat auraliseen esittämiseen soveltuvat ominaisuudet. Joskin suurin osa muotoiluominaisuuksista on määritetty pelkästään visuaaliselle medialle.

SIVUTUS JA SIVUASETTELUT

Monisivuisten dokumenttien sivutus tapahtuu sivun koon sekä sivulle tulostettavan tietosisällön ja sen asettelun perusteella. Sivujen kokonaislukumäärä muodostuu siten esitysprosessin aikana esitettävän tietosisällön laajuuden ja sivuasettelujen perusteella. Sivutusta voidaan hallita sivujaksojen avulla siten, että esimerkiksi jokainen luku muodostaa oman sivujaksonsa. Koska jokainen uusi sivujakso aloittaa aina uuden sivun, alkaa jokainen luku siten omalta sivultaan.

XSL-tyylikielen ominaisuuksista voidaan lopuksi mainita sivuasettelujen kansainvälisyys. XSL-tyylikielessä sivuasettelu kuvataan abstraktisti siten, että myös muu kuin länsimainen tekstin kulkusuunnan asetteleminen on mahdollista. Teksti voidaan tämän ansiosta tuottaa sivulle luettavaksi esimerkiksi oi-



kealta vasemmalle ja alhaalta ylös. Sivumarginaalit määritetään käsittein "ylä", "ala", "oikea" ja "vasen", mutta sivun runko-osan sisällä operoidaan käsitteillä "alku", "loppu", "ennen" ja "jälkeen". Kuvassa 13 on esitetty XSL-muotoilukielen kansainvälistä sivun asettelumallia havainnollistava asettelumalli suomenkielisen dokumentin visuaalisen esityksen tuottamisen näkökulmasta. Kuvan mukaisessa asettelussa teksti juoksee länsimaisen lukutavan mukaisesti vasemmalta oikealle ja ylhäältä alas.

Kuva 13. XSL-muotoilukielen sivun asettelumalli (Adler ym. 2001)

5.6 XSL-muotoilukielen käyttö

Tyylisivun toteuttajan näkökulmasta esityksen määrittäminen XSL-muotoilukielillä sisältää kolme erilaista tehtävää. Ensimmäiseksi määritetään esitykseen tulevat erilaiset sivutyypit ja niiden yleinen sivuasettelu. Tämän jälkeen määritetään kullekin erilaiselle sivutyypille sivun sisäisen rakenteen muodostavat graafiset lohko-oliot kuten lohkot, luettelot tai taulukot. Tarvittaessa edellisten sisälle määritetään vielä rivin sisäiset osat. Lopuksi jokaiselle graafiselle oliolle määritetään halutut muotoiluominaisuudet kuten esimerkiksi rivinväli, marginaalit, tekstin sovitus, kirjasimet ja muut typografiset ominaisuudet. XSL-muotoilukielen määrittämisen lisäksi tyylisivun toteutus sisältää erillisenä tehtävänä myös muunnoksen määrittämisen.

Kuvan 14 yksinkertaisessa esimerkissä valotetaan käytännön tasolla XSL-muotoilukielillä tapahtuvaa dokumentin esityksen määrittämistä. XSL-muotoilukielen semantiikan ymmärtämiseksi esimerkki sisältää puumuunnoksen lopputuloksesta, tulospuusta, sarjallistetun XSL-dokumentin. Muunnoksen kohteena ollut XML-dokumentti on tutkielman liitteenä 1 ja muunnosmäärittämisen sisältävä XSL-tyylisivu liitteenä 2.

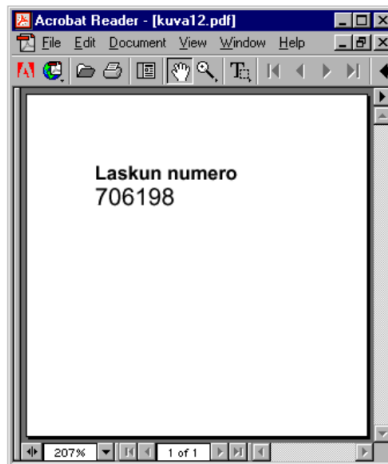
Esimerkistä havaitaan, että XSL-dokumentin juuren muodostaa `fo:root`-elementti. Eri sivutyypit ja niiden yleiset ominaisuudet kuten marginaalit ja sivutyypin nimi määritellään `fo:layout-master-set`-elementissä. Esimerkissä sivulle tuotetaan sisältöä vain runko-osaan (`body`), jota kuvaa `fo:region-body`-elementti.

Sivujakso alkaa `fo:page-sequence-element`illä, jossa viitataan `master-reference`-attribuutilla sivujaksoon sovellettavaan sivutyyppiin. Viittauksen kohteena tulee olla jokin `fo:layout-master-set`-elementin lapsielementtinä määritelty `fo:simple-page-master-element`. Sivujakson lapsielementtinä on `fo:flow-element`, joka kokoaa jakson sisäisen lohkovuon. Tässä tapauksessa lohkovuo muodostuu kahdesta yksinkertaisesta `fo:block-element`illä määritellystä lohkosta. Laskun numeron otsikointi ja laskun numero (`INVOICE_ID`) esitetään kumpikin omilla lohkoillaan. Lohkoihin sijoitettujen tekstien ominaisuudet kuvataan lohkon muotoiluominaisuuksilla. Tässä tapauksessa ominaisuudet ovat kirjasinmäärittämiä (kirjasimen tyyliperhe sekä kirjasinkoko ja -leikkaus). Ensimmäisessä lohkossa on lisäksi marginaalin määrittely lohkoa edeltävän muotoiluolion ja lohkon välille.

```

<?xml version="1.0"?>
<fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">
  <fo:layout-master-set>
    <fo:simple-page-master master-name="sisasivu"
      page-width="50mm" page-height="50mm"
      margin-left="10mm" margin-right="10mm"
      margin-top="5mm" margin-bottom="10mm">
      <fo:region-body region-name="mallisivunRunko"/>
    </fo:simple-page-master>
  </fo:layout-master-set>
  <fo:page-sequence master-reference="sisasivu">
    <fo:flow flow-name="mallisivunRunko">
      <fo:block space-before="5mm"
        font-family="sans-serif" font-size="8pt"
        font-weight="bold">Laskun numero</fo:block>
      <fo:block font-family="sans-serif" font-size="10pt"
        font-weight="normal">706198</fo:block>
    </fo:flow>
  </fo:page-sequence>
</fo:root>

```



Kuva 14. Esimerkki sarjallistetusta tulospuusta (XSL-dokumentista) sekä koodia vastaava esitys PDF-dokumenttina avattuna Acrobat Reader -ohjelmaan. Muunnos PDF-muotoon on toteutettu Apachen XSL-tyylisivuprosessorilla FOP.

Koodin alapuolella näkyy XSL-dokumentin esitys PDF-muodossa Adoben Acrobat Reader -ohjelmassa. Esimerkin XSL-dokumentti on muunnettu PDF-muotoon Apachen XSL-tyylisivuprosessorilla FOP. Liitteen 2 XSL-tyylisivu eroaa kuvan 14 XSL-dokumentin koodista siten, että se sisältää myös XSLT-kielisen muunnoksen määrittelyn sekä tietosisällön haun muunnoksen kohteena olevasta XML-dokumentista.

Tämän tutkimuksen tekohetkellä lokakuussa 2002 XSL-spesifikaation julkistamisesta on kulunut vuosi. Vaikka XSL-tyylikielen spesifikaatio on laaja, on kieltä ehditty soveltamaan jo useiden esitysovellusten rajapintoihin (XML-Software 2002). Onkin esitetty tulevaisuuden visioita, joissa XSL-tyylikieli nähdään yleisenä XML-dokumenttien esityksen kuvauskielenä niin sovellusohjelmissa (WWW-selaimet, tekstinkäsittelyohjelmat ym.) kuin myös laitteistoissa (tulostimet, matkapuhelimet ym.) (Holman 1999; Neugebauer 2002). Esitettyjen visioiden käyttökelpoisuutta arvioitaessa on kuitenkin pidettävä mielessä, että vaikka XSL-tyylikieltä käytettäisiinkin jonain päivänä universaalina XML-dokumenttien esityksen kuvauskielenä, tulee esitys Lyytikäisen ja Tiitisen suunnitteluohjeiden (2001, 26-27) mukaan toteuttaa tietosisältömääränsä ja visuaalisen ulkoasunsa suhteen aina kunkin päätelaitteelle ominaisuuksien ehdoilla. Mahdollinen visioiden toteutuma ei siten poista tarvetta tuottaa eri käyttötilanteisiin erilaisia visuaalisia esityksiä, jotka perustuvat useisiin erilaisiin tyylisivumäärittelyihin. Seuraavassa luvussa syvennyttään tyylisivumäärittelysten hallintaan, jotta erilaisten esitysvariaatioiden kasvaessa ei päädyttäisi hankalasti ylläpidettäviin ratkaisuihin.

6 TYYLISIVUJEN TOTEUTUSMALLI

Tässä luvussa esitellään tämän tutkimuksen keskeinen anti: XML-dokumenttien esittämisen määrittämiseen tarkoitettujen tyyllisivujen toteutusmalli. Mallin muodostaminen perustuu esimerkkitapauksesta tehtyihin havaintoihin. Vaikka esimerkkitapauksen sovellusalue, rakenteiset organisaatioiden väliset liiketoimintadokumentit, tuo erityispiirteitä malliin, on toteutusmalli sovellettavissa myös muille rakenteisten dokumenttien sovellusalueille. Koska toteutusmallin painotus on tyyllisivuissa ja niihin kohdistuvassa ylläpityössä, jää visualisointiprosessiin tarvittavien muiden komponenttien yksityiskohtainen arviointi ja valinta mallin ulkopuolelle.

Toteutusmalli jakautuu kolmeen erilliseen osaan: visualisointiympäristön kartoitukseen, muutostekijöiden arviointiin sekä tyyllisivujen vaatimusmäärittämiseen. Kutakin mallin vaihetta, vaiheeseen liittyviä tehtäviä sekä kokemuksia ja ratkaisuja esimerkkitoteutuksesta käsitellään tässä luvussa omissa kohdissaan. Luvun lopuksi arvioidaan tutkimuksessa tuotettua esimerkkitoteutusta kokonaisuutena.

6.1 Visualisointiympäristön kartoitus

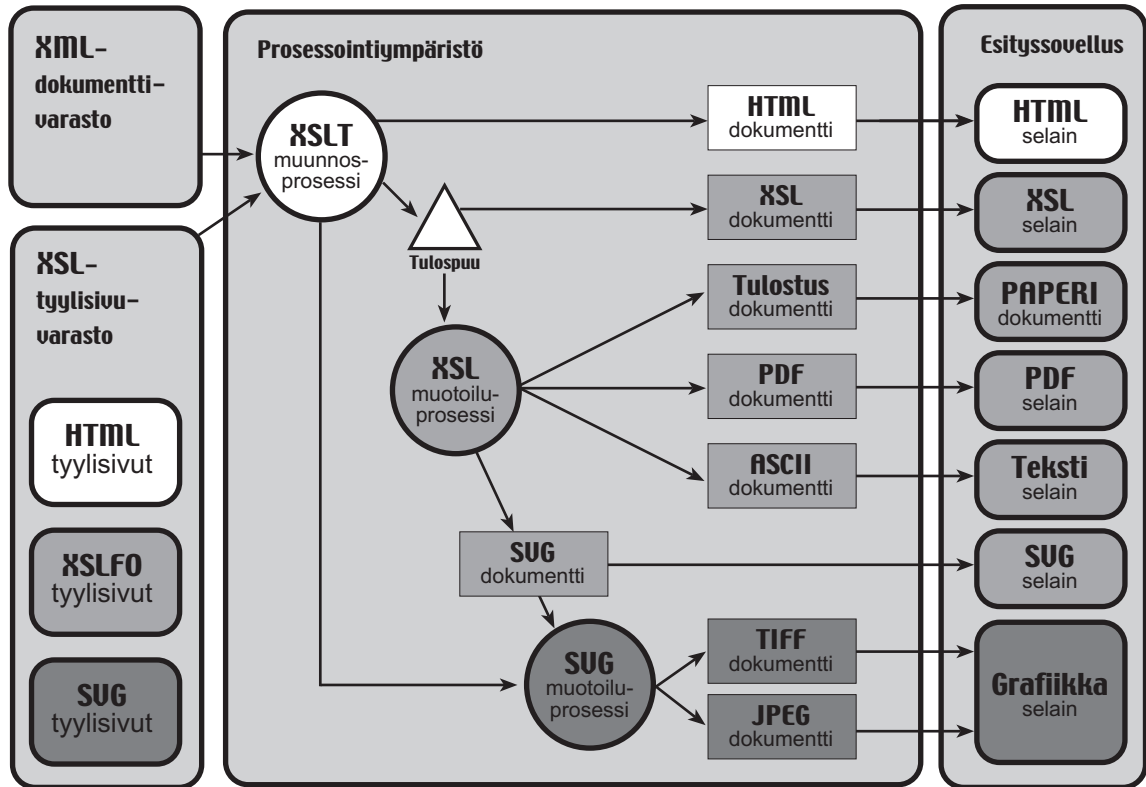
Tyyllisivujen toteutusmallin ensimmäinen osa muodostuu visualisointiympäristön kartoituksesta. *Visualisointiympäristöllä* tarkoitetaan sitä teknistä ja organisationaalista kokonaisuutta, johon kuuluu XML-dokumenttien ja tyyllisivujen lisäksi sekä visualisointia varten tarvittava muunnosympäristö että päätelaitteiden ja esitysovellusten muodostama esitysympäristö. *Esitysovelluksella* tarkoitetaan sovellusta, joka ymmärtää dokumentin muunnoksen ja muotoilun seurauksena syntyneen dokumenttimuodon syntaksia.

Visualisointiympäristö ja sen ominaisuudet vaikuttavat käytettävän tyyllikielen valinnan lisäksi suuresti myös yksittäisten tyyllisivujen toteuttamiseen. Mikäli visualisointiympäristössä tullaan tukemaan yhden XML-dokumentin visualisointia useaksi erilaiseksi esitykseksi, tarkoittaa tämä lähtökohtaisesti sitä, että jokaista toisestaan poikkeavaa esitystä vastaa viime kädessä aina oma tyyllisivu. Tällaisessa ympäristössä tulee jo visualisointiympäristön suunnitteluvaiheessa huomioida tyyllisivujen varastointi- ja ylläpidettävyyssnäkökohdat.

Tyylisivujen määrä kasvaa tuettavien esitysovellusten määrän kasvaessa. Mikäli XML-dokumentti halutaan muuntaa esimerkiksi kolmelle erilaiselle esitysovellukselle kuten HTML-selaimelle, tulostusformaattiin paperitulostusta varten ja PDF-selaimelle, on kullekin muunnosprosessin kohdekielille toteutettava oma tyylisivunsa. Jos tämän lisäksi halutaan tukea erilaisia ulkoasuja, on tyylisivujen kokonaismäärä pahimmassa tapauksessa erilaisten ulkoasujen määrän ja erilaisten esitysmuotojen kohdekielien määrän tulo. Jos esimerkiksi kohdekieliä on kolme ja erilaisia ulkoasuja viisi, on lopullinen tyylisivujen määrä 3×5 eli 15. Erilaisten tyylisivujen tarve pienenee, jos jossakin kohdekielessä rajoitetaan erilaisten ulkoasujen määrä esimerkiksi yhteen, vaikka muissa kohdekielissä sallitaan ulkoasujen variaatiot. Samoin XSL-muotoilijan monipuolinen esitysovellusten tuki pienentää tarvittavien tyylisivujen määrää. Ylläpidon kannalta olisi suositeltavaa pitäytyä mahdollisimman pienessä erilaisten ulkoasujen määrässä tai vaihtoehtoisesti suunnitella jo visualisointiympäristöä mietittäessä tyylisivujen ylläpito ja uudelleenkäytön mahdollisuudet huolella.

Tyylisivujen ylläpidon ongelmaa selventää kuva 15. Siinä havainnollistetaan yhtä mahdollista visualisointiympäristöä sekä ympäristössä tapahtuvaa visualisointiprosessia. Kuvassa vasemmalla näkyvät visualisointiprosessiin tarvittavat syötedokumentit: XML-dokumentit ja XSL-tyylisivut. Prosessointiympäristö muodostuu XSL-spesifikaation määrittämistä XSLT-muunnosprossessorista ja XSL-muotoilijasta sekä prosessien tulosteista. Tarpeiden mukaan prosessointiympäristöön voi kuulua myös muita muotoilijoita, kuten esimerkiksi SVG-muotoilija (Scalable Vector Graphic; Bowler, Brown, Capsimails & Cole 2001). Kuvassa oikealla näkyy erilaisia esitysovelluksia, jotka tulkitsevat muunnos- tai muotoiluprosessin tulosten ja esittävät visualisointiprosessin lopputuloksena dokumentin käyttäjälle. Tilannetta on kuvassa yksinkertaistettu siten, että dokumenttien on ajateltu olevan vain yhtä dokumenttityyppiä.

Visualisointiprosessin syötteet varastoidaan kahteen käsitteellisesti erilliseen varastoon: XML-dokumenttivarastoon ja XSL-tyylisivuvarastoon. Tyylisivut voidaan jakaa vielä edelleen erilaisiin varastoluokkiin tyylisivujen sisältämän kohdekielen perusteella. Siten tyylisivuvarastossa voi olla esimerkiksi HTML-tyylisivuja, XSL-tyylisivuja ja SVG-tyylisivuja. Tyylisivuluokkien lukumäärä on riippuvainen siitä, miten montaa erilaista esitysovellusta visuaali-



Kuva 15. Esimerkki mahdollisesta visualisointiympäristöstä, jossa yhden dokumenttityypin mukainen XML-dokumentti prosessoidaan esitettäväksi eri tavoin eri käyttötilanteissa.

sointiympäristössä halutaan tukea sekä muunnos- ja muotoiluprosessiin käytettävien komponenttien kyvystä tuottaa tulosteenaan eri dokumenttimuotoja. Lisäksi myös prosessin kontrollointi vaikuttaa tyyllisivuluokkien määrään. Varastointitapaan sekä varastoitavien dokumenttien tai tyyllisivujen nimeämiseen ei tässä mallissa oteta kantaa. Eri kohdekieltä sisältävät tyyllisivut on kuitenkin syytä erotella jonkin metatietomenettelyn avulla toisistaan oikeiden tyyllisivujen löytymisen helpottamiseksi sekä prosessin suorittamisen että tyyllisivujen ylläpitovaiheen aikana.

Muunnos- ja muotoiluprosessi koostuu edellisen luvun kuvauksen mukaisesti kahdesta pääprosessista: XSLT-muunnoksesta ja XSL-muotoilusta. Prosessiin voidaan sisällyttää myös rinnakkais- tai jatkoprosesseja vaadittujen esityssovellustarpeiden mukaisesti. Yksi mahdollinen lisäprosessi voisi olla SVG-prosessi, jossa SVG-kieltä kohdekielenä sisältävän XSL-tyyllisivun ja XML-dokumentin XSLT-muunnoksena syntyy SVG-vektorigrafiikkadoku-

mentti, joka syötetään SVG-muotoiluprosessiin. Vaihtoehtoisesti SVG-dokumentti voi syntyä XSL-muotoiluprosessin lopputuloksena, jolloin tarve ylimääräiseen tyylisivuluokkaan ja sen ylläpitoon poistuu.

Vaikka tässä tutkimuksessa rajoitutaan XSL-tyylikieltä sisältävien tyylisivujen toteuttamisen tutkimiseen, on visualisointiympäristöä tarkasteltaessa luotava silmäys myös erilaisiin esitysovelluksiin. Esitysovelluksia on hyvin eri tyyppisiä. Esitysovellus voi olla esimerkiksi HTML-selain, paperidokumentti, XSL-selain, PDF-selain, ASCII-tekstiselain, SVG-selain tai bittikartta-grafiikkaselain. Visualisointiympäristössä vaadittu esitysovellusten tuki on riippuvainen dokumenttien käsittelyyn tarkoitettujen sovellusten ratkaisusta. Lähinnä kyse on siitä, miten hyvin sovellus tukee rakenteista toimintatapaa ja kykenee esittämään käyttäjälle dokumentin käsittelyssä tarvittavan liiketoimintainformaation. Parhaimmassa tapauksessa dokumenttien käsittelysovellukseen pystytään varastoimaan sekä XML-dokumentit että XSL-tyylisivut ja tuottamaan visualisointiprosessi dokumenttien käsittelysovellukseen upotetulla XSL-selaimella. Huonoimmassa tapauksessa dokumenttien käsittelysovellus on niin rajoittunut, että se ei pysty ottamaan vastaan automatisoituna syötteenä kaikkia XML-dokumentissa olevia, dokumentin liiketoiminnalliseen käsittelyyn tarvittavia tietoja. Tällaisessa tapauksessa käyttäjän liiketoimintaprosessiin liittyvää dokumentin käsittelyä saatetaan kyllä tukea käsittelysovellukseen avautuvan dokumentin kuvan avulla, mutta dokumentin kuvan sisäinen esitysmuoto voi olla rajoitettu vain yhteen dokumenttimuotoon, esimerkiksi TIFF-dokumentteihin.

Tämän tutkimuksen esimerkkiteutuksessa toteutettiin yhden visuaalisen ulkoasun mukainen tyylisivu kahteen eri tyylisivuluokkaan: HTML-tyylisivuna ja XSL-tyylisivuna. Esimerkkiteutuksessa visualisointiympäristön prosessointikomponentteina käytettiin Apachen XML-projektissa kehitettyjä avoimen lähdekoodin Java-sovelluksia. Näiden toimivuus edellyttää Javan 1.2.x version tai uudemman käyttöä. XSL-tyylisivuprosessoriksi valittiin Apachen FOP (Formatting Object Processor), koska se on pisimpään kehitetty XSL-tyylisivuprosessori ja koska se tukee varsin laajasti W3-järjestön XSL-suosituksen versiota 1.0. FOP sisältää oman XML-jäsentimen (Xerxes-J) ja XSLT-prosessorin (Xalan). Sillä pystyy suorittamaan laboratorio-oloissa kaikki tarvittavat visualisointiprosessin muunnokset komentotulkkiin syötettävillä yksinkertaisilla käskyillä. Todellisessa visualisointiympäristössä FOP

voidaan liittää sovellusohjelmista suoraan kutsuttavaksi XSL-prosessointi-komponentiksi.

Apache XML FOP -projektin tavoitteena on tuottaa XSL-muotoilija, joka ymmärtää W3-järjestön XSL-suosituksen mukaisesti XSL-syntaksia ja tuottaa Adoben 1.3-version mukaisia PDF-dokumentteja. Vaikka ensisijainen tuotettava dokumenttiformaatti onkin PDF, on tutkimustyön aikana julkaistulla FOP:n versiolla 0.20.4 mahdollista tukea myös PCL-, PostScript-, SVG-, XML-aluepuu, tulostus-, AWT-, MIF- ja ASCII-tekstimuodon ymmärtäviä esitysovelluksia (FOP 2002).

Esimerkkitoteutuksessa esitysovelluksena käytettiin HTML-selaimina Microsoftin Explorerin versiota 5.5 sekä Netscapen Navigatorin versiota 6.01. PDF-selaimena käytettiin Adoben Acrobat Readerin versioita 4.0 ja 5.0.

6.2 Tyylisivujen muutostekijöiden arviointi

Koska yksittäisellä tyylisivulla on riippuvuuksia hyvin laajalle, voivat hyvin monet eri tekijät aiheuttaa tyylisivujen ylläpitotarpeen. Tyylisivuihin kohdistuvaa ylläpitotyötä aiheuttavat muutokset missä tahansa tyylisivun sovellusympäristössä tai kohteessa, johon tyylisivussa viitataan. Mahdollisia muutostekijöitä ovat

- muutokset visualisointiympäristössä käytössä olevissa suosituksissa ja standardeissa
- muutokset visualisointiympäristössä käytettävissä sovelluksissa, kuten muunnos- ja muotoilukomponenteissa, esitysovelluksissa sekä visualisointiprosessin ohjauksessa
- muutokset skeema- tai DTD-määrittämisessä, jonka mukaisen XML-dokumentin esittämiseen tyylisivu on tarkoitettu.
- muutokset dokumenttien esittämisen vaatimusmäärittämisessä
- muutokset tyylisivuissa viitattujen ulkoisten entiteettien sijainnissa.

Seuraavassa kuvataan tarkemmin näitä muutostekijöitä.

Suosituksiin ja standardeihin liittyviä tyylisivujen muutostekijöitä ovat muutokset XML-suosituksessa sekä muunnos- ja tyylikielisuosituksissa. Esimerkiksi XSLT-, XSL-muotoilu- ja XPath-kielten kehitys on vaiheistettu useaan

versioon ja W3-järjestössä on tutkimustyön tekoheikellä kehitteillä uudet versiot kustakin kielestä. Ylläpitotyötä voivat aiheuttaa myös muutokset visualisointiympäristössä tuettavien kohdekielten suosituksissa tai standardeissa. Suosituksiin ja standardeihin liittyvät muutokset saattavat aiheuttaa muutoksia jokaiseen XSL-tyylisivuvaraston tyyli sivuun. Muutosten toteutus kulkee rinnan visualisointiympäristössä käytössä olevien muunnos-, muotoilu- tai esityskomponenttien muutoksen kanssa. Esimerkkitaapauksessa havaitut tähän astiset muutostarpeet ovat aiheutuneet muun muassa XSLT-kielen nimiavaruuden muutoksesta sekä kielioppimuutoksista XSL-kielen suositusehdotuksen ja suosituksen välillä. Ylläpitotyötä aiheuttavat muutokset ovat todennäköisempiä suositusehdotuksen ja suosituksen välillä kuin eri suositusversioiden välillä.

Teknisiin sovelluksiin liittyviä muutostekijöitä ovat muutokset visualisointiympäristössä käytettävissä muunnos- ja tyyli prosessoreissa. Ylläpitotyötä aiheuttavat käyttöön valittujen sovellusten suosituksesta poikkeavat ratkaisut. Poikkeamat suosituksesta voivat johtua siitä, että sovellus tukee suositusehdotusta, johon on tehty vielä muutoksia ennen kuin varsinainen suositus on julkaistu. Toinen syy ovat sovelluskohtaiset ratkaisut erityisesti muunnossovelluksissa. Jo XSL-kielen nimi – Extensible Stylesheet Language, laajennettava tyylikieli – kertoo, että suositus sallii sovelluskohtaiset laajennukset. Laajennusten käytöstä aiheutuvaa ylläpitotyötä voi helpottaa pidättäytymällä vain suosituksen mukaisissa ratkaisuissa. Myös tarjolla olevat sovellusriippumattomat laajennukset, kuten EXSLT (Extension to XSLT; EXSLT 2002), vähentävät osaltaan XSL-kielen laajennuksista aiheutuvaa ylläpitotyötä.

Eniten ylläpitotyötä aiheuttavat todennäköisesti skeema- tai DTD-määrittysten muutokset. Havaintojen perusteella liiketoimintadokumenttien rakennemääritykset näyttäisivät olevan iteroituvaa kehitystyötä. Siitä kertovat jo tutkimuksessa esimerkkinä käytetyn XML-dokumentin rakennemäärityksen version numero 2.3 sekä Verkkolaskukonsortion yhteisen rakennekuvauksen version numero 1.3. Tyyli sivua toteutettaessa on kuitenkin mahdotonta ennakoita tulevia rakennesääntömuutoksia, joten muutosten toteuduttua on mahdollisesti edessä kaikkien rakennemääritystä vastaavien tyyli sivujen muutos työ.

Mikäli XML-dokumentin rakennemäärityksestä poistetaan joitain rakenneosia, ei ylläpitoa välttämättä tarvita. Rakennemääritystä vastaavissa, muutoksen jälkeen tuotetuissa XML-dokumenteissa ei siten ole poistettuja rakenneosia. Jos näiden rakenneosien esittäminen on kuitenkin edelleen määritettynä tyyllisivussa, mallipohjat eivät vain täsmäydy muunnoksen aikana. Tästä ei aiheudu dokumentin käyttötilanteessa harmia. Sen sijaan rakennemääritysten lisäyksiä vastaavat muutokset on aina toteutettava kaikkiin tyyllisivuihin, sillä muutoin käyttäjä ei näe uusien rakenneosien sisältöä. Tämä ongelma voitaisiin tietysti ratkaista siten, että jokaisen tyyllisivun loppuun lisättäisiin mallipohja niille dokumentin rakenteille, jotka eivät täsmää mihinkään muuhun tyyllisivussa olevaan mallipohjaan. Mutta kuten luvussa 4 olleesta kuvasta 7 nähtiin, ei liiketoimintadokumentin tietosisältö yksinään ole kovinkaan informatiivista käyttäjälle. Skeema- tai DTD-muutosten toteuduttua edessä on siten kaikkien rakennemääritystä vastaavien tyyllisivujen ylläpitotyö.

Rakennemääritysten muutosten kanssa samaan ylläpitomuuttujaluokkaan voidaan ajatella kuuluvan myös rakennesääntöjen soveltamisesta aiheutuva tyyllisivujen ylläpito. Liiketoimintadokumentteja tuottavan järjestelmän rajapintamuunnos voi tuottaa vain osan käytettäväksi sovitun rakennemäärityksen sisällöstä. Mikäli tyyllisivuun toteutetaan mallipohjat vain näiden rajattujen sisältöjen esittämisen osalta, on edessä sama tilanne kuin lisättäessä rakennesääntömääritykseen uusia rakenneosia: käyttäjä ei näe rajapintamuunnokseen lisättyä liiketoimintainformaatiota ennen kuin tyyllisivua on muutettu näyttämään myös lisätyt dokumentin rakenneosiot. Tämä ongelma voidaan kuitenkin ehkäistä siten, että heti tyyllisivujen ensimmäisessä toteutusvaiheessa tyyllisivuun määritetään valmiiksi myös niiden rakennemäärityksessä olevien rakenteiden esittäminen, joita sovellus tai sen rajapinta ei vielä tue.

Dokumenttien esittämisen vaatimusmääritykset ja niiden muutokset ovat peräisin joko ihmisten tarpeista tai säädöksistä. Ihmislähtöiset muutokset liittyvät lähinnä liiketoimintadokumenttien rakenteistamisprosessin aikaiseen kehitystyöhön. Liiketoimintadokumentteja rakenteistettaessa voidaan käyttää hyvinkin paljon aikaa ja vaivaa siihen, millaisin vakiotekstein liiketoimintainformaatio pitäisi otsikoida, missä järjestyksessä dokumentin mikäkin tietosisältö esitetään ja millä tavoin. Ihmislähtöiset vaatimusmääritysten muutokset ja niistä aiheutuva ylläpitotyö kertovat lähinnä heikosta suunnitelmallisuudesta, mikäli niiden vuoksi rakenteistamisprosessi pitkittyy. Tällaiseenkin yl-

ylläpitotyöhön on kuitenkin varauduttava. Lainsäädäntölähtöinen ylläpitotyö tarkoittaa liiketoimintadokumenttien kohdalla esimerkiksi sitä, että yritysrekisterisäädöksen muuttuessa tyyllisivulla vakiotekstinä määritetty yrityksen LY-tunnus muutetaan käsitteeksi Y-tunnus ja poistetaan samalla kaupparekisterinumeroiden esittäminen. Tällaiset muutokset on toteutettava jokaiselle tyyllisivulle, jossa on määritetty niiden esiintyminen.

Tekninen tyyllisivujen muutostekijä ovat muutokset tyyllisivuissa viitattujen ulkoisten entiteettien sijainnissa. Ulkoisia entiteettejä voivat olla kuvat, logot, XML-dokumentit tai ositetun tyyllisivun komponentit. Ylläpitotyötä aiheuttavat muutokset viitattujen kohteiden sijainnissa tai saatavuudessa. Mikäli entiteetit eivät ole syystä tai toisesta saatavilla, on seuraukset vaihtelevat. Seurauksena voi olla puutteellinen visuaalinen esitys, kuten silloin, kun viitattu kuva ei ole saatavilla viitatussa sijainnissa. Pahempi seuraus on koko visualisointiprosessin keskeytyminen, mikä voi tapahtua silloin, kun ositetun tyyllisivun komponentti ei ole saatavilla. Ulkoisten entiteettien käyttö on tämän vuoksi syytä arvioida ja suunnitella huolella koko visualisointiympäristö ja tyyllisivujen käyttötarpeet huomioiden, jotta vältytään etenkin tilanteilta, joissa visualisointiprosessi keskeytyy siitä syystä, että entiteetti ei ole saatavilla.

6.3 Tyyllisivujen vaatimusmäärittäminen

Yksittäisen tyyllisivun toteutus alkaa vaatimusmäärittämisellä, jota seuraa suunnittelu- ja toteutus- ja testausvaiheet. Mikäli samaan aikaan toteutetaan tyyllisivuja eri kohdekieliin, ovat vaatimusmäärittäminen ja mahdollisesti osa muista vaiheista yhteisiä tai ainakin yhteistyössä tehtäviä eri tyyllisivuille.

Edellisissä kohdissa kuvattujen visualisointiympäristön kartoituksen sekä tyyllisivujen muutostekijöiden arvioinnin perusteella tutkimustyön esimerkkitoteutukselle asetettiin erilaisia vaatimuksia, jotka voidaan jakaa tyyllisivujen monikäyttöisyys- ja ylläpidettävyyksivaatimuksiin sekä vaatimuksiin uudelleenkäytettävistä tyyllisivujen osista. Näiden lisäksi kohteena olevalla sovellusalueella on huomioitava kirjanpito- ja raportointi. Seuraavassa on kuvattu asetetut vaatimukset edellä esitetyn luokituksen mukaisesti sekä keinot vaatimusten täyttämiseksi esimerkkitoteutuksessa.

Vaatus 1: Tyylisivujen monikäyttöisyys

Tyylisivujen monikäyttöisyydellä tarkoitetaan sitä, että yhden tyylisivun määrittämän XML-dokumentin esityksen tulee olla käytettävissä erilaisissa käyttötilanteissa. Erilaisilla käyttötilanteilla tarkoitetaan sitä, että

- yhtä tyylisivua tulee voida soveltaa usean eri organisaation tuottamien liiketoimintadokumenttien visualisointiin
- tyylisivun tulee olla sovellettavissa useiden eri järjestelmien tuottamaan, käytössä olevan skeeman mukaisen XML-dokumentin sisällön esittämiseen riippumatta siitä, tuottaako järjestelmä kaikki skeemassa määritetyt valinnaiset rakenteet vai vain osan niistä
- tyylisivun tulee olla sovellettavissa eri kieltä ymmärtävien käyttäjien tarpeisiin
- tyylisivun tulee soveltua sekä dokumentin tuottaneen organisaation että dokumentin vastaanottaneen organisaation työprosessien tukemiseen
- tyylisivun tulee tukea dokumentin esittämistä erilaisissa esitysovelluksissa
- tyylisivun tulee sallia muunnoskomponenttien variaatiot.

Seuraavassa kuvataan esimerkkitoteutuksessa tehdyt ratkaisut edellä listattujen tyylisivujen monikäyttöisyysvaatimusten täyttämiseksi.

Useiden organisaatioiden tuottamien liiketoimintadokumenttien tuki ratkaistiin esimerkkitoteutuksessa siten, että visuaalisen ulkoasun lähtökohdaksi otettiin standardoitu visuaalinen lomakemalli, jota laajennettiin standardista puuttuvilla, tapausympäristössä käytössä olevan rakennemäärityksen rakenteilla. Visuaalinen ulkoasu muodostaa esityksen typograafiset raamit, jonka sisällä eri toimittajien järjestelmien muodostamien XML-dokumenttien tietosisältö vaihtelee. Dokumentin tuottavan organisaation mukaan vaihtuva logon entiteettiä on sidottu dokumentin tuottajakohtaisesti yksilöityvään tietosisältöön, jonka mukaan logotiedostot on nimetty. Entiteettiä muodostuu siten dynaamisesti dokumentin tuottajan perusteella.

Vaatus tyylisivun sovellettavuudesta eri järjestelmien tuottamien XML-dokumenttien visualisointiin ratkaistiin esimerkkitoteutuksessa siten, että tyylisivuun määritettiin esitysrakenteet kaikille skeeman mukaisille dokumentin

käyttötilanteen tietotarpeeseen liittyville sisältöosille riippumatta siitä, ovatko ne rakennemäärityksessä pakollisia tai vapaavalintaisia rakenneosia. Samoin esimerkkitoteutuksen tyyllisivut tukevat kaikkia esimerkkitoteutukseen valitun dokumenttityypin eri alatyyppejä. Laaja sovellettavuus eri alatyypeille on toteutettu esitysrakenteiden vaihtoehtoisuudella suhteessa XML-dokumentin sisältöön.

Liiketoimintadokumenttien tyyllisivut sisältävät runsaasti vakiotekstiä, jota ei ole esityksen kohteena olevassa XML-dokumenteissa. Vaatimus tyyllisivun sovellettavuudesta eri kieltä ymmärtävien käyttäjien tarpeisiin edellyttää siten sitä, että tyyllisivun tulee sisältää vakiotekstit eri kielillä esitettynä. Esimerkkiteutuksessa ensisijaiseksi vakiotekstin kieleksi määriteltiin suomi ja ruotsi. Näille vaihtoehtoiseksi vakiotekstikieleksi määriteltiin englanti. Liiketoimintadokumenteissa dokumentin tuottaja yleensä tietää vastaanottajan käyttämän kielen. Tieto vastaanottajan ymmärtämästä kielestä voidaan siten sisällyttää liiketoimintadokumentiin, jolloin vakiotekstien kielen valinta XML-dokumentin sisällön perusteella on mahdollista visualisointiprosessin aikana.

Vaatimus tyyllisivun sovellettavuudesta sekä dokumentin tuottaneen organisaation että dokumentin vastaanottaneen organisaation työprosessien tukemiseen edellytti esimerkkitoteutukseen valitun lomakestandardin ulkoasun muuttamista. Muutos tapahtui selvittämällä dokumentin vastaanottajan työprosesseihin liittyvät tietotarpeet. Selvitys tapahtui haastatteleamalla sovellettualueen dokumenttien käytön asiantuntijaa. Tietotarpeiden lisäksi huomioitiin digitaalisessa muodossa käsiteltävien dokumenttien käyttöympäristön ominaisuudet. Esimerkkiteutuksessa tämä tarkoitti sitä, että tyyllisivun muodostamassa esityksessä näytetään dokumentin yläosassa ensin liiketoimintadokumentin yksilöivät perustiedot ja heti näiden alla dokumentin hyväksymiseen ja muuhun käsittelyyn liittyvät tietosisällöt ryhmiteltyinä kokonaisuuksinaan. Paperidokumentissa vastaavat tiedot näytettäisiin joko dokumentin ensimmäisen sivun alareunassa tai useampisivuisen dokumentin kohdalla viimeisen sivun lopussa.

Vaatimus yhdellä tyyllisivulla saavutettavasta laajasta esitysovellusten tuesta pystyttiin esimerkkitoteutuksessa täyttämään vain osittain. Ylläpidon näkökulmasta tyyllisivujen toteutuksen yleisenä pyrkimyksenä on tyyllisivujen

lukumäärän minimointi. Ideaalina tavoitetilana on tällöin yksi tyyლისivu, jonka kohdekielenä on kaikkien visualisointiympäristössä tuettujen esitysovellusten ymmärtämä universaali esityksen kuvauskieli. XSL-tyylikielen tuen yleistyminen erilaisissa esitysovelluksissa tekisi siitä universaalinkin XML-dokumenttien esityksen kuvauskielen. Vaikka XSL-kieltä onkin jo tutkimuksen tekohetkellä sovellettu eri tavoin, eivät käytännön sovellukset kuitenkaan toistaiseksi tue XSL-tyylikieltä vielä kovin laajasti (XMLSoftware 2002). Tämän vuoksi vaatimusta laajasta esitysovellusten tuesta ei saatu täytettyä, vaan kullekin kohdekielille toteutettiin oma tyyლისivu. Yhden kohdekielen ymmärtämien esitysovellusten variaatioita tuettiin esimerkkitoteutuksessa siten, että tyyლისivujen kohdekielissä pidättäytyttiin niiden suosituksessa. Siten esimerkiksi eri HTML-selainten suosituksista poikkeavia erityispiirteitä ei tuettu.

Esitysympäristön muunnos- ja muotoiluprosessit toteuttavien komponenttien variaatiomahdollisuutta tuettiin esimerkkitoteutuksessa siten, että tyyლისivuisissa pidättäytyttiin vain XSLT-suosituksen tarjoamassa toiminnallisuudessa eikä käytetty esimerkiksi XSLT-prosessorikohtaisia laajennuksia. Tämän ansiosta esimerkkitoteutuksen tyyლისivuja voidaan käyttää eri valmistajien muunnos- ja muotoiluprosessoreilla toteutettaviin munnoksiin.

Vaatus 2: Tyyლისivujen ylläpidettävyys

Tyyლისivujen ylläpidettävyydellä tarkoitetaan sitä, että tyyლისivujen sisältöä tulee pystyä kohtuullisella vaivalla muuttamaan. Kohtuullinen vaiva on varsin summittainen määre, joten käytännössä vaatimus tarkoittaa tyyლისivujen ohjelmakoodin ymmärrettävyyttä ja mahdollisuutta muuttaa esityksen rakennetta vähäisin tai vähän aikaa vievin toimenpitein. Keinoja tyyლისivujen ylläpidettävyystavoitteen saavuttamiselle ovat:

- tyyლისivujen koodin kommentointi
- tyyლისivujen koodin looginen osittaminen
- XPath-kielen ilmaisujen selkeys
- esityksen mitta-arvojen suhteellisuus.

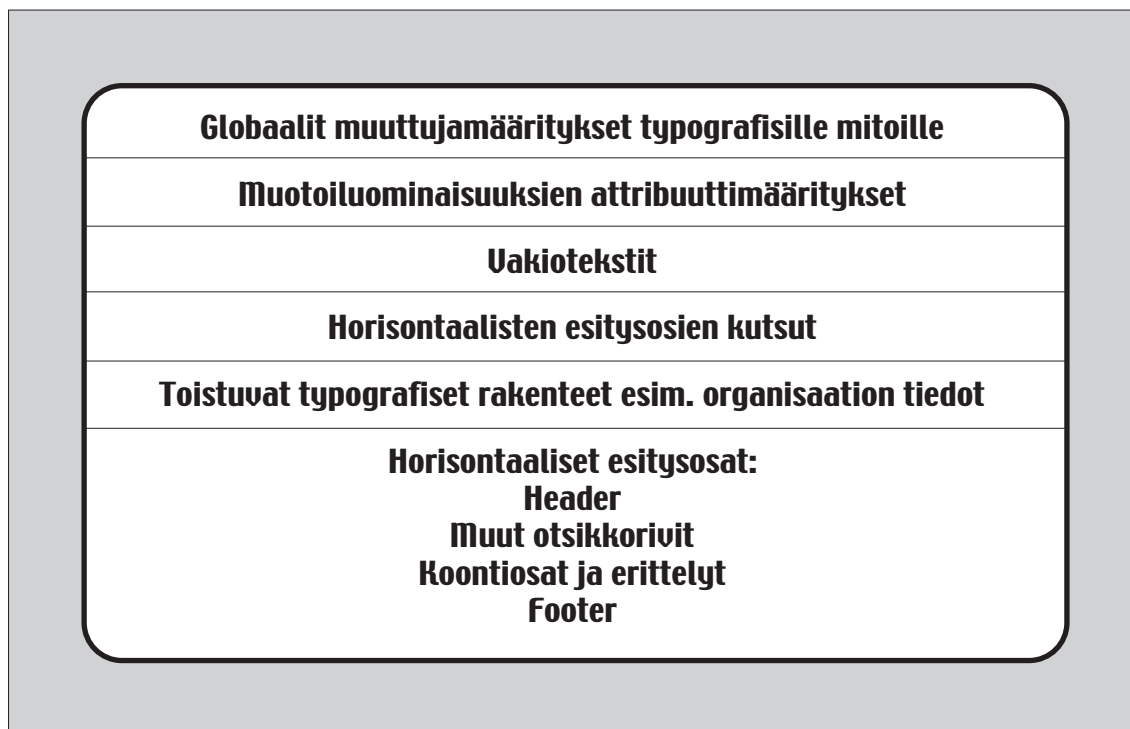
Seuraavassa selvitetään tarkemmin edellä listattuja keinoja sekä ratkaisuja, joilla esimerkkitoteutuksessa pyrittiin täyttämään ylläpidettävyysvaatus.

Ensimmäinen mahdollista ylläpitotyötä helpottava ratkaisu muutostyön toteuttajaa ajatellen on tyylisivujen koodin kommentointi. Huolellisella kommentoinnilla ja mahdollisella tyylisivuun liitettyllä metatiedolla tyylisivusta voidaan saada aikaan itsensä kommentoiva ohjelmistokomponentti. Kommentointi toteutettiin esimerkkitoteutuksessa siten, että tyylisivujen alussa kuvattiin kommentoiden tyylisivun käyttötarkoitus sekä ohjelmakoodin looginen rakenne.

Ylläpitotyötä helpottaa myös tyylisivujen looginen osittaminen. Ositus nopeuttaa ylläpitoa erityisesti silloin, kun on kyse satoja rivejä sisältävästä tyylisivusta, mutta myös pienemmissä tyylisivuissa osituksesta on hyötyä. Esimerkkiteutuksen tyylisivut ositettiin siten, että typograafisia arvoja määrittävät globaalit muuttujat, kohdekielien attribuuttimääritykset sekä vakiotekstit koottiin omiin loogisiin lohkoihinsa.

Koska esimerkkitoteutuksen liiketoimintadokumenttien ulkoasu vastasi selkeää lomaketyyppistä taulukkorakennetta, voitiin tyylisivujen ositusta jatkaa myös visuaalisen esityksen raamit kasaavan taulukon rivien perusteella. Esimerkkiteutuksen tyylisivut ositettiin siten taulukkoon allekkain tulostettavia rivejä tuottaviin mallipohjiin, jotka nimettiin yksilöivien tunnistein name-attribuutilla. Esityksen kohteena olevan XML-dokumentin juureen täsmävässä mallipohjassa kutsutaan määrittelyssä järjestyksessä kutakin taulukon rivin muodostavaa nimettyä mallipohjaa. Osituksen lopuksi usein toistuvat typografiset elementit, kuten vaakalinjat, koottiin vielä omiin yksilöidysti nimettyihin mallipohjiin, joita kutsutaan nimellä aina, kun esitykseen halutaan tuottaa kyseinen typografinen elementti. Esimerkkiteutuksessa tyylisivun ositus muodostuu kuvan 16 mukaisesti seuraavista osista: globaalit, lähinnä typograafisia mittoja määrittävät muuttujat, muotoiluominaisuuksien attribuuttimääritykset, vakiotekstit, rakenteessa usein toistuvien typografisten rakenteiden mallipohjat ja lopuksi horisontaalisessa esiintymisjärjestyksessä jokainen taulukon rivin muodostava mallipohja.

Koska tyylisivujen tuottama esitys perustuu suurelta osin XML-dokumentin rakenneosien valintaan ja täsmäykseen XPath- ja XSLT-kielien avulla, on yksi ylläpitotyötä helpottava tekijä XPath-ilmainsujen ja sisältöjen valinnan koodauksen selkeys. Selkeyttä on syytä korostaa, sillä XPath ja XSLT sallivat viittaus- ja valintasyntaksin, joka voi näyttää kieliin vähän perehtyneille kryptisel-



Kuva 16. Esimerkkitoteutuksen yksittäisen tyylisivun looginen ositus.

tä. Kryptisyys perustuu muutamiin varattuihin merkkeihin, joita ovat XPat-hissa piste (.) ja XSLT:ssa attribuuttien sisällön valintaan käytettävät kaarisulut ({}). Näiden käytön sijaan on suositeltavaa viitata XML-dokumentin rakenteisiin ja niiden valintaan selkeästi varsinkin silloin, kun on todennäköistä, että ylläpitotyötä toteuttaa henkilö, joka ei ole tottunut XSL-kielen käyttäjä.

Tyylisivujen ylläpitotyötä helpottaa myös esityksen mitta-arvojen määrittelyminen suhteellisesti. Suhteellisella mitta-arvojen määrittelyllä tarkoitetaan sitä, että tyylisivuun kiinnitetään mahdollisimman vähäinen määrä absoluuttisia leveys- ja korkeusarvoja. Näiden muutamien perusarvojen pohjalta lasketaan muut mitta-arvot. Esimerkiksi XSL-muotoilukielellä määritellyn esityksen leveyden ja marginaalin leveyden perusteella voidaan laskea esityksen runko-osan leveysarvo. Esimerkkitoteutuksessa HTML-syntaksia sisältävän tyylisivun mitat määriteltiin kokonaan suhteellisina prosenttiosuuksina. Koska esimerkkitoteutuksessa käytetty FOP-muotoiluprosessorin tutkimustyön aikainen kehitysversio vaatii leveysarvojen absoluuttisen määrittelyn, ratkaistiin mitta-arvojen suhteellisuus esimerkkitoteutuksessa laskennallisesti.

XSL-muotoilukielen syntaksia sisältävässä tyylisivussa esimerkiksi kiinnettiin ainoastaan yhden taulukon sarakkeen leveys, jonka perusteella muiden sarakkeiden leveysarvot määräytyvät.

Vaatus 3: Uudelleenkäyttö

Edellä kuvatun loogisen osittamisen jälkeen voidaan tyylisivun ositus toteuttaa myös fyysisesti. Tämä tukee tyylisivujen osien uudelleenkäyttöä erityisesti silloin, kun yhtä fyysistä tyylisivun osaa voidaan käyttää usean eri tyylisivun osana. Esimerkiksi vakiotekstien määritykset tai muut eri tyylisivuille yhteiset rakenteet voidaan tallentaa fyysisesti erillisiin tyylisivuihin, joita kutsutaan muilta tyylisivuilta. XSLT sallii tyylisivujen ulkoisten mallipohjien liittämisen ohjelmointikielien kirjastojen kutsujen tavoin. Itse toteutettujen mallipohjakirjastojen lisäksi tyylisivuihin voidaan liittää myös yleisen XSLT-kirjaston (XSLT Standard Library) mallipohjia esimerkiksi merkkien käsittelemiseksi (Ball 2001).

Fyysinen modularisointi saattaa aiheuttaa hankaluuksia silloin, kun tyylisivuja on tarpeen välittää dokumentteja tuottaviin ja vastaanottaviin järjestelmiin. Tätä silmällä pitäen esimerkkitapauksessa nähtiin kannattavammaksi pidättäytyä fyysisesti yhtenäisissä tyylisivuissa. Fyysistä modularisointia kannattaa kuitenkin harkita silloin, kun tyylisivujen määrän odotetaan kasvavan ja kun useille tyylisivuille on nähtävissä selkeitä yhtenäisiä rakenteita. Ylläpidon näkökulmasta olisi suositeltavampi vaihtoehto, mikäli jo jotain olemassa olevaa tyylisivua tai sen osaa voitaisiin käyttää myös eri organisaatioiden välillä joko sellaisenaan tai hieman mukauttaen.

Myös yksittäisten elementtien tai attribuuttien sisältöjen esittämisen muotoilu voidaan määrittää yhteen fyysisesti erilleen tallennettuun tyyli tiedostoon. Esimerkiksi suomalaisia, ranskalaisia ja amerikkalaisia dokumenttien käyttäjiä varten voidaan toteuttaa yksi fyysisesti erillinen tyylisivudokumentti, joka sisältää kaikki variaatiot numeeristen raha-arvojen muotoilulle. Myös typografiset ominaisuudet voitaisiin kerätä fyysisiin itsenäisiin moduleihin, mikäli pystytään määrittelemään riittävän yleinen tapa kuvata typografisia ominaisuuksia kaikille tarvittaville kohdekielille.

Vaatus 4: Kirjanpito näkökulman huomiointi

Kirjanpitolainsäädännön (FINLEX 2003) asettaman näkökulman huomiointi tarkoittaa sitä, että tyylisivuja toteutettaessa esitykseen ei tuoteta sellaista tietosisältöä, jota ei löydy esityksen kohteena olevasta XML-dokumentista.

Vaatumuksen huomiointi tarkoitti esimerkkitoteutuksessa sitä, että numeeristen arvojen muotoilu ei saanut olla laskennallista tai pyöristämistä. Kirjanpito näkökulmasta olisi ollut sallittua muokata lukuarvojen desimaalien määrää sekä kokonaislukujen ja desimaalien erottamiseen merkintätapaa. Esimerkkitooteutuksessa ei kuitenkaan muokattu lukuarvoja millään tavoin, vaan ne esitettiin siinä muodossa, missä XML-dokumentin tuottanut järjestelmä oli ne dokumenttiin tuottanut.

6.4 Esimerkkitooteutuksen arviointi

Esimerkkitooteutuksen tarkoituksena oli tuottaa monikäyttöinen tyylisivumääritys siten, että ylläpito näkökulma on huomioitu. Edellä kuvattua tyylisivujen vaatimusmääritystä testattiin siten, että vaatimusmääritysten perusteella toteutettiin kaksi kohdekieleltään eroavaa tyylisivua: HTML-kieltä sisältävä tyylisivu ja XSL-muotoilukieltä sisältävä tyylisivu. Vaatimusmäärityksen asettamat tavoitteet olivat ratkaistavissa samalla tavoin kummassakin kohdekieleessä.

Esimerkkitapauksessa oli ennen tutkimustyötä toteutettu muutama tyylisivumääritys, joita voidaan käyttää konstruktion alustavan arvioinnin vertailukohtana. Tyylisivun monikäyttöisyydestä saatiin palautetta dokumenttien käyttäjiltä, jotka kiittelivät esimerkkitoteutuksessa tuotetun tyylisivun avulla muodostuvaa dokumentin esitystä siitä, että heidän liiketoimintadokumenttien käsittelyyn liittyvä tietotarpeensa tulee uudesta esityksestä helpommin tyydytettyä kuin vanhasta. Myös visuaalista esitystä pidettiin selkeänä.

Tyylisivujen ylläpidettävyyttä arvioi yksi tutkimuksen tapausympäristön ohjelmoija, joka oli perehtynyt XSL-kieleen, mutta ei ollut aiemmin tarkastellut esimerkkitoteutuksessa tuotetun ja vertailukohtana käytetyn vanhan tyylisivun koodia. Saadussa palautteessa esimerkkitoteutuksessa tuotettujen tyyli-

sivujen ohjelmakoodia pidettiin ymmärrettävämpänä ja ylläpidettävämpänä kuin aiempia toteutuksia.

Tyylisivujen uudelleenkäyttöä heikentää pidättäytyminen tyylisivujen fyysisestä osittamisesta. Esimerkkitoteutusta olisi voitu parantaa tyylisivujen fyysisen osittamisen suunnittelulla. Samoin toteutusmallia voitaisiin kehittää laajemmin suunnittelemalla yksi metatason esityskuvaus, josta voidaan johdattaa tyylisivut kullekin visualisointiympäristössä tuetulle kohdekielelle, jotta tyylisivujen ylläpitotyötä voitaisiin edelleen vähentää.

Tutkimuksessa tuotetulla tyylisivujen toteutusmallilla on yksittäisten tyylisivujen esimerkkitoteutuksen vuoksi myös mahdollisia haitallisia seurauksia. Haitat voivat aiheutua siitä, että esimerkkitoteutuksen kohdeympäristössä on nyt mahdollista ottaa käyttöön esimerkkitoteutuksessa tuotetut tyylisivut, joiden sisäinen ohjalmarakenne poikkeaa aiemmin tuotettujen tyylisivun sisäisestä rakenteesta. Ylläpitotilanteissa ohjelmarakenteiden eroavaisuudella voi olla haitallisia seurauksia esimerkiksi ohjelmakoodin ymmärrettävyyden ja tulkinnan osalta. Tyylisivujen toteutusmallin laajamittainen käyttöönotto edellyttäisi esimerkkitapauksessa visualisointiprosessin laajentamista kuvan 15 mukaisesti XSL-muotoiluprosessilla.

7 YHTEENVETO

Tämän tutkimuksen tarkoituksena oli selvittää, miten XSL-tyylikielillä kuvautet, rakenteisten liiketoimintadokumenttien visualisointiin tarkoitetut tyyllisivut tulisi toteuttaa. Toteutustavan selvittämisessä painotettiin organisaatioiden välisiä dokumentteja sekä tyyllisivujen ylläpitoonäkökulmaa. Tutkimuskykyksen ratkaisemiseksi perehdyttiin aluksi dokumenttien hallinnan kehittämistä käsittelevään kirjallisuuteen. Kirjallisuuden pohjalta luotiin kuva dokumentin elinkaaresta, dokumenttien rakenteistamisprosessista sekä timanttillista, jonka perusteella hahmotettiin dokumentin suhdetta organisaation prosesseihin, työrooleihin sekä dokumenttien käsittelyssä sovellettavaan teknologiaan. Tämän jälkeen yleistä dokumenttien hallinnan teoriaa suhteutettiin liiketoimintadokumentteihin perehtymällä liiketoimintadokumenttien merkitykseen, sisältöön ja tuottamiseen. Lisäksi perehdyttiin dokumenttien rakenteistamiseen sekä rakenteisten liiketoimintadokumenttien välittämiseen tuottajalta vastaanottajalle.

Tutkimus jatkui rakenteisten dokumenttien esittämiseen liittyvien mahdollisuuksien ja ongelmien kuvaamisella ja pohdinnalla. Tyyli- ja muunnoskielten käytöllä dokumenttien visuaalisen ulkoasun määrittämisessä havaittiin olevan useita hyödyllisiä, mutta myös haitallisia vaikutuksia. Muunnos- ja tyylikielten avulla tapahtuvan dokumenttien visualisoinnin todettiin kuitenkin tuovan enemmän hyötyjä kuin haittoja, joskin ongelmakohtat on tiedostettava dokumenttien rakenteistamisprosessissa. XML-dokumenttien visualisointiin soveltuvia muunnos- ja tyylikieliä löydettiin useita. Näistä XSL-tyylikielen todettiin tarjoavan eniten mahdollisuuksia sekä kielen monipuolisen sovellettavuuden että saatavilla olevien sovellusohjelmien vuoksi.

XSL-kielen todettiin muodostuvan erillisistä muunnos- ja muotoilukielistä, joista tässä tutkimuksessa perehdyttiin tarkemmin XSL-muotoilukieleen. XSL-muotoilukieli on sivunkuvauskieli, jonka soveltaminen edellyttää kielen abstraktin tason sivunkuvauksen, aluemallin, ymmärtämistä. XSL-kielen soveltamiseen liittyvän muunnos- ja muotoiluprosessin sekä tyyllisivujen sisällön ja merkityksen kuvaamisen pohjalta muodostettiin tämän tutkimuksen keskeinen tulos: tyyllisivujen toteutusmalli. Toteutusmalli muodostuu tyylikielten soveltamisessa vaadittavien tehtävien mukaisesti kolmesta erillisestä osasta: visualisointiympäristön kartoituksesta, muutostekijöiden arvioinnista

sekä tyyllisivujen vaatimusmäärytyksestä. Kutakin mallin vaihetta, vaiheeseen liittyviä osatehtäviä sekä kokemuksia ja ratkaisuja käsiteltiin tutkimuksessa tuotettuun esimerkkitoteutukseen peilaten. Toteutusmallista voidaan havaita, että myös yksittäisten tyyllisivujen toteuttaminen vaatii monien visuaaliointiympäristöön liittyvien seikkojen huomiointia.

Tutkimus on aikaisempaan tyyllisivujen käyttöä soveltavaan tutkimukseen verrattuna kattavampi. Tutkimuksen vahvuutena on tyyllisivujen käyttöön liittyvän teoreettisen ja käytännön toteutukseen perustuvan tiedon kokoaminen yhteen liiketoimintadokumenttien näkökulmaa painottavalla tavalla. Aiemmat tutkimusalueen piiriin kuuluvat tutkimukset ovat keskittyneet lähinnä tyyllisivulähestymistavan mahdollisuuksien kuvaamiseen, muunnosprosessin kehittämiseen ja tyyllisivulähestymistavasta saatujen kokemusten kuvaamiseen (Cooper ym. 2000; Lie & Saarela 1999; Honkaranta, Lyytikäinen & Tiitinen 2001; Honkaranta & Tyrväinen 2001). Aiempien tutkimusten kokemukset ovat perustuneet XSLT-kielillä kuvattuihin muunnoksiin, joissa tyyllisivujen kohdekielenä on HTML-kieli. Tutkimusta XSL-muotoilukielen käytöstä tai organisaatioiden välisten dokumenttien visualisoinnissa käytettävien tyyllisivuvuorastojen tai yksittäisten tyyllisivujen sisäisen rakenteen hallinnoinnista ei löytynyt. Liiketoimintadokumenttien välittämisen alueella tehtävä tutkimus on tuonut lähinnä uutta tietoa dokumenttien rakenteen muunnoksesta ja muunnosten automatisoinnista (Diaz, Wüstner & Buxmann 2002). Dokumenttien esittämiseen liittyvä tuorein tutkimus on taas keskittynyt eri muotoisten dokumenttien esittämisen mahdollistamiseen yhdessä dokumentin katseluun tarkoitettussa sovellusohjelmassa (Phelps & Wilensky 2001).

Tutkimuksen tuotoksia voidaan soveltaa sellaisenaan liiketoimintadokumenttien tuottamiseen, välittämiseen ja käsittelyyn liittyvien järjestelmien ja komponenttien kehityksessä. Myös osaa rakenteisten dokumenttien esittämistä pohtivan luvun sisällöstä voidaan hyödyntää eri organisaatioiden liiketoimintadokumenttien rakenteistamisprosessin aikana. Vaikka tutkimuksessa tarkastellaan erityisesti liiketoimintadokumentteja ja niistäkin esimerkkinä lasku-tyyppistä dokumenttia, on tutkimuksessa tuotettu tyyllisivujen toteutusmalli sovellettavissa myös muun tyyppisten rakenteisten dokumenttien tyyllisivujen toteuttamiseen. Soveltamisessa on kuitenkin huomioitava erityisesti liiketoimintadokumenttien poikkeavat piirteet muihin dokumentteihin verrattuna. Muista rakenteisista dokumenteista liiketoimintadokumentit

poikkeavat siten, että niiden tietosisältö koostuu lähes pelkästään yksittäistä liiketoimintainformaation palasista, joiden ymmärrettävyys perustuu joko dokumentin rakenteen merkitsemisessä käytettyyn rakenneosien nimeyksen semantiikkaan tai tyyllisivuilla esitykseen liitettävään vakiotekstiin.

Tutkimuksen rajoitteena on tyyllisivujen toteutusmallin ja erityisesti tyyllisivujen vaatimusmääritysten teoreettislähtöisyys. Vaatimusmääritysten lähtökohdana ovat olettamukset ylläpitoon vaikuttavista muutostekijöistä, jotka on johdettu tyyllisivujen riippuvuuksien perusteella. Ylläpitoon vaikuttavista tekijöistä tulisi saada lisää käytäntöön pohjautuvaa tietoa, jotta mallin oikeellisuutta voitaisiin paremmin arvioida ja jotta mallia voitaisiin kehittää edelleen. Myös ylläpitoon vaikuttavien muutosten aiheuttajien tutkiminen edistäisi tyyllisivujen hallinnan ja toteutuksen suunnittelua.

Edellisten lisäksi mielenkiintoinen tutkimuksessa vastaan tullut ongelma-alue liittyy rakenteisista dokumenteista tuotettavien erilaisten visuaalisten esitysten haittatekijöihin. Tutkimuskohteena voisi olla esimerkiksi se, aiheuttaako dokumentin tuottajan ja vastaanottajan näkemä erilainen ulkoasu samasta liiketoimintadokumentista liiketoimintaprosessin etenemiselle ongelmia.

LÄHDELUETTELO

Kirjallisuuslähteet:

- Adler S., Berglund A., Caruso J. ym. (toim.). 2001. Extensible Stylesheet Language (XSL) Versio 1.0 [online]. W3-järjestön suositus 15.10.2001. [Viitattu 7.3.2002]. Saatavilla WWW-muodossa <URL:<http://www.w3.org/TR/2001/REC-xsl-20011015/>>.
- Apparao V., Byrne S., Champion M. ym. (toim.). 1998. Document Object model (DOM) Taso 1 Versio 1.0 [online]. W3-järjestön suositus 1.10.1998. [Viitattu 23.9.2002]. Saatavilla WWW-muodossa <URL: <http://www.w3.org/TR/REC-DOM-Level-1/>>.
- Banerjee S. & Kumar R. L. 2002. Managing electronic interchange of business documents. Communications of the ACM 45(7), 96-102.
- Berners-Lee T. & Connolly D. 1995. Hypertext Markup Language - 2.0 (HTML) [online]. [Viitattu 8.10.2002]. Saatavilla WWW-muodossa <URL:<http://www.ietf.org/rfc/rfc1866.txt>>
- Birbeck M., Duckett J., Gudmundsson O. G. ym. 2001. Professional XML 2nd edition. Chigaco: Wrox Press Inc.
- Bos B., Lie H. W., Lilley C. & Jacobs I. 1998. Cascading Style Sheets (CSS) Taso 2 [online]. W3-järjestön suositus 12.5.1998. [Viitattu 2.10.2002] Saatavilla WWW-muodossa <URL: <http://www.w3.org/TR/REC-CSS2/>>.
- Bos B. 2002a. Web Style Sheets home page [online]. W3C 4.3.2002. [Viitattu 11.3.2002]. Saatavilla WWW-muodossa <URL:<http://www.w3.org/Style/>>.
- Bos B. 2002b. Style Activity Statement [online]. W3C 22.10.2002 [Viitattu 11.3.2002]. Saatavilla WWW-muodossa <URL:<http://www.w3.org/Style/Activity/>>.
- Bowler J.B., Brown C., Capsimails M. & Cole L. 2001. Scalable Vector Graphics (SVG) [online]. W3-järjestön suositus 4.9.2001. [Viitattu 9.11.2002]. Saatavilla WWW-muodossa <URL: <http://www.w3.org/TR/2001/REC-SVG-20010904/>>.

- Bray T., Hollander D. & Layman A. (toim.). 1999. Namespaces in XML [online]. W3-järjestön suositus 14.1.1999. [Viitattu 3.3.2002]. Saatavilla WWW-muodossa <URL:<http://www.w3.org/TR/1999/REC-xml-names-19990114/>>.
- Bray T., Paoli J., Sperberg-McQueen C. M. & Maler E. (toim.). 2000. Extensible Markup Language (XML) 1.0 (Second Edition) [online]. W3-järjestön suositus 6.10.2000. [Viitattu 7.3.2002]. Saatavilla WWW-muodossa <URL:<http://www.w3.org/TR/2000/REC-xml-20001006/>>.
- Clark J. (toim.). 1999a. Associating Style Sheets with XML documents Versio 1.0 [online]. W3-järjestön suositus 29.6.1999. [Viitattu 14.10.2002]. Saatavilla WWW-muodossa <URL:<http://www.w3.org/1999/06/REC-xml-stylesheet-19990629/>>.
- Clark J. (toim.). 1999b. XSL Transformations (XSLT) Versio 1.0 [online]. W3-järjestön suositus 16.11.1999. [Viitattu 7.3.2002]. Saatavilla WWW-muodossa <URL:<http://www.w3.org/TR/1999/REC-xslt-19991116/>>.
- Clark J. & DeRose S. (toim.). 1999. XML Path Language (XPath) Versio 1.0. [online]. W3-järjestön suositus 16.11.1999. [Viitattu 7.3.2002]. Saatavilla WWW-muodossa <URL:<http://www.w3.org/TR/1999/REC-xpath-19991116/>>.
- Cooper M., Santacruz Valencia L. P., Donnelly A. & Sergeant P. 2000. User interface approaches for accessibility in complex World-Wide-Web applications - an example approach from the PEARL project [online]. Teoksessa P. L. Emiliani & C. Stephanidis (toim.) Proceedings of the 6th ERCIM Workshop "User Interfaces for All", CNR-IROE, Florence, Italy, 25-26.10.2000. [Viitattu 11.3.2002]. Saatavilla Adobe Acrobat -muodossa <URL:http://ui4all.ics.forth.gr/UI4ALL-2000/files/Position_Papers/Cooper.pdf>.
- Diaz L. M., Wüstner E. & Buxmann P. 2002. Inter-organizational document exchange - facing the conversion problem with XML. Teoksessa Proceedings of the 2002 ACM Symposium on Applied Computing, Madrid, Espanja, Maaliskuu 2002. New York: ACM Press, 1043-1047.

- FINLEX. 2003. Valtion säädöstietopankki [online]. [Viitattu 5.2.2003]. Saatavilla WWW-muodossa <URL:<http://www.finlex.fi/lains/index.html>>.
- Glushko R.J, Tenenbaum J.M. & Meltzer B. 1999. An XML framework for agent-based e-commerce. *Communications of the ACM* 42 (3), 106-114.
- Grosso P. & Walsh N. 2000. XSL concepts and practical use [online]. XML Europe 2000, Pariisi, Ranska, 12.6.2000. Päivitetty 6.6.2002. New York: Arbortext Inc. ja Santa Clara:Sun Microsystems Inc. [Viitattu 30.9.2002]. Saatavilla HTML-muodossa <<http://www.nwalsh.com/docs/tutorials/xsl/xsl/slides.html>>.
- Haimes R. 1994. Document interface. *Interactions* 1(4), 15-18.
- Harold E. R. 1999. *The XML Bible*. New York: Hungry Minds, Inc.
- Heeman F. C. 1992. Granularity in structured documents [online]. *Electronic Publishing - Origination, Dissemination, and Design*, 5(3), 143-155.
- Hirvonen M. 1997. Tyylikielet ja niiden soveltaminen lainsäädäntöasiakirjoihin. Jyväskylän yliopisto, Tietojärjestelmätieteen pro gradu -tutkielma.
- Holman G. K. 1999. What´s the big deal with XSL [online]. XML.com [Viitattu 11.7.2002]. Saatavilla WWW-muodossa <<http://www.xml.com/lpt/a/1999/04/holman/xsl.html>>.
- Holman G. K. 2002. Practical formatting using XSLFO (Extensible Stylesheet Language Formatting Objects) [online]. Kars: Crane Softwrights Ltd. [Viitattu 30.9.2002]. Saatavilla PDF-muodossa <<http://www.cranesoftwrights.com/training/#pfux-dl>>.
- Honkaranta A., Lyytikäinen V. & Tiitinen P. 2001. Experiences of document transformation with XSLT and DOM. *Proceedings of XML Scandinavia, Kööpenhamina, Tanska, 5-7.11.2001. XML Users Group Denmark*, 52-59. Saatavilla WWW-muodossa <URL:<http://www.cc.jyu.fi/~ankarjal/Experiences.html>>.
- Honkaranta A. & Tyrväinen P. 2001. Possibilities and constraints for managing and reusing information content of structured documents:

the case of operation and maintenance manuals. Teoksessa Proceedings of IRIS 24, Information Systems Research Seminar in Scandinavia, Ulvik in Hardanger, Norja, 11-14.8.2001. University of Bergen, Department of Information Science, Norja, 381-396. Saatavilla PDF-muodossa <URL:<http://www.cc.jyu.fi/~ankarjal/julkaisut.html>>.

ISO 8879.1986. International Organization for Standardization, Information Processing - text and office systems - Standard Generalized Markup language (SGML).

ISO 10179.1996. International Organization for Standardization, Information technology - Text and office systems - Document Style Semantics and Specification Language (DSSSL) [online]. [Viitattu 8.10.2002]. Saatavilla WWW-muodossa <URI:<http://www.ibiblio.org/pub/sun-info/standards/dsssl/draft/>>.

Kay M. 2001. XSLT Programmer's Reference. Chigaco: Wrox Press Inc.

Kuikka E. 1996. Processing of Structured Documents Using a Syntax-Directed Approach. PhD Thesis, Kuopion yliopiston julkaisuja C, Luonnontieteet ja ympäristötieteet 53. Kuopion yliopisto.

Lamming M., Eldridge M., Flynn M., Jones C. & Pendlebury D. 2000. Satchel: providing access to any document, any time, anywhere. ACM Transactions on Computer-Human Interaction 7(3), 322-352.

Le Hors A., Le Hégarret P., Wood L. ym. (toim.) 2000. Document Object model (DOM) Taso 2 Versio 1.0 [online]. W3-järjestön suositus 13.11.2000. [Viitattu 23.9.2002]. Saatavilla WWW-muodossa <URL:<http://www.w3.org/TR/DOM-Level-2-Core/>>.

Lie H. W. & Bos B. 1996. Cascading Style Sheets (CSS) Taso 1 [online]. W3-järjestön suositus 17.12.1996. [Viitattu 1.10.2002] Saatavilla WWW-muodossa <URL:<http://www.w3.org/TR/REC-CSS1/>>.

Lie H. W. & Bos B. 1998. Using XSL and CSS together [online]. W3C Note, Syyskuu 1998 [Viitattu 3.3.2002]. Saatavilla WWW-muodossa <URL:<http://www.w3.org/TR/NOTE-XSL-and-CSS>>.

Lie H. W. & Saarela J. 1999. Multipurpose web publishing using HTML, XML and CSS. Communications of the ACM 42(10), 95-101.

- Lindén G. 1997. Structured Document Transformations. PhD Thesis, Series of Publications A, Report A-1997-2. Helsingin yliopisto, Tietojenkäsittelytieteen laitos.
- Liu Z. & Stork D. G. 2000. Is paperless really more? Rethinking the role of paper in the digital age. *Communications of the ACM* 43(11), 94-97.
- Lyytikäinen V., Päivärinta T., Salminen A. & Tiitinen P. 1997. Valtion talousarvioon liittyvien asiakirjojen rakenteistaminen [online]. RASKE-projektin raportti. Helsinki: Eduskunta. [Viitattu 28.10.2002]. Saatavilla PDF-muodossa <URL:<http://www.cs.jyu.fi/~airi/raportit/RASKE-talousarvioraportti.pdf>>.
- Lyytikäinen V. & Tiitinen P. 2001. Usability of structured documentation. Methods of evaluation. Jyväskylä: Jyväskylän yliopisto.
- Meehan J., Taft E., Chernicoff S. & Rose C. (toim.). 2000. PDF Reference Second Edition, Adobe Portable Document Format Version 1.3 [online]. Boston: Addison-Wesley. [Viitattu 12.10.2002] Saatavilla PDF-muodossa <URL:<http://partners.adobe.com/asn/developer/acrosdk/docs/filefmtspecs/PDFReference13.pdf>>.
- Neugebauer F. 2002. XSL formatting objects: here today, huge tomorrow. Is XL-FO the “next big thing” to present XML data? [online]. *XML Journal* www.XML-Journal.com 3(1) [Viitattu 28.1.2002] Saatavilla WWW-muodossa <URL:<http://www.sys-con.com/xml/articleprint.cfm?id=324>>.
- van Ossenbruggen J., Geurts J., Cornelissen F., Hardman L. & Rutledge L. 2001. Towards second and third generation web-based multimedia. Teoksessa *Proceedings of The Tenth International World Wide Web Conference on World Wide Web, Hong Kong, Hong Kong*. New York: ACM Press, 479-488.
- Pawson D. 2001. XSL-FO an introduction to XSL formatting objects [online]. Dava Pawson 2001. [Viitattu 11.3.2002]. Saatavilla WWW-muodossa <URL:<http://dpawson.co.uk/xsl/sect3/bk/>>.

- Pemberton S., Austin D., Axelsson J. ym. (toim.). 2000. The Extensible HyperText Markup Language (Second Edition) A Reformulation of HTML 4 in XML 1.0 [online]. W3-järjestön suositus 26.1.2000. [Viitattu 23.12.2002]. Saatavilla WWW-muodossa <URL:<http://www.w3.org/TR/2002/REC-xhtml1-20020801/>>.
- Phelps T. A. & Wilensky R. 2001. The multivalent browser: A platform for new ideas. Teoksessa Proceedings of Document Engineering 2001, 9-10.11.2001, Atlanta, Georgia, USA. [Viitattu 8.10.2002]. Saatavilla PDF-muodossa <URL:<http://http.cs.berkeley.edu/~phelps/Multivalent/papers/doceng2001.pdf>>.
- Päivärinta T. 2000. Organisaation dokumenttien hallinta ja sen kehittäminen: periaatteet, osa-alueet ja lähtökohdat. Jyväskylä: Tietotekniikan tutkimusinstituutti, Jyväskylän yliopisto.
- Päivärinta T. & Peltola T. 2001. Engineering of a genre-based method for developing electronic document management: The consultant's viewpoint. Teoksessa J. Krogstie, K. Siau, T. Halpin (toim.) Proceedings of the Sixth EMMSAD, Interlaken, Switzerland, June 4-5.
- Päivärinta T. & Tyrväinen P. 2001. Documents in information management: diverging connotations of "a document" in digital era. Teoksessa Proceedings of IRMA '98, Boston, 1998, 163-173.
- Reid G. C. 1990. Thinking in PostScript. Boston: Addison-Wesley. [Viitattu 8.10.2002]. Saatavilla PDF-muodossa <URL:<http://www.rightbrain.com/download/books/ThinkingInPostScript.pdf>>.
- RTF. 1999. Rich Text Format (RTF) Specification, versio 1.6 [online]. Microsoft Corporation. [Viitattu 8.10.2002]. Saatavilla WWW-muodossa <URL:<http://msdn.microsoft.com/library/?url=/library/en-us/dnrftspec/html/rftspec.asp?frame=true>>
- Salminen A. 1992. Rakenteisen tekstin hallinta. Jyväskylä: Jyväskylän yliopisto, Tietojenkäsittelyopin laitos.
- Salminen A. 1994. Reflections on EDIFACT: Seven issues for the future international EDI message standardization. Teoksessa W. R. J. Baets (toim.) Proceedings of the Second European Conference on Information Systems PART III. Breukelen: Nijenrode University Press, 631-642.

- Salminen A. 2000. Methodology for document analysis. Teoksessa Encyclopedia of Library and Information Science. New York, Basel: Marcel Dekker, 299-320.
- Salminen A., Lyytikäinen V. & Tiitinen P. 2000. Putting documents into their work context in document analysis. Information Processing and Management 36 (2000) 623-641.
- Salminen A., Lyytikäinen V., Tiitinen P. & Mustajärvi O. 2001. Experience of Finnish SGML standardization: The case of the Finnish legislative documents. Teoksessa R.H.Sprague, Jr. (toim.) Proceedings of the Thirty-Fourth Hawaii International Conference on System Science. Los Alamitos, CA: IEEE Computer Society.
- SAX. 2002. [Viitattu 14.10.2002]. <http://www.saxproject.org/>
- Stonebraker M. & Hellerstein J. M. 2001. Content integration for e-business. Teoksessa Proceedings of the 2001 ACM SIGMOD International Conference on Management of Data, Santa Barbara, California, USA, 21-24.5.2001. New York: ACM Press, 552-560.
- TeX Users Group. 2002. [Viitattu 14.10.2002]. <http://www.tug.org/>
- TIFF. 1992. Revision 6.0. Mountain View, CA: Adobe Systems Inc. [Viitattu 8.10.2002]. Saatavilla PDF-muodossa <URL:<http://partners.adobe.com/asn/developer/pdfs/tn/TIFF6.pdf>>
- Tsalgatidou A. 1999. Re-engineering and automation of business processes: criteria for selecting supporting tools. Teoksessa A. Dogac, M.T. Özsu & O. Ulusoy (toim.) Current Trends in Data Management Technology. Hershey: Idea Group Publishing, 59-67.
- Tyrväinen P. & Päivärinta T. 1998. Analyzing document genres for smoothening the IT integration in global organization networks. Teoksessa Roche E. M., Kangas K. & Suomi R. (toim.) MIS – shaped or misshaped? Emerging Issues in the New Global Arena. Proceedings of the IFIP WG 8.7 Working Conference, Helsinki, Joulukuu 1998. 101-113.

Tyrväinen P. 2002. Mikä on dokumenttien hallinnan konkreettinen merkitys liiketoiminnassa? Dokumenttien hallinta -seminaari, Katajanokan kasino, Helsinki 6.-7.3.

Weiss, A. 1999. XML gets down to business. NetWorker 3(3), 36-43.

Wiggins, B. 2000. Effective Document Management: unlocking corporate knowledge. Aldershot: Gower.

Tuotteet ja palvelut:

Adobe Acrobat Reader 5. [Viitattu 14.10.2002]. <URL:<http://www.adobe.com/products/acrobat/readstep2.html>>.

Anilinker. [Viitattu 29.12.2002]. <URL:<http://www.anilinker.fi/>>.

Balise. 2000. Balise Documentation Release 4 [online]. [Viitattu 14.10.2002] <URL:<http://balise.xoasis.com/doc/doc.htm>>.

Ball S. 2001. XSLT Standard Library. Version 1.1. [online]. [Viitattu 14.10.2002]. Saatavilla WWW-muodossa <URL:<http://xslt.sourceforge.net/>>

BasWare. [Viitattu 11.11.2002]. <URL:<http://www.basware.fi/>>.

Elisa. [Viitattu 11.11.2002]. <URL:<http://www.elisa.fi/>>.

ELMA. [Viitattu 11.11.2002]. <URL:<http://www.elma.fi/fi/tuotteet/services/f-laskutus.html>>

EXSLT. [Viitattu 14.10.2002]. <URL:<http://www.exslt.org/index.html>>

FOA. [Viitattu 11.11.2002]. <URL:<http://foa.sourceforge.net/>>

FOP. [Viitattu 11.11.2002]. <URL:<http://xml.apache.org/fop/>>

Omnimark. 2000. Guide to Omnimark 6 [online]. Ottawa: OmniMark Technologies Corporation 28.6.2002. [Viitattu 1.10.2002]. Saatavilla WWW-muodossa <URL:<http://www.omnimark.com/documentation/docs/html/concept/603.htm>>.

Suomen Posti Oyj. [Viitattu 11.11.2002]. <URL:<http://www.posti.fi/>>

TietoEnator. [Viitattu 11.11.2002]. <URL:<http://www.rm.tietoenator.com>>

Verkkolaskukonsortio. [Viitattu 11.11.2002]. <URL:<http://www.invoiceconsortium.com>>

X-Smiles. [Viitattu 1.10.2002]. <URL:<http://www.xsmiles.org/>>

XMLSoftware. [Viitattu 11.11.2002]. <URL:<http://www.xmlsoftware.com/xslfo.html>>

TUTKIELMASSA KÄYTETYT LYHENTEET

ASCII	American Standard Code for Information Interchange
CSS	Cascading Style Sheet Language
DOM	Document Object Model
DSSSL	Document Style Semantics and Specification Language
DTD	Document Type Definition
EDI	Electronic Document Interchange
HTML	Hypertext Markup Language
OVT	Organisaatioiden välinen tiedonsiirto
PCL	Printer Command Language
PDF	Portable Document Format
RTF	Rich Text Format
SAX	Simple API for XML Parsing
SGML	Standard Generalized Markup Language
SVG	Scalable Vector Graphic
TIFF	Tagged Image File Format
WWW	World Wide Web
XHTML	eXtensible Hypertext Markup Language
XML	Extensible Markup Language
XPath	XML Path Language
XSL	Extensible Stylesheet Language
XSLT	XSL Transformations

LIITE 1: LASKUN XML-MUOTOINEN MALLIDOKUMENTTI

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<INVOICE>
<HEADER>
<INVOICE_ID>706198</INVOICE_ID>
<PROCESS_CODE>00</PROCESS_CODE>
<INVOICE_TYPE>07</INVOICE_TYPE>
<METHOD_OF_CHARGE>00</METHOD_OF_CHARGE>
<INVOICE_DATE>
<DATE>
<DAY>25</DAY>
<MONTH>01</MONTH>
<CENTURY>20</CENTURY>
<DECADE_AND_YEAR>01</DECADE_AND_YEAR>
</DATE>
</INVOICE_DATE>
<DUE_DATE>
<DATE>
<DAY>08</DAY>
<MONTH>02</MONTH>
<CENTURY>20</CENTURY>
<DECADE_AND_YEAR>01</DECADE_AND_YEAR>
</DATE>
</DUE_DATE>
<VALUE_DATE>
<DATE>
<DAY>25</DAY>
<MONTH>01</MONTH>
<CENTURY>20</CENTURY>
<DECADE_AND_YEAR>01</DECADE_AND_YEAR>
</DATE>
</VALUE_DATE>
<CASH_DISCOUNT>
<PER_CENT>0.00</PER_CENT>
<CASH_DISCOUNT_AMOUNT>
<AMOUNT SIGN="+">0.00</AMOUNT>
```

(jatkuu)

```
</CASH_DISCOUNT_AMOUNT>
<FREE_TEXT>14 PV NETTO</FREE_TEXT>
</CASH_DISCOUNT>
<PAYMENT_OVERDUE_FINE>
<INTEREST_RATE>13.00</INTEREST_RATE>
</PAYMENT_OVERDUE_FINE>
<CURRENCY>
<CODE>FIM</CODE>
<RATE>5.945730</RATE>
</CURRENCY>
<ORDER_REFERENCE>MARI MALLI</ORDER_REFERENCE>
<LINKS>
<LINK>www.malliyritys.com</LINK>
</LINKS>
<FREE_TEXT>Testi 27.06.2001</FREE_TEXT>
</HEADER>
<PAYEE>
<CUSTOMER_INFORMATION>
<CUSTOMER_NAME>MALLIYRITYS OYJ</CUSTOMER_NAME>
<ADDRESS>
<STREET_ADDRESS1>PL 101</STREET_ADDRESS1>
<STREET_ADDRESS2>MALLITIE 6</STREET_ADDRESS2>
<POSTAL_CODE>00001</POSTAL_CODE>
<POST_OFFICE>MALLILA</POST_OFFICE>
<COUNTRY/>
<COUNTRY_CODE>FI</COUNTRY_CODE>
</ADDRESS>
<VAT_NUMBER>FI012345678</VAT_NUMBER>
<ORGANIZATION_NUMBER>01234567-8</ORGANIZATION_NUMBER>
<TRADE_REGISTRY_NUMBER>123.123</TRADE_REGISTRY_NUMBER>
<CONTACT_INFORMATION>
<CONTACT_PERSON>Maija Malli</CONTACT_PERSON>
<TELEPHONE_NUMBER>09-1234 5678</TELEPHONE_NUMBER>
<TELEFAX_NUMBER/>
</CONTACT_INFORMATION>
<EU-COUNTRY>
<EU_COUNTRY_NAME/>
```

(jatkuu)

```
<EU_COUNTRY_CODE>FI</EU_COUNTRY_CODE>
</EU-COUNTRY>
<E-MAIL_ADDRESS>malli.lasku@mallyritys.com</E-MAIL_ADDRESS>
</CUSTOMER_INFORMATION>
<BANKS>
<BANK_CODE>1</BANK_CODE>
<BANK_NAME>PANKKI</BANK_NAME>
<BANK_ACCOUNT_NUMBER>123456-12345</BANK_ACCOUNT_NUMBER>
</BANKS>
<NET_SERVICE_ID>TE00371234567812</NET_SERVICE_ID>
<DETAILS_OF_PAYMENT>
<FI_PAYMENT_REFERENCE>859522706198204</FI_PAYMENT_REFERENCE>
</DETAILS_OF_PAYMENT>
<BANK_BARCODE>
<FI_BANK_BARCODE>122811800068219000734200000085952270619820401020800008
</FI_BANK_BARCODE>
</BANK_BARCODE>
</PAYEE>
<RECEIVER>
<CUSTOMER_INFORMATION>
<CUSTOMER_NAME>ESIMERKKI OYJ</CUSTOMER_NAME>
<ADDRESS>
<STREET_ADDRESS1>PL 1</STREET_ADDRESS1>
<STREET_ADDRESS2>ESIMERKKITIE 71</STREET_ADDRESS2>
<POSTAL_CODE>00001</POSTAL_CODE>
<POST_OFFICE>MALLILA</POST_OFFICE>
</ADDRESS>
<VAT_NUMBER/>
<ORGANIZATION_NUMBER/>
<CONTACT_INFORMATION>
<CONTACT_PERSON>MIKA MALLI</CONTACT_PERSON>
<TELEPHONE_NUMBER>09-987 987</TELEPHONE_NUMBER>
<TELEFAX_NUMBER>09-987 986</TELEFAX_NUMBER>
</CONTACT_INFORMATION>
<E-MAIL_ADDRESS>mika.malli@esimerkki.fi</E-MAIL_ADDRESS>
</CUSTOMER_INFORMATION>
<CUSTOMER_ID>859522</CUSTOMER_ID>
```

(jatkuu)

```
<NET_SERVICE_ID/>
</RECEIVER>
<DELIVERY_PARTY>
<CUSTOMER_INFORMATION>
<CUSTOMER_NAME>ESIMERKKI OYJ</CUSTOMER_NAME>
<ADDRESS>
<STREET_ADDRESS1>PL 1</STREET_ADDRESS1>
<STREET_ADDRESS2>ESIMERKKITIE 71</STREET_ADDRESS2>
<POSTAL_CODE>00001</POSTAL_CODE>
<POST_OFFICE>MALLILA</POST_OFFICE>
</ADDRESS>
<VAT_NUMBER/>
<ORGANIZATION_NUMBER/>
<CONTACT_INFORMATION>
<CONTACT_PERSON>MIKA MALLI</CONTACT_PERSON>
<TELEPHONE_NUMBER>09-987 987</TELEPHONE_NUMBER>
<TELEFAX_NUMBER>09-987 986</TELEFAX_NUMBER>
</CONTACT_INFORMATION>
<E-MAIL_ADDRESS>mika.malli@esimerkki.fi</E-MAIL_ADDRESS>
</CUSTOMER_INFORMATION>
</DELIVERY_PARTY>
<PAYER>
<CUSTOMER_INFORMATION>
<CUSTOMER_NAME>ESIMERKKI OYJ</CUSTOMER_NAME>
<ADDRESS>
<STREET_ADDRESS1>PL 1</STREET_ADDRESS1>
<STREET_ADDRESS2>ESIMERKKITIE 71</STREET_ADDRESS2>
<POSTAL_CODE>00001</POSTAL_CODE>
<POST_OFFICE>MALLILA</POST_OFFICE>
</ADDRESS>
<VAT_NUMBER/>
<ORGANIZATION_NUMBER/>
<CONTACT_INFORMATION>
<CONTACT_PERSON>MIKA MALLI</CONTACT_PERSON>
<TELEPHONE_NUMBER>09-987 987</TELEPHONE_NUMBER>
<TELEFAX_NUMBER>09-987 986</TELEFAX_NUMBER>
</CONTACT_INFORMATION>
```

(jatkuu)

```
<E-MAIL_ADDRESS>mika.malli@esimerkki.fi</E-MAIL_ADDRESS>
</CUSTOMER_INFORMATION>
<NET_SERVICE_ID/>
</PAYER>
<ROWS>
<ROW>
<ARTICLE>
<ARTICLE_ID>802680</ARTICLE_ID>
<ARTICLE_NAME>TUOTE 802680</ARTICLE_NAME>
<ARTICLE_NAME2/>
<DELIVERY_DATE>
<DATE>
<DAY>01</DAY>
<MONTH>01</MONTH>
<CENTURY>20</CENTURY>
<DECADE_AND_YEAR>01</DECADE_AND_YEAR>
</DATE>
</DELIVERY_DATE>
</ARTICLE>
<QUANTITY>
<OFFERED>0</OFFERED>
<ORDERED>0</ORDERED>
<DELIVERED>1</DELIVERED>
<CHARGED>1</CHARGED>
</QUANTITY>
<ORDER_NUMBER>17713</ORDER_NUMBER>
<ORDER_DATE>
<DATE>
<DAY>01</DAY>
<MONTH>12</MONTH>
<CENTURY>20</CENTURY>
<DECADE_AND_YEAR>00</DECADE_AND_YEAR>
</DATE>
</ORDER_DATE>
<NOTE_NUMBER>585676</NOTE_NUMBER>
<UNIT>KPL</UNIT>
<PRICE_PER_UNIT>
```

(jatkuu)

```
<AMOUNT SIGN="+" VAT="EXCLUDED">601.80000</AMOUNT>
</PRICE_PER_UNIT>
<ROW_TOTAL>
<AMOUNT SIGN="+" VAT="EXCLUDED">601.80</AMOUNT>
</ROW_TOTAL>
<ROW_AMOUNT>
<AMOUNT SIGN="+" VAT="EXCLUDED">601.80</AMOUNT>
</ROW_AMOUNT>
<VAT>
<RATE>22.00</RATE>
<VAT_AMOUNT>
<AMOUNT SIGN="+">132.41</AMOUNT>
</VAT_AMOUNT>
<FREE_TEXT>+ ALV</FREE_TEXT>
</VAT>
<CURRENCY_INFORMATION>
<FIXED_PRICE_PER_UNIT>
<AMOUNT SIGN="+" VAT="EXCLUDED">0.00000</AMOUNT>
</FIXED_PRICE_PER_UNIT>
<FIXED_EXCHANGE_RATE>0.000000</FIXED_EXCHANGE_RATE>
<INVOICE_DATE_RATE>5.945730</INVOICE_DATE_RATE>
</CURRENCY_INFORMATION>
</ROW>
</ROWS>
<SUMMARY>
<ROWS_TOTAL>
<AMOUNT SIGN="+" VAT="EXCLUDED">601.80</AMOUNT>
</ROWS_TOTAL>
<INVOICE_TOTAL>
<AMOUNT SIGN="+" VAT="INCLUDED">734.20</AMOUNT>
</INVOICE_TOTAL>
<INSURANCE_TOTAL>
<AMOUNT SIGN="+" VAT="EXCLUDED">0.00</AMOUNT>
</INSURANCE_TOTAL>
<VAT_SUMMARY>
<RATE>22.00</RATE>
<ACCORDING>
```

(jatkuu)

```
<AMOUNT SIGN="+">601.83</AMOUNT>
</ACCORDING>
<VAT_RATE_TOTAL>
<AMOUNT SIGN="+">132.41</AMOUNT>
</VAT_RATE_TOTAL>
<DEFAULT_VAT_RATE_ACCOUNT/>
</VAT_SUMMARY>
<VAT_TOTAL>
<AMOUNT SIGN="+">132.41</AMOUNT>
</VAT_TOTAL>
<ROUNDINGS>0.01</ROUNDINGS>
</SUMMARY>
</INVOICE>
```

LIITE 2: XSL-TYYLISIVUESIMERKKI

```

<?xml version="1.0"?>
<xsl:stylesheet xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
                version="1.0">

  <!-- tulospuun muodon ja merkkikoodauksen määrittely: -->
  <xsl:output method="xml" encoding="iso-8859-1"/>

  <!-- XML-dokumentin juurielementtiin täsmäävä mallipohja: -->
  <xsl:template match="/">

    <!-- dokumentin esityksen yleisten asetustelujen määrittely: -->
    <!-- tulospuun juurielementti: -->
    <fo:root xmlns:fo="http://www.w3.org/1999/XSL/Format">

      <!-- sivutyyppeiden määrittely: -->
      <fo:layout-master-set>

        <!-- tässä tyyllisivussa vain yksi sivutyyppi: -->
        <fo:simple-page-master master-name="sisasivu"
          page-width="50mm" page-height="50mm"
          margin-left="10mm" margin-right="10mm"
          margin-top="5mm" margin-bottom="10mm">
          <fo:region-body region-name="mallisivunRunko"/>
        </fo:simple-page-master>

      </fo:layout-master-set>

      <!-- sivujakson määrittely, viittaus sivutyyppiin: -->
      <fo:page-sequence master-reference="sisasivu">
        <xsl:apply-templates select="INVOICE/HEADER"/>
      </fo:page-sequence>

    </fo:root>
  </xsl:template>

```

(jatkuu)


```
<!-- HEADER-elementtiin täsmävä mallipohja: -->

<xsl:template match="HEADER">

  <!-- lohkovuon aloitus: -->
  <fo:flow flow-name="mallisivunRunko">

    <!-- ensimmäinen lohko liitettävälle vakiotekstille: -->
    <fo:block space-before="5mm"
              font-family="sans-serif"
              font-size="8pt"
              font-weight="bold">
      <xsl:text disable-output-escaping="yes">Laskun numero</xsl:text>
    </fo:block>

    <!-- toinen lohko XML-dokumentista haettavalle tietosisällölle: -->
    <fo:block font-family="sans-serif"
              font-size="10pt"
              font-weight="normal">
      <xsl:value-of select="INVOICE_ID"/>
    </fo:block>

  </fo:flow>

</xsl:template>

</xsl:stylesheet>
```