

Olli Kauppinen

ODF- ja OOXML-toimisto-ohjelmaformaattien vertailu

Tietotekniikan
kandidaatintutkielma
22. helmikuuta 2012

Jyväskylän yliopisto

Tietotekniikan laitos

Jyväskylä

Tekijä: Olli Kauppinen

Yhteystiedot: olli.kauppinen@jyu.fi

Työn nimi: ODF- ja OOXML-toimisto-ohjelmaformaattien vertailu

Title in English: Comparison of ODF and OOXML Office Formats

Työ: Tietotekniikan kandidaatintutkielma

Sivumäärä: 54

Tiivistelmä: Tutkielma esittelee ja vertailee toimisto-ohjelmien tiedostoformaatteja. Tiedostoformaateiksi valittiin ODF ja OOXML, jotka ovat kumpikin XML-merkkaukieleen pohjautuvia kansainvälisiä standardeja. Tutkielma esittelee lyhyesti myös XML-merkkaukieleen ja sen tuomat edut dokumenttien tallentamiseen. Tiedostoformaattien esittely ja vertailu painottuu tekstidokumentteihin. Vertailun pohjana käytetään valmiita tutkimuksia ja tutkielman kirjoittajan omaa kokemusta kyseisten tiedostoformaattien tulkinnasta. Vertailun pohjana ovat tiedostoformaattien selkeys, pitkäaikaisuus ja pysyvyys sekä tietoturva. Tutkielman lopputuloksena saatiin tiedostoformaatin valintaan vaikuttavia seikkoja ja suosituksia. Tiedostoformaattit eivät ole vielä lopullisessa muodossa, joten isoihin muutoksiin ei kannata vielä ryhtyä.

Abstract: The thesis examines and compares the file formats of office suites. Chosen file formats are ODF and OOXML, which are international standards and based on XML markup language. The thesis examines XML markup language and benefits of using XML based file formats in documents. File formats examination and comparison are focused in word documents. Existing studies and writers own experience in analysing the file formats are used in the comparison. The file formats clearness, long-term usability, availability and security are compared. Good notices for file formats selection was found in thesis. It was perceived that file formats are not completed yet and people shouldn't have a hurry with the selection.

Avainsanat: ODF, OOXML, XML, vertailu, toimisto-ohjelma, XML-perustainen, tiedostoformaatti, tekstinkäsittely

Keywords: ODF, OOXML, XML, comparison, office suite, XML-based, file format, word processing

Sisältö

1	Johdanto	1
2	Toimisto-ohjelmien ja tiedostoformaattien kehitys	2
2.1	Toimisto-ohjelmien kehitys	2
2.2	Rakenteiset ja avoimet dokumentit	3
3	XML-merkkaukieli	6
4	ODF-tiedostoformaatti	9
4.1	Kehitys	9
4.2	ODF-tiedostoformaattia käyttävät ohjelmistot	10
4.3	Tiedostopäätteet	10
4.4	Tiedoston rakenne	11
4.4.1	Content.xml-tiedoston sisältö	12
4.4.2	Styles.xml-tiedoston sisältö	13
4.4.3	Muiden tiedostojen ja hakemistojen sisältö	15
4.5	Hyödyt ja vahvuudet	16
4.6	Rajoitukset ja heikkoudet	16
4.7	Käsittely ohjelmointikielillä ja työkaluilla	17
5	OOXML-tiedostoformaatti	18
5.1	Kehitys	18
5.2	OOXML-tiedostoformaattia käyttävät ohjelmistot	18
5.3	Tiedostopäätteet	19
5.4	Tiedoston rakenne	19
5.4.1	Document.xml-tiedoston sisältö	21
5.4.2	Styles.xml-tiedoston sisältö	22
5.4.3	Muiden tiedostojen ja hakemistojen sisältö	22
5.5	Hyödyt ja vahvuudet	23
5.6	Rajoitukset ja heikkoudet	24
5.7	Käsittely ohjelmointikielillä ja työkaluilla	25
6	Tutkimusmenetelmät	26

7	Vertailu	28
7.1	Kilpailutilanne	28
7.2	Jo tehdyt vertailut	30
7.2.1	Tiedostoformaattien selkeys	30
7.2.2	Pitkäaikaisuus ja pysyvyys	32
7.2.3	Tietoturva	33
7.2.4	Tiedostoformaatin valinta	34
7.3	Empiirinen vertailu	36
7.3.1	Tausta	36
7.3.2	Tiedostoformaattien selkeys	37
7.3.3	Pitkäaikaisuus ja pysyvyys	39
7.4	Pohdintaa tiedostoformaattien vertailusta	40
7.5	Tutkimuksen luotettavuuden tarkastelu	42
8	Johtopäätökset ja yhteenveto	44
	Lähteet	46

1 Johdanto

Toimisto-ohjelmat ovat hyvin yleisiä ja käytettyjä ohjelmia. Niitä käytetään erilaisien dokumenttien kuten tekstinkäsittelydokumenttien, laskentataulukoiden ja graafisten esitysten laadintaan. Tässä tutkielmassa käsitteellä dokumentti tarkoitetaan tiedostoa, joka on luotu toimisto-ohjelmaa käyttäen. Esimerkiksi Microsoft Wordilla luotua tekstinkäsittelytiedostoa kutsutaan tutkielmassa dokumentiksi.

Nykyään toimisto-ohjelmien tiedontallennus perustuu yhä useammin XML-pohjaiseen tiedostoformaattiin. XML (Extensible Markup Language) on merkintäkieli, jolla tiedon merkitys on kuvattavissa tiedon sekaan. Sitä käytetään tiedon välitykseen ja rakenteisten dokumenttien formaateissa. Se avulla suuria tietomassoja voidaan jäsentää selkeämmin. XML-kieli kuvaa ainoastaan dokumentin loogisen rakenteen. Dokumentin esityksestä vastaa sovellus, jolla dokumenttia katsellaan, esim. OpenOffice.org.

Tutkielmassa esitellään toimisto-ohjelmien käyttämät yleisimmät tiedostoformaattit. Vaikka toimisto-ohjelmat ovat laajasti käytettyjä ohjelmia ja niillä tallennetaan tärkeitä dokumentteja, silti käyttäjät eivät välttämättä osaa huomioida tiedostoformaatin merkitystä valitessaan toimisto-ohjelmaa. Tiedostoformaatin toimivuus on kuitenkin oleellinen asia yhdessä toimisto-ohjelman toimivuuden kanssa. Tutkielman tarkoitus on esitellä ja vertailla kahta yleisintä tiedostoformaattia. Tutkielman tavoite on selvittää, kumpi tiedostoformaatti soveltuu paremmin dokumenttien laadintaan ja muokkaukseen. Vertailuun käytetään valmiita tutkimuksia sekä kirjoittajan omaa kokemusta. Tutkielma ei vertaile itse toimisto-ohjelmia.

Luvussa 2 kuvataan toimisto-ohjelmien ja tiedostoformaattien kehitystä alkua ajoista nykypäivään sekä esitellään kehitykseen vaikuttavia tekijöitä. Luvussa 3 esitellään XML-merkkäuskieli ja sen merkitys tiedostoformaatteihin. Luvuissa 4 ja 5 esitellään kaksi hallitsevaa tiedostoformaattia. Luvussa 6 esitellään käytetyt tutkimusmenetelmät. Tiedostoformaatteja vertaillaan luvussa 7. Vertailun kohteina ovat tiedostoformaatin selkeys, pitkäaikaisuus ja pysyvyys sekä tietoturva. Luvussa 8 vedetään yhteen havainnot ja pohditaan tiedostoformaatin valintaan vaikuttavia seikkoja.

2 Toimisto-ohjelmien ja tiedostoformaattien kehitys

Luvussa käydään läpi toimisto-ohjelmien kehitys alusta nykypäivään. Toimisto-ohjelmien kehityksessä nykyiseen tilanteeseen Microsoft on ollut hyvin merkittävässä asemassa. Tiedostoformaattit ovat kehittyneet ohjelmistojen mukana ja niiden kehitys on usein sidoksissa jonkin ohjelmiston kehitykseen.

2.1 Toimisto-ohjelmien kehitys

Toimisto-ohjelmien kehitys alkoi 1980-luvulla, jolloin ohjelmia ja tiedostoformaatteja oli useita, kuten WordStar, WordPerfect, MultiMate, Microsoft Word Dos, IBM Display Writer, Visicalc, Lotus 1-2-3 ja Microsoft Multiplan. Nämä ohjelmat ja niiden käyttämät tiedostoformaattit olivat keskenään huonosti yhteensopivia. [13]

Toimisto-ohjelmien markkinat kaventuivat 1990-luvulla muutamaaan hallitsevaan tuotteeseen. Merkittävimpiä olivat Microsoft Office, IBM Lotus SmartSuite, WordPerfect, Microsoft Works ja Borland Office. Näistä Microsoft Office saavutti valta-aseman 1990-luvun aikana. 1990-luvulla tiedostoformaattit olivat omistusoikeudellisesti rajattuja ja suljettuja. Niiden määrittelyt olivat vain ohjelmistoyritysten tiedossa. [13]

Grönroosin ja Karjalaisen [13] mukaan tähän asti dokumentteja pidettiin enemmän henkilökohtaisina. Niitä käytettiin usein samalla koneella ja julkaistiin tulostettuna. Dokumenttien käytössä on kuitenkin siirrytty suuntaan, jossa samoja asiakirjoja muokataan eri henkilöiden toimesta, niitä varastoidaan ja jaetaan organisaatioiden välillä. Organisaatioiden laatimat ja varastoimat dokumenttimäärät ovat vahvassa kasvussa. Lisäksi organisaatioiden tallentama tieto on tallennettu dokumentteihin tietokantojen sijaan. Honkarannan [15] mukaan jopa 80–90 prosenttia tallennetusta tiedosta on tallennettu dokumentteihin. Dokumenttien käsittely ja niistä tiedon haku on tietokantoja hitaampaa.

Dokumenttien käytön tarpeen muuntuminen henkilökohtaisesta organisaatiokohtaiseksi ja dokumenttien määrän kasvu aiheuttivat sen, että tarvittiin avoin standardoitu tiedostoformaatti. Avoimen asiakirja-arkkitehtuurin (ODA, Open Document Architecture) määrittely aloitettiin 1980-luvulla, mutta vasta vuonna 1999 julkaistiin kansainvälinen standardi ISO 8613. ODA oli liian monimutkainen, eikä sitä tukevia välineitä tullut käyttöön. Vuonna 2004 EU:n toimesta julkaistu

Valoris-raportti ja sen tulokset antoivat uutta vauhtia avointen tiedostoformaattien kehitykselle. Raportin yhteydessä annettiin suositukset avoimien ja standardoitujen XML-pohjaisten tiedostoformaattien käyttöön julkishallinnossa ja näin ollen välttämään omistusoikeudellisia tiedostomuotoja. [13]

2000-luvun alkupuolella alettiin kehittää XML-pohjaisia tiedostoformaatteja laajemmin. Microsoft käytti jo Office XP:ssa XML:ää apuna. Myös OpenOfficen kehitys alkoi 2000-luvun alkupuolella, kun Sun Microsystems osti StarOffice-paketin StarDivisionilta. 2000-luvun puolivälissä julkaistiin ODF-tiedostoformaatti Sun Microsystemsin ja hieman myöhemmin OOXML-tiedostoformaatti Microsoftin toimesta. [17], [30]

ODF- ja OOXML-tiedostoformaattit ovat nykypäivän markkinoita hallitsevat standardit. Tutkielmassa esitellään ja verrataan näitä kahta toisiinsa. On olemassa myös kolmas avoin XML-pohjainen UOF-tiedostoformaatti (Uniform Office Format, myös Unified Office Format), joka on Kiinassa kehitetty ja sillä on siellä runsas kannattajajoukko [2]. Kiinan julkishallinto edistää UOF-tiedostoformaatin käyttöönottoa. Tutkielmassa ei kuitenkaan käydä UOF-tiedostoformaattia sen tarkemmin läpi, koska se ei ole kansainvälinen standardi ja sen käyttö on länsimaissa vähäistä.

2.2 Rakenteiset ja avoimet dokumentit

Käyttäjien ja organisaatioiden tarpeiden muutoksen takia nykyään tiedostoformaatin pitää olla pysyvä ja luotettava. Dokumentti täytyy saada auki samanlaisena aina. Suljetut tiedostomuodot eivät pysty takamaan pysyvyyttä, koska ne ovat yhden toimijan takana. Jos toimija lopettaa ohjelmiston tuottamisen, työskentely dokumenttien kanssa vaikeutuu ja jossain kohtaan loppuu kokonaan. [7], [13]

Suljettujen tiedostoformaattien korvaajaksi on tullut useita avoimia tiedostoformaatteja. Nämä avoimet tiedostoformaattit perustuvat yleensä XML-merkkauskielen. XML-merkkauskielen perustuvia dokumentteja sanotaan rakenteisiksi dokumenteiksi. [7], [13]

Ditchin [7] mukaan avoin ja rakenteinen tiedostomuoto mahdollistaa eri ohjelmistojen välisen yhteensopivuuden. Niitä voidaan myös muokata muilla ohjelmilla kuin toimisto-ohjelmilla. Tämä mahdollistaa dokumenttien automatisoidun prosessoinnin. Automatisoidussa prosessoinnissa voidaan dokumentteihin tehdä esimerkiksi tarkastuksia, lisätä organisaation logo tai etsiä niistä jotain sisältöä. Rakenteisiin dokumentteihin voidaan myös sisällyttää metadatasia. Metadata on tietoa tiedosta, eli se sisältää tietoja laaditusta dokumentista. Metadataan tallennetaan yleensä dokumentin laadinta- ja muokkauspäivät, laatijan tiedot, avainsanat, organisaation

nimi sekä muita dokumentin tunnistamiseen ja hallintaan tärkeitä tietoja. Salmisen [27] mukaan dokumenttien metadata on tiedon haun ja dokumenttien hallinnan kannalta merkittävää.

Rakenteisten dokumenttien käyttö mahdollistaa Salmisen [27] ja Suhosen [29] mukaan dokumenttien yhtenäisyyden ja virheettömyyden, mahdollisuuden tiedonhakuun, tiedon uudelleenkäyttöön, monikanavajulkaisuun ja ohjelmistoriippumattomuuteen sekä pitkäaikaisen pääsyn dokumentin sisältöön.

Avoimet ja rakenteiset dokumentit saadaan näin ollen aina tulkittua, vaikka dokumentin laadintaan käytettyä ohjelmistoa ei olisikaan. Dokumenttien pysyvyys on usealle käyttäjälle tärkeä ominaisuus. Esimerkkinä voidaan mainita julkishallinto, jonka dokumenttien arkistointi ja pysyvä saatavuus ovat ehdottoman tärkeitä. Euroopan komission alainen IDABC kehittää julkishallinnon organisaatioiden välisiä sähköisen tiedonvaihdon edellytyksiä sekä kansalaisille ja yrityksille suunnattujen palvelujen yhteentoimivuutta. IDABC:n suositukset joulukuulta 2006 eli ns. PEGSCO-suositukset sisältävät suositukset avointen tiedostoformaattien käytöstä julkishallinnossa. Suositusten mukaan julkishallinnon pitäisi käyttää hyväksyen kansainvälisesti standardoituja avoimia tiedostoformaatteja sekä käyttää ainoastaan monien ohjelmatuotteiden tukemia tiedostoformaatteja ja välttää tietyn tuotteen käyttöpakkoa. Mikäli käytetään suljettua omistusoikeudellista tiedostoformaattia, pitäisi tarjota myös vaihtoehtona kansainvälisesti standardoitu avoin tiedostomuoto. [13]

PEGSCO-suositukset sisältävät myös ohjeistuksia teollisuuteen ja ei-julkishallinnollisiin organisaatioihin. Suositusten mukaan toimijoiden pitää pyrkiä kohti yhtä kansainvälistä kaikille avointa dokumenttistandardia. Toimijoiden tuottamien ohjelmistojen täytyy tukea asiaankuuluvia kansainvälisiä standardeja ja olla laajasti yhteensopivia. Toimijoiden on myös osallistua nykyisten standardien kehitykseen ja edistettävä niiden asemaa. [7]

Tiedostoformaatin on toteutettava tietyt vaatimukset, jotta siitä voidaan käyttää termiä avoin. Avoin tiedostoformaatti on kansainvälisen standardointijärjestön standardoima ja sen dokumentaation on oltava kaikkien saatavilla, eikä sen käytöstä saa koitua kohtuuttomia kustannuksia. Tiedostoformaatti on vapaasti ja veloittamatta käytettävissä ilman omistusoikeudellisia rajoitteita käyttäjien omissa sovelluksissa. Myös tiedostoformaatin kehityksen on oltava avointa, jolloin keskeiset käyttäjät, asiantuntijat ja ohjelmistovalmistajat voivat osallistua kehitykseen. Usein myös riippumattomuus käyttöjärjestelmästä luetaan avoimuuden vaatimukseksi. [7], [13]

Avoimen tiedostoformaatin määrittäykset täytyvät ODF-tiedostoformaattissa (ks. luku 4). OOXML-tiedostoformaatti (ks. luku 5) ei täytä kaikkia avoimuuden määri-

tyksiä, koska sen kehitys on yhden organisaation varassa. Microsoft on myös rajoittanut tiedostoformaatin käyttöä käyttäjien omissa sovelluksissa. Tiedostoformaatin dokumentaatio ja määrittely ovat kuitenkin kaikkien halukkaiden saatavilla, joten myös OOXML-tiedostojen tulkinta onnistuu aina.

3 XML-merkkauskieli

XML (Extensible Markup Language) on yksinkertainen ja joustava merkintäkieli. Joustavuutta tuo laajennettavuus, järjestelmäriippumattomuus sekä tuki lokalisointiin ja kansainvälistämiseen. Se koostuu joukosta sääntöjä rakenteisen tiedon esittämiseen tarkoitettujen tekstiformaattien suunnitteluun. Tietokoneiden on helppoa tuottaa ja tulkita tietoa täsmällisessä muodossa XML:n avulla. XML:n on kehittänyt ja standardoinut W3C (World Wide Web Consortium). Sitä käytetään tiedonvälitykseen järjestelmien välillä sekä dokumenttien tallentamiseen. XML-kieli on rakenteellinen kuvauskieli, eli siinä tiedon merkitys on kuvattavissa tiedon sekaan. Dokumentissa kuvataan looginen rakenne, mutta siinä ei oteta kantaa miten rakenteen eri osat tulee esittää. XML-kieli on SGML-kielen yksinkertaistettu osajoukko. XML:n kehityksessä omaksuttiin SGML:n parhaat puolet, hyödyntäen toisen SGML-kieleen perustuvan HTML:n käytöstä kertyneitä kokemuksia. Lopputuloksena oli huomattavasti säännöllisempi ja helppokäyttöisempi kieli. [35]

XML-dokumentit ovat tekstimuotoisia. Yleensä tekstin tallentamiseen käytetään UTF-8 -koodausta. XML-dokumentti alkaa aina prologimäärittelyllä, joka sisältää tiedon käytetystä XML-versiosta. Se sisältää usein myös tiedon dokumentin koodauksesta. Myös muita dokumentin kannalta olennaisia tietoja voidaan määrittää prologissa. [35]

Vaikka XML-tiedosto on tekstiä, ei sitä ole tarkoitettu suoraan luettavaksi, vaan tiedostoa käsitellään usein jollain XML-editorilla. Dokumenttien tallennuksen XML-merkkauskieleen perustava toimisto-ohjelmisto voidaan luokitella XML-editoriksi. Ohjelma vastaa tiedon esittämisestä käyttäjälle koostamalla luettavan dokumentin XML-tiedostosta. XML-tiedostoja voidaan tulkita ja käsitellä kaikilla ohjelmointikielillä, joissa on XML-tiedostojen käsittelyn tuki [26]. Ohjelmointikielien tukevat XML-tiedostojen käsittelyä usein DOM- (Document Object Model) tai SAX- (Simple API for XML) ohjelmointirajapinnoilla.

Seuraavassa on kuvattu XML-dokumentin prologin lähdekoodi:

```
<?xml version="1.0" encoding="utf-8"?>
```

Itse dokumentti koostuu elementeistä. Elementit kuvataan tageilla. Tagi koostuu alkumerkistä ja loppumerkistä. Isot ja pienet kirjaimet luetaan eri merkeiksi. Dokumentissa voi olla useita elementtejä ja elementin sisällä voi olla useita elementtejä.

Kullakin elementillä voi olla sisältöä ja useita attribuutteja. XML 1.0 -spesifikaatio määrittää mitä tagit ja attribuutit ovat. [35]

Seuraavassa on kuvattu XML-dokumentin rakenteen lähdekoodi:

```
<elementti attribuutti="arvo">  
  Elementin sisältö  
</elementti>
```

XML-dokumentin on noudatettava tiettyjä vaatimuksia, jotta se on oikein muodostettu. Mikäli dokumentti täyttää kaikki syntaksisäännöt, eli lopetuselementit ovat olemassa ja oikeassa järjestyksessä, sanotaan dokumentin olevan hyvin muodostettu (*Well-formed*). Mikäli dokumentti noudattaa jonkun dokumenttityypin määrittämiä, on se validi (*Valid*) dokumentti. Määrittelyn mukaan hyvin muodostettu dokumentti sisältää yhden ja vain yhden juurielementin, ei-tyhjillä elementeillä on sekä alku- että loppumerkki, jokainen attribuutti on lainausmerkkien sisällä sekä elementit eivät saa mennä ristiin toistensa kanssa. [35]

XML-merkkauskielessä käytetään nimiavaruuksia yhteensopivuuden parantamiseksi. Mikäli kahta formaatiltaan erilaista XML-tiedostoa halutaan tulkita samanaikaisesti, tulee ongelmaksi elementtien mahdolliset samat nimet ilman nimiavaruutta. Nimiavaruuden tarkoitus on yksilöidä tietty elementin nimi tiettyyn yhteyteen. Nimiavaruus, johon elementti kuuluu, määritetään ennen elementin nimeä. Ne erotetaan toisistaan käyttämällä kaksoispistettä. [35]

Seuraavassa on kuvattu XML-dokumentin nimiavaruuden määrittämisen lähdekoodi:

```
<nimiavaruus:elementti>
```

Tutkielman kannalta nimenomaan XML:n kyvykkyys dokumenttien tallentamiseen on kiinnostavaa. Salmisen [27] mukaan XML-dokumenttien rakenteisuus tuo tiedon tallennukseen seuraavia hyötyjä:

- yhtenäisyys ja oikeellisuus,
- tiedonhaun mahdollisuudet,
- ohjelmisto riippumattomuus,
- tiedon uudelleenkäyttö ja
- pitkäaikainen käytettävyys.

Salmisen [27] mukaan metatietojen hyödyntäminen on myös yksi tärkeä ominaisuus, jonka XML:n käyttö tuo tiedon tallentamiseen. Metatietoa tarvitaan etsiessä jotain tiettyä dokumentteja ja niiden sisältämää tietoa. Grönroosin ja Karjalaisen [13] mukaan metatietojen hyödyntämistä haittaa usein toimisto-ohjelmien metatietojen syöttöön käytettävien käyttöliittymien vaikeahko käytettävyys. Osa metatiedoista tulee itsestään toimisto-ohjelmien asetuksista, loput täytyy käyttäjän syöttää itse. Indeksointia tehtäessä epästandardi toteutustapa voi aiheuttaa ongelmia.

Avointen XML-pohjaisten tiedostoformaattien käyttö on hyödyttänyt etenkin kolmansien osapuolten työtä. Erilaisten työkalujen ja ohjelmien tekeminen sekä dokumenttien muokkaaminen niillä on helpottunut huomattavasti. [18]

4 ODF-tiedostoformaatti

ODF (lyhennetty nimestä OASIS Open Document Format for Office Applications, käytetään myös nimeä OpenDocument) on avoimen lähdekoodin dokumenttien tallentamiseen käytettävä tiedostoformaatti. Se perustuu XML-merkintäkieleen. Luvussa esitellään ODF-tiedostoformaatin keskeiset piirteet.

4.1 Kehitys

ODF-standardin kehitti OASIS (Organization for the Advancement of Structured Information Standards). Standardin spesifikaation loi Sun Microsystems ja sen kehitykseen on osallistunut useita eri yrityksiä ja organisaatioita. ODF-tiedostoformaatin kehityksen aloitti StarDivision vuonna 1999 kehitystyöllä, jonka tarkoitus oli kehittää avoin ja yhteentoimiva tiedostoformaatti. Tiedostoformaatin käyttöä ei ollut tarkoitus alusta asti rajata pois muiden valmistajien käytöstä. Sun Microsystems osti samana vuonna StarDivisionin. Se jatkoi tiedostoformaatin kehitystä OpenOffice.org-toimisto-ohjelmistoa varten. Vuonna 2006 ODF-tiedostoformaatti hyväksyttiin ISO-standardiksi (ISO 26300). Se on nykyäänkin avoin tiedostoformaatti ja sen käyttöä ei ole rajoitettu. [23]

Vuonna 2007 ODF-tiedostoformaattista julkaistiin versio 1.1. Uusin, eli 1.2-versio, hyväksyttiin standardiksi 2011 vuoden lopussa ja julkaistiin vuoden 2012 alussa [32]. Versiota 1.1 ei standardoitu, koska se sisälsi vain pieniä muutoksia ja version 1.2 kehitystyö oli jo käynnissä [33]. Versioon 1.2 tuli uusina ominaisuuksina RDF-pohjaiset metatiedot, OpenFormulaan perustuvat taulukkolaskennan kaavat sekä tuki digitaaliselle allekirjoitukselle [33].

ODF-tiedostoformaatti pohjautuu vahvasti muihin olemassa oleviin avoimiin kansainvälisiin standardeihin. Tiedon esityksissä käytetään lukuisia muita ISO-standardeja, kuten kielten koodit (ISO 639), maakoodit (ISO 3155) sekä aikojen ja päivämäärien esitys (ISO 8601). Osa asiakirjan metatiedoista esitetään Dublin Core -metatietostandardin (ISO 15836:2003) alijoukkona. Myös monia W3C-konsortion julkaisemia standardeja käytetään hyväksi. Matemaattiset kaavat esitetään MathML-merkintäkielellä, viittaukset ulkoisiin kohteisiin esitetään XLink-viittauskielellä, animoidut esitykset esitetään SMIL-multimediakielellä, sivu- ja tekstinasetteluun merkitsemiseen käytetään XSL FO -muotoilukieltä sekä lomakkei-

den toteutuksessa käytetään XForms-kieltä ja sen käyttämää XPath-kyselykieltä. Kyseisten standardien mukaisia nimiavaruuksia käytetään elementeissä. [11]

ODF Alliance on järjestö, jonka tarkoitus on edistää ODF-tiedostoformaatin levinneisyyttä ja käyttöä. Sen perustivat IBM ja Sun yhdessä 33 muun yrityksen ja organisaation kanssa vuonna 2006. Nykyään siihen kuuluu yli 500 jäsentä. Sen päätarkoitus on saada julkishallinnot ottamaan ODF-tiedostoformaatin ensisijaiseksi dokumenttien tallennustavaksi. [34]

Standardoinnin jälkeen ODF-tiedostoformaatti on yleistynyt nopeasti organisaatioiden käytössä [13]. Antikaisen [2] tutkimuksen mukaan ODF-tiedostoformaatin käyttö on saanut julkishallinnossa enemmän kannatusta kuin OOXML- tai UOF-tiedostoformaattit. Useissa maissa julkinen sektori on määrätty käyttämään ODF-tiedostoformaattia [12].

4.2 ODF-tiedostoformaattia käyttävät ohjelmistot

Vaikka ODF-tiedostoformaatti kehitettiin alun perin OpenOffice.org-toimisto-ohjelmalle, se ei silti ole ollut mikään yhden ohjelmiston tiedostoformaatti, vaan se on ollut laajasti käytössä. Nykyään ODF-tiedostoformaattia käyttävät useat eri ohjelmistot, kuten OpenOffice.org, LibreOffice, StarOffice, KOffice ja Google Docs. Useat tarjoavat ODF-tiedostoformaattia vaihtoehtoisena tallennusmuotona. Myös Microsoft Office 2010 -toimisto-ohjelmistolla sekä versiolla 2007 SP2-päivityksen jälkeen onnistuu tallentaminen ODF-tiedostoformaattiin. Päivityksen toimivuutta on kuitenkin kritisoitu. [33]

4.3 Tiedostopäätteet

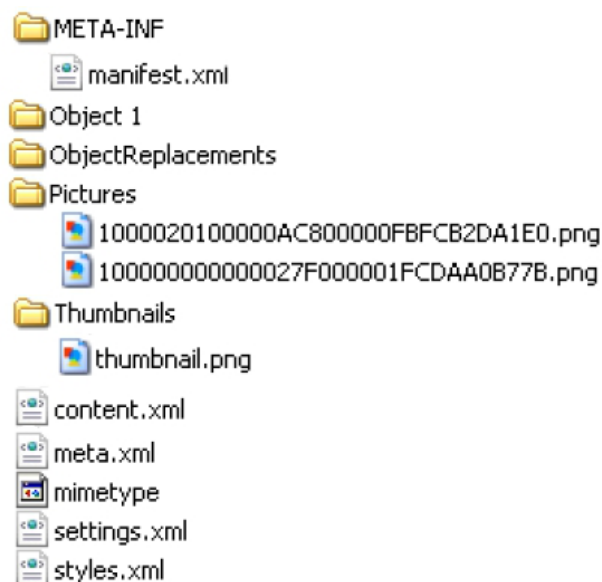
ODF-standardi määrittelee kaikkiaan 16 eri asiakirjatyyppeä. Asiakirjatyypit erotetaan toisistaan käyttämällä kolmen kirjaimen tarkenteita. Taulukossa 4.1 ovat standardin mukaiset tiedostomuodot ja tiedostotarkenteet.

Taulukko 4.1: ODF:n mukaiset tiedostomuodot ja tiedostotarkenteet [11]

Asiakirjatyyppi	Tiedostotarkenne	Mallin tiedostotarkenne
Tekstiasiakirja	.odt	.ott
Laskentataulukko	.ods	.ots
Esite	.odp	.otp
Piirros	.odg	.otg
Kaavio	.odc	.otc
Kaava	.odf	.otf
Kuva	.odi	.oti
Verkkosivu	-	.oth
Perusasiakirja	.odm	-

4.4 Tiedoston rakenne

Dokumentti koostuu useista eri XML-tiedostoista, kansioista, mahdollisista kuvista ja muista tiedostoista. Eri tiedostot on pakattu yhdeksi tiedostoksi käyttämällä ZIP-pakkausta. Tutkielmassa perehdytään tarkemmin tekstinkäsittelydokumentin sisältöön. Puretun tekstinkäsittelydokumentin tiedostorakenne esitetään kuvassa 4.1. [11]



Kuva 4.1: Puretun ODF-tiedostoformaatin tekstinkäsittelydokumentin tiedostorakenne [11]

Dokumentin sisällön suhteen tärkeimmät tiedostot ovat `content.xml` ja `styles.xml`. Tiedosto `content.xml` sisältää suurimman osan dokumentin sisällöstä. Tiedosto `styles.xml` sisältää ulkoasuun liittyvät määrittelyt. [11]

4.4.1 Content.xml-tiedoston sisältö

Tiedosto `content.xml` sisältää dokumentin varsinaisen sisällön. Tiedoston juurielementtinä on `office:document-content`, joka sisältää nimiavausmäärittelyt ja ODF-version määrittelyn. Juurielementin sisällä ovat elementit `office:scripts`, jossa määritellään dokumentin sisältämät makrot, `office:font-face-decls`, joka sisältää tietoa dokumentin luojan käyttämistä fonteista, `office:automaticstyles`, joka sisältää automaattiset tyylit sekä `office:body`, joka on asiakirjan varsinainen runko ja se sisältää dokumenttityypin mukaisen elementin. Dokumenttityypin mukainen elementti on tekstinkäsittelydokumentissa `office:text`. Tämän elementin alla on dokumentin varsinainen sisältö. [11]

Seuraavassa on kuvattu tekstinkäsittelydokumentin oleelliset elementit:

```
<office:document-content office:version="1.2">
  <office:scripts/>
  <office:font-face-decls>
    <style:font-face style:name="Mangal1" svg:font-family="Mangal"/>
    ...
  </office:font-face-decls>
  <office:automatic-styles/>
  <office:body>
    <office:text>
      ...
    </office:text>
  </office:body>
</office:document-content>
```

Dokumentti koostuu kappaleista. Kappale alkaa elementillä `text:p`, jonka sisällä on varsinainen tekstisisältö. Elementin attribuutti `text:style-name` määrittelee käytetyn tyylin. Tyyli voi viitata tiedoston `content.xml` alussa elementin `office:automatic-styles` sisällä määritettyihin kappalekohtaisiin tyyliin tai tiedostossa `styles.xml` määritettyihin varsinaisiin tyyliin. Kappalekohtaiset tyylit määritellään elementeissä `style:style` ja ne koostuvat P-kirjaimesta sekä juoksevasta numerosta. Kappalekohtaisessa tyyliin määritetään vain kappaleeseen liittyviä ominaisuuksia. Tällaisia ovat esimerkiksi sivun tai listojen asettelun vaikuttavan tyylin vaihtuminen sekä käyttäjän määrittelemä sivunvaihto. Tyy-

lin muut tiedot peritään varsinaisesta tyylistä. Kappaleiden sisällä voidaan käyttää tekstikohtaisia tyylejä, jotka määritellään muuten samalla tavalla kuin kappalekohtaiset paitsi niiden nimet alkavat T-kirjaimella. [11]

Seuraavassa on lyhyen tekstikappaleen ja sen käyttämien tyylien määrittely:

```
<text:p text:style-name="P1">
  Tämä on kappaleen tekstiä. Sana
  <text:span text:style-name="T1">
    lihavoitu
  </text:span>
  on kirjoitettu käyttäen paksumpaa kirjasinta.
</text:p>

<style:style style:name="P1" style:family="paragraph"
style:parent-style-name="Text_20_body"
style:master-page-name="First_20_Page">
  <style:paragraph-properties style:page-number="auto"/>
</style:style>

<style:style style:name="T1" style:family="text">
  <style:text-properties fo:font-weight="bold"/>
</style:style>
```

4.4.2 Styles.xml-tiedoston sisältö

Tiedostossa `styles.xml` määritellään kaikki tyylit, poislukien kappale- tai tekstikohtaiset. Tyylit voivat olla käyttäjän muokkaamia sisäänrakennettuja tai omia tyylejä. Nämä tyylit on määritetty käyttäen `office:styles`-elementtejä. Sivutyyleihin liittyvien ylä- ja alatunnisteiden määrittelyt ovat `office:master-styles`-elementeissä. `office:automatic-styles`-elementeillä määritetään automaattisesti luodut tyylit. Näitä tyylejä käytetään tehtäessä kohdekohtaisia asetuksia, jolloin ohjelmisto luo käyttäjälle näkymättömän tyylin. `office:font-face-decls`-elementit sisältävät fonttien esittelyitä. Näitä käytetään avuksi löytämään lähin vastaava fontti, mikäli toisesta tietokoneesta puuttuu jokin dokumentissa käytetty fontti. [11]

Tyylin varsinaiseen määrittelyyn käytetään paljon eri sisäkkäisiä elementtejä ja attribuutteja. Kaikki tyylit määritellään elementin `office:styles` sisällä määrittelemällä kukin tyyli oman elementin `style:style` sisällä. Elementin attribuutteina ovat tyylin tunnistamiseen liittyviä attribuutteja, kuten `style-name`, jonka arvona on tyylin nimi, `style:parent-style-name`, jonka arvona on perittävä tyyli sekä `style:family`, jonka arvona on tyylin perhe. Tyylin nimessä ei saa ol-

la muita merkkejä kuin kirjaimet a–z, numerot ja alaviiva. `style:display-name` kertoo käyttäjälle näkyvän nimen. Se voi muitakin merkkejä ja sen arvona käytetään kielikohtaisia tyylejä. Tyylit periytyvät, eli kaikki ominaisuudet mitkä ei ole määritetty tyylin määrittelyssä peritään vanhemmilta tyyleiltä. Elementti `style:style` voi sisältää elementin `style:paragraph-properties`, jossa määritetään sen attribuutteina kappaleisiin liittyviä ominaisuuksia, kuten marginaalit ja sidonta sekä elementin `style:text-properties`, jossa määritetään sen attribuutteina kirjaintyyliin liittyviä ominaisuuksia, kuten kirjasimen tyyli, koko, korostus ja väri. [11]

Seuraavassa on kuvattu esimerkkinä Text body -tyylin määrittelyä:

```
<style:style style:name="Text_20_body" style:display-name="Text body"
style:family="paragraph" style:parent-style-name="Standard"
style:class="text">
  <style:paragraph-properties fo:margin-top="0cm"/>
  <style:text-properties fo:font-size="12pt" fo:font-style="italic"/>
</style:style>
```

Sivun asetusten määrittelyyn käytetään `style:page-layout-elementtejä`. Jokaista erilaista sivun asettelua varten luodaan oma tyyli. Sivun asettelussa määritellään koko, marginaalit, sivunumeron kirjasin sekä ylä- ja alatunnisteiden alueen koko. Sivun asetteluun viitataan `style:master-page-elementin` `style:page-layout-name-attribuutilla`. Ylä- ja alatunnisteiden sisällön määrittelyyn käytetään omia elementtejä `office:master-page-elementin` alla.

Seuraavassa on esimerkki sivun asetteluun määrittelyistä:

```
<style:master-page style:name="First_20_Page" style:display-name=
"First Page" style:page-layout-name="Mpm2">
  <style:header>
    <text:p text:style-name="MP1">
      Ylätunniste
    </text:p>
  </style:header>
  <style:footer>
    <text:p text:style-name="Footer">
      <text:page-number style:num-format="i" text:select-page=
"current">
        i
      </text:page-number>
    </text:p>
  </style:footer>
</style:master-page>
```

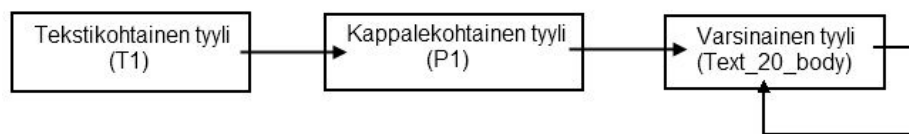
```

<style:style style:name="MP1" style:family="paragraph"
style:parent-style-name="Header">
  <style:paragraph-properties fo:text-align="start"
  style:justify-single-word="false"/>
</style:style>

<style:page-layout style:name="Mpml">
  <style:page-layout-properties fo:page-width="21.001cm" fo:page-
height="29.7cm" style:num-format="1" style:print-orientation=
"portrait" fo:margin-top="2cm" fo:margin-bottom="2cm" fo:margin-
left="2cm" fo:margin-right="2cm"/>
  <style:header-style>
    <style:header-footer-properties fo:min-height="0cm" fo:margin-
left="0cm" fo:margin-right="0cm" fo:margin-bottom="0.499cm"/>
  </style:header-style>
  <style:footer-style/>
</style:page-layout>

```

Kuvassa 4.2 on kuvattuna tyylien periytyminen. Varsinainen tyyli voi periytyä toisesta varsinaisesta tyylistä.



Kuva 4.2: ODF-tiedostoformaatin tyylien periytyminen

4.4.3 Muiden tiedostojen ja hakemistojen sisältö

Tiedostossa `manifest.xml` on tieto kaikista dokumentin tiedostoista. Se määrittää jokaiselle tiedostolle tiedostotyyppin ja osoitteen. Tiedosto `meta.xml` sisältää dokumentin metatiedot, kuten tekijän, päivämäärän, otsikon sekä avainsanat. Metatiedoissa on meta- ja dc-nimiavaruudessa määritettyjä elementtejä. Näistä dc-elementit ovat Dublin Core -metatietoja. Tiedosto `mimetype` määrittelee dokumentin tyyppin MIME-tyyppitunnisteella. Tekstinkäsittelydokumentin MIME-tyyppitunniste on esimerkiksi `application/vnd.oasis.opendocument.text`. Tiedosto `settings.xml` määrittää sovelluskohtaiset asetukset. Näillä asetuksilla on merkitystä vain dokumentin laadintaan käytetyille toimisto-ohjelmalle. Hakemistokansio `Pictures` sisältää dokumentin kuvat. Toimisto-ohjelma yleensä muuttaa

kuvien nimet helpommin ja varmemmin viitattaviin nimiin, kuten kuvassa 4.1. Hakemistokansiot `Object 1` ja `ObjectReplacements` sisältävät OLE-objektien määrykset. [11]

4.5 Hyödyt ja vahvuudet

ODF-tiedostoformaattilla on useita vahvuuksia. Suurena vahvuutena on hyväksyminen ISO-standardiksi. ODF-tiedostoformaattille on erittäin laaja ohjelmistojen tuki. Sitä tukevat jossain määrin lähes kaikki toimisto-ohjelmistot, joista useat ovat ilmaisia. ODF-tiedostoformaatti käyttää valmiita standardeja aina kuin mahdollista, mikä parantaa yhteentoimivuutta. ODF-tiedostoformaattilla on eri organisaatioiden ja yritysten laaja tuki. ODF-tiedostoformaatti ei ole alustariippuvainen, vaan kaikille yleisimmille käyttöjärjestelmille on olemassa ODF:ää tukevia ohjelmistoja. [23]

ODF Alliance ja laaja yritysten tuki takavat korkean yhteentoimivuuden ja tiedostoformaatin pysyvyyden. Isoista yrityksistä ainakin Adobe, BBC, Corel, IBM, Intel, MySQL AB, Novell, Oracle ja Sun Microsystems kuuluvat ODF:n kehityksestä vastaaviin organisaatioihin. [23]

4.6 Rajoitukset ja heikkoudet

ODF-tiedostoformaatti ei määrittele kaikkia dokumentin yksityiskohtia. Puuttuvat määrykset liittyvät usein ulkoasuun. Näiden lopullinen toteutus riippuu käytetystä ympäristöstä. Toimisto-ohjelma ja käyttöjärjestelmä vaikuttavat puuttuvien määryksien lopputulokseen. Toteutuksesta riippuvia tekijöitä ovat muun muassa fontit ja niiden piirtoalgoritmit, makrokieli, tavutusalgoritmit sekä rivitys- ja sivutusalgoritmit. [13]

Mikäli käytetystä ympäristöstä ei löydy samaa kirjasintyyliä, korvataan se samankaltaisella kirjasintyyllillä. Kirjaimet ovat eri kirjasintyyleissä eri kokoisia, joten rivitys saattaa olla erilainen käytetystä ympäristöstä johtuen. Myös kirjasinten piirtoyyli vaihtelevat eri ympäristöissä. ODF-tiedostoformaatti ei tue kirjasintyylien sisällyttämistä dokumentteihin mm. tekijänoikeudellisista syistä. [13]

Sanojen tavutus vaikuttaa myös dokumentin lopulliseen ulkonäköön. Tavutusta ei välttämättä ole asennettu kaikkiin käyttäjän tietokoneisiin. Toimisto-ohjelmat käyttävät useita tavutuskomponentteja, kuten ohjelman sisäisiä ja laajenuksina asennettuja. Komponentit pohjautuvat usein joko sanastopohjaiseen tai sääntöpoh-

jaiseen tavutukseen. Suomen kielen tavutukseen on tehty sanastopohjaiset komponentit Soikko ja Voikko. Eri tavutuskomponentit tavuttavat vähän eri tavoilla. [13]

4.7 Käsittely ohjelmointikielillä ja työkaluilla

ODF-tiedostoformaatin pohjana oleva XML-merkkäuskieli mahdollistaa dokumenttien käsittelyn muillakin kuin toimisto-ohjelmilla. Erilaisia muuntajia ja suotimia on paljon tarjolla. Muuntajilla voidaan dokumentit muuttaa johonkin toiseen tiedostoformaattiin. Suotimilla voidaan suodattaa jotain tiettyä tietoa ulos dokumentista. Tätä voidaan käyttää avuksi esimerkiksi dokumenttien indeksoinnissa.

Dokumentin käsittelyyn vaaditaan ZIP-purku ja pakkausohjelmaa tai ZIP-ohjelmointirajapintaa, koska dokumentti on kasattu yhdeksi tiedostoksi käyttämällä kyseistä ZIP-pakkausta. [11]

Dokumentteja voidaan käsitellä erilaisilla menetelmillä. Yksi merkittävä keino on XSLT-suotimien käyttö. Siinä XML-tiedostoa voidaan muokata ja lukea tai se voidaan muuttaa esimerkiksi HTML-tiedostoksi. [11]

XML-pohjaisuuden vuoksi dokumenttia voidaan muokata XML-rajapintojen avulla. Yleisimmin käytetyt rajapinnat ovat SAX (Simple API for XML) ja DOM (Document Object Model). SAX-rajapinta perustuu yksisuuntaiseen tiedoston lukemiseen alusta loppuun. Siinä käsitellään tapahtumia. DOM-rajapinnassa taas luetaan koko dokumentti ja muodostetaan siitä puutietorakenne. Tietorakenteesta voidaan hakea elementtejä ja attribuutteja. API- ja DOM-rajapinnat on saatavilla kaikille yleisimmille ohjelmointikielille, kuten Java, Perl, Python ja C-kielen eri versiot. [11], [13]

Sun ja IBM ovat luoneet avoimen koodin yhteisön ODF Toolkit Unionin, jonka tarkoitus on tarjota kehittäjille työkaluja ODF-tiedostoformaatin kanssa työskentelyyn [31]. Grönroosin ja Karjalaisen [13] mukaan myös sovellusohjelmakohtaiset rajapinnat tarjoavat välineen hallita dokumentteja toimisto-ohjelman kautta. Esimerkiksi UNO-rajapintaa (Universal Network Objects) voidaan käyttää monista eri ohjelmointikielistä ja sitä voidaan käyttää prosessien sisällä sekä hajautetussa arkkitehtuurissa.

5 OOXML-tiedostoformaatti

OOXML (lyhennetty sanoista Office Open XML) (käytetään myös nimeä OpenXML) on Microsoftin kehittämä tiedostoformaatti. Se kehitettiin korvaamaan aikaisemmin käytetty binäärimuotoinen tiedostoformaatti, joka ei enää pystynyt vastaamaan muuttuviin haasteisiin. Uusi tiedostoformaatti on kuitenkin suunniteltu tukemaan kaikkia ominaisuuksia, jotka ovat olleet binääri-tiedostoformaattissa. Se perustuu XML-merkintäkieleen. Luvussa esitellään OOXML-tiedostoformaatin keskeiset piirteet. [26]

5.1 Kehitys

Office sai paljon kritiikkiä binäärimuotoisesta tiedostoformaattista, jolla luotuja dokumentteja ei saanut luotettavasti auki ilman kyseistä ohjelmistoa. Myös ODF:n kehitys toi paineita Microsoftille julkaista vastaava avoin XML-pohjainen tiedostoformaatti. ECMA (European Computer Manufacturers Association) International hyväksyi OOXML:n kansainväliseksi standardiksi (ECMA-376) vuonna 2006. Vuonna OOXML hyväksyttiin ISO-standardiksi (ISO 29500). Microsoft otti käyttöön OOXML-tiedostoformaatin ensimmäistä kertaa Microsoft Office 2007-versioon. [2]

Antikaisen [2] mukaan tiedostoformaatin spesifikaatio on jaettu kahteen eri versioon, siirtymään (transitional) ja tiukkaan (strict). Siirtymä-versio parantaa yhteensopivuutta vanhaan binääriseen tiedostoformaattiin ja tiukasta versiosta on karsittu perinneominaisuudet, jotka heikentävät yhteensopivuutta muihin avoimiin tiedostoformaatteihin, ja tekevät siitä osittain suljetun. Tiukasta versiosta on tarkoitettu tulla puhdas avoin versio. Kahtiajako vaikutti ISO-standardiksi hyväksymiseen. Microsoft Office 2010 -version piti tukea standardia täysin. Microsoftin [21] mukaan ohjelmisto tukee tiedostojen lukemisen ja kirjoittamisen osalta siirtymäversiota sekä tiukkaa versiota ainoastaan lukemisen osalta.

5.2 OOXML-tiedostoformaattia käyttävät ohjelmistot

Antikaisen [2] mukaan OOXML-tiedostoformaattia käyttävät Microsoft Officen versiot 2007 ja 2010. Microsoft Office 2003 -versioon on saatavilla tuki-

paketti, jolla OOXML-tiedostoformaatti saadaan käyttöön. Novellin versio OpenOffice.orgista, Pages ja WordPerfect tukevat OOXML-tiedostoformaatin käyttöä. Näistä OpenOffice.org tukee OOXML-tiedostoformaattia parhaiten, mutta myös siinä on useita ongelmia. Myös usean muun valmistajan ohjelmistossa on tuki, mutta näiden toimivuus on usein puutteellista. Docx4all on avoimen lähdekoodin toimisto-ohjelmisto projekti, joka käyttää OOXML-tiedostoformaattia [12].

5.3 Tiedostopäätteet

Ricen [26] mukaan Microsoft Office käytti ennen toimisto-ohjelmisto dokumenttien tallentamiseen mm. .doc- ja .dot-päätteisiä tiedostoja. Uuden tiedostoformaatin myötä tuttuihin päätteisiin lisättiin x-kirjain loppuun (doc -> docx) symboloimaan XML-merkkaukieleen perustuvaa tallennustapaa. Uuteen tiedostomuotoon ei haluttu ottaa mukaan makroja oletuksena, joten makrot hyväksyvät tiedostot tallennetaan käyttämällä m-kirjainta (doc -> docm). Tiedostoformaattissa käytetyt tiedostopäätteet on kuvattu taulukossa 5.1.

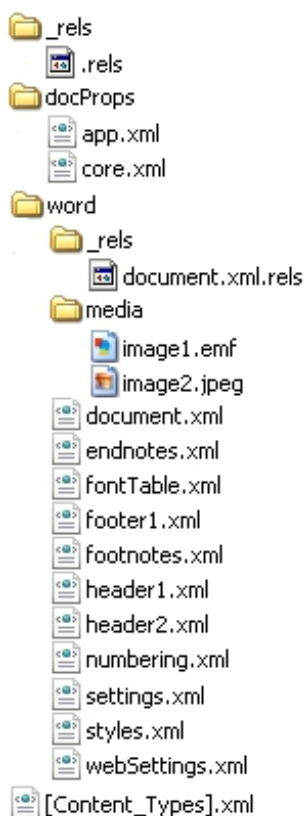
Taulukko 5.1: OOXML:n mukaiset tiedostomuodot ja tiedostotarkenteet [26]

Asiakirjatyyppi	Tiedostotarkenne	Makroilla
Tekstiasiakirja	.docx	.docm
Tekstiasiakirjan malli	.dotx	.dotm
Laskentataulukko	.xlsx	.xlsm
Laskentataulukon malli	.xltx	.xltm
Binaarinen laskentataulukko	.xlsb	-
Add-In laskentataulukko	-	.xlam
Esitys	.pptx	.pptm
Esityksen malli	.potx	.potm
Add-In esitys	-	.ppam
Esitys (vain luku)	.ppsx	.ppsm

5.4 Tiedoston rakenne

Dokumentti muodostuu useista eri XML-tiedostoista, kansioista, mahdollisista kuvista ja muista tiedostoista. Eri tiedostot on pakattu yhdeksi tiedostoksi ZIP-pak-

kausta käyttäen. XML-tiedostot sisältävät sekä dokumenttiin että ohjelmistoon liittyviä määrittelyksiä. [26]



Kuva 5.1: Puretun OOXML-tiedostoformaatin tekstinkäsittelydokumentin tiedostorakenne [26]

Dokumentin sisällön kannalta merkittävimmät tiedostot ovat `document.xml`, jossa määritetään suurin osa dokumentin tekstisisällöstä, sekä `styles.xml`, jossa määritellään tyylit ja muotoilut. Poiketen ODF-tiedostoformaatin tekstiä on määritetty myös muissa tiedostoissa. Näitä tiedostoja ovat esimerkiksi ylä- ja alatunnisteet määrittelevät tiedostot. [26]

Kaikki tekstinkäsittelytiedostot eivät ole edellisen listan mukaisia. Listasta puuttuu kaikki mediatiedostot, kuten kuva-, video- sekä äänitiedostot. Jotkin `.xml`-päätteiset tiedostot syntyvät vasta, kun jotain tiettyä ominaisuutta käytetään. Näiden tiedostojen määrä vaihtelee dokumenttien mukaan. Ylä- ja alatunnisteiden määrittelyyn käytettyjä tiedostoja `headerX.xml` ja `footerX.xml` voi olla useampia tai niitä ei ole ollenkaan. Myös tiedostoja `endnotes.xml` ja `footnotes.xml` ja `numbering.xml` ei välttämättä esiinny. [26]

Muun tyyppisissä tiedostoissa on osittain samat tiedostot. Kansio `word` sisältää tekstinkäsittelydokumenttiin liittyvät tiedostot. Taulukkolaskentadokumentissa vastaavana kansiona on `xl` ja esitysgrafiikkadokumentissa `ppt`. Muut tiedostot ja kansiot ovat yleisiä ja näin ollen ovat saman nimisiä kaikissa dokumenttityypeissä. [19]

5.4.1 Document.xml-tiedoston sisältö

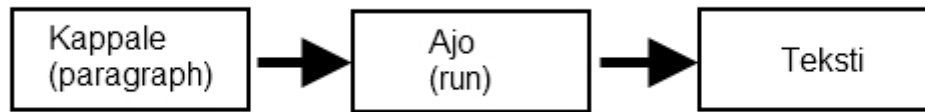
Kuten jo aikaisemmin mainittiin, varsinainen dokumentin tekstisisältö on määritelty tiedostossa `document.xml`. Tiedoston juurielementtinä on `w:document`. Juurielementin attribuutteina ovat nimiavaruusmäärittelyt. Esimerkiksi tekstinkäsittelydokumentin nimiavaruutena käytetty `w` määritellään `http://schemas.openxmlformats.org/wordprocessingml/2006/main-nimiavaruuteen`. Sen sisällä on elementti `w:body`, joka sisältää kappaleiden sisällön määrittämiin käytettyjä `w:p`-elementtejä sekä mahdollisesti osioon määrittämiin käytetyn `w:sectPr`-elementin (*section properties*). Elementti `w:p` voi sisältää yhden tai useamman ajon (*run*) sekä mahdollisesti kappaleen ominaisuudet määrittävän `w:pPr`-elementin (*paragraph properties*). Ajot on määritetty `w:r`-elementeillä. Elementin sisällä on ajon ominaisuudet määrittävä `w:rPr`-elementti (*run properties*) sekä varsinaisen tekstin sisältävä `w:t`-elementti. [22]

Seuraavassa on esitetty elementtien koostuminen: [22]

```
<w:document>
  <w:body>
    <w:p>
      <w:r>
        <w:t>tekstiä...</w:t>
      </w:r>
    </w:p>
  </w:body>
</w:document>
```

Kappale koostuu siis yhdestä tai useammasta ajosta ja ajojen sisällä kuvataan varsinainen tekstisisältö. Kuvassa 5.2 on esitetty tekstihierarkia.

Kappaleiden sisällä ajot määrittävät yhteisiä ominaisuuksia sisältäviä tekstialueita. Ominaisuudet on määritetty `w:r`-elementin sisällä. Elementin ensimmäinen lapsielementti `w:rPr`-elementti määrittää ominaisuudet, eli sen avulla säädellään sisällä olevien teksti-elementtien sisällön esittäminen. [29]



Kuva 5.2: OOXML-tiedostoformaatin tekstihierarkia [29, s. 59]

Tyylimääritykset vaikuttavat kaikkiin lapsielementteihin. Kappaleen määritykset vaikuttavat sen sisällä olevien ajojen määrityksiin. Ajojen määritykset vaikuttavat tekstien määrityksiin. [2]

Tyylit koostetaan eli tyylin ominaisuudet peritään, ellei ominaisuutta ole määritetty kyseisessä kohdassa. Esimerkiksi tekstiosion määrityksissä voi olla pelkkä lihavoitintymääritys ja muut ominaisuudet on peritty. Varsinaisiin tyyleihin viitataan `w:rsidR`-attribuuteilla. Varsinainen tyyli on määritetty `styles.xml`-tiedostossa.

5.4.2 Styles.xml-tiedoston sisältö

Tiedosto `styles.xml` määrittää käytetyt tyylit. Oletustyyli määritetään elementin `w:docDefaults` sisällä. Tyylit määritellään elementeillä `w:style`. Kuten aiemmin mainittiin attribuutti `w:rsidR` toimii tyylin viitteenä `document.xml`-tiedostosta. Tyyliden määrityksissä vastaava määrittäminen on elementissä `w:rsid`, jolla on attribuutin `w:val` arvona vastaava merkkijono.

Seuraavassa on kuvattu esimerkkinä Normal-tyylin määrittelyä:

```

<w:style w:type="paragraph" w:default="1" w:styleId="Normal">
  <w:name w:val="Normal"/>
  <w:qFormat/>
  <w:rsid w:val="00BF081C"/>
  <w:rPr>
    <w:sz w:val="24"/>
    <w:szCs w:val="24"/>
  </w:rPr>
</w:style>
  
```

5.4.3 Muiden tiedostojen ja hakemistojen sisältö

Ehrlin [10] mukaan tiedosto [`Content_Types.xml`] sisältää tiedon kaikista dokumentin osista ja osien sisältötyypit. Kansio `_rels` sisältää `.rels`-tiedostot, jotka määrittävät dokumentin eri osien suhteet. Suhteet määritetään `Relationship`-ele-

menteillä. Elementillä on attribuutit `Id`, joka yksilöi suhteen, `Type`, joka kertoo suhteen tyyppin sekä mahdollisesti `Target`, joka kertoo suhteen kohteen. Toimisto-ohjelmat aloittavat dokumentin koostamisen `.rels`-tiedostojen mukaan [19].

Seuraavassa on kuvattu kuvattu tiedostojen välisen suhteen määrittely:

```
<Relationship Id="ID" Type="relationshipType" Target="targetPart"/>
```

Kansio `docProps` sisältää dokumentin ja ohjelmiston metatietoja sisältävät tiedostot. Kansio sisältää tiedostot `app.xml`, joka sisältää dokumentin laadintaan käytetyn toimisto-ohjelman määrittelyksiä, sekä `core.xml`, joka sisältää itse dokumentin määrittelyksiä, kuten tekijän, otsikon, avainsanat ja kuvauksen. Osiin määrittelyksiin käytetään Dublin Core -metatietoja. [10]

Tiedosto `endnotes.xml` sisältää loppuviitteet ja tiedosto `footnotes.xml` sisältää alaviitteet. Tiedosto `settings.xml` sisältää toimisto-ohjelmiston asetusten määrittelyksiä. Tiedosto `fontTable.xml` sisältää määrittelyksiä yleisimmistä fonteista. Tiedosto `headerX.xml` sisältää ylä- ja alaluvun määrittelyksiä ja tiedosto `footerX.xml` sisältää alaluvun määrittelyksiä. Jokaiselle erilaiselle ylä- ja alaluvulle on oma tiedosto. Ylä- ja alaluvut voi vaihtua osanvaihdon yhteydessä. Hakemistokansio `media` sisältää dokumentin kuvat. [10]

5.5 Hyödyt ja vahvuudet

OOXML-tiedostoformaattilla on useita vahvuuksia verrattuna aikaisempaan binäärimuotoiseen tiedostoformaattiin. OOXML on yhteentoimivuudeltaan parempi. Useilla eri ohjelmistoalustoilla voidaan tuottaa OOXML-dokumentteja tuottavia ja hyödynnäviä sovelluksia. OOXML ei myöskään rajoita käytettävien video-, kuva- tai äänitiedostojen tiedostoformaattia. Puuttuvien fonttien määrittelyyn OOXML:ssä on fonttimäärittelyksiä, jotka helpottavat vastaavan fontin löytämistä. XML-kielen ja XML-nimiavaruuksien käyttö on tärkein yhteentoimivuuteen vaikuttava tekijä. [22]

OOXML on kansainvälinen tiedostoformaatti ja siinä on otettu huomioon useita merkittäviä seikkoja. OOXML tukee useiden kielten ominaispiirteitä. Numeroiden esitystyyli, kirjoitussuunta ja erilaiset päiväykset ovat esimerkkejä tuetuista ominaispiirteistä. Kieliasetukset voidaan määrittellä jopa kappalekohtaisiksi. [22]

Sovelluskehittäjien on suhteellisen helppo omaksua OOXML-dokumenttien käsittely, koska kaikkiin osa-alueisiin ei tarvitse puuttua, vaan voidaan muuttaa vain haluttua osiota. XML-merkkauskieli ja sen käsittelyyn tarjolla olevat työkalut ovat useille sovelluskehittäjille tuttuja ennestään. Yksittäisiä XML-tiedostoja voidaan myös käsitellä vaikuttamatta muihin tiedostoihin. [22]

OOXML-dokumentit ovat keskimäärin 25 % pienempiä kuin binäärimuotoiset. Ne voivat olla jopa 75 % pienempiä. ZIP-pakkaus, lyhyiden elementtien nimien ja runsas viitteiden käyttö pienentävät dokumenttien kokoa. Viitteet vähentävät saman tiedon esittämistä useaan kertaan. Taulukkolaskentadokumentissa jätetään tyhjät solut tallentamatta, mikä pienentää taulukkolaskentadokumenttien kokoa. [22]

OOXML on myös luotettavampi tiedostoformaatti verrattuna binääriseen. Vioittuneet tiedostot saadaan avattua varmemmin, koska toimisto-ohjelmat tulkitsevat OOXML-tiedostoformaattia paremmin. Myös dokumentin koostuminen useista tiedostoista lisää luotettavuutta. Mikäli yksi tiedosto on vioittunut, saadaan muut osat kuitenkin tulkittua luotettavasti. [26]

Metatiedot, kuten tekijän nimen, voi muuttaa tai poistaa helposti, jolloin voidaan lisätä yksityisyyttä. Makrojen puuttuminen oletustiedostomuodoista lisää tietoturvaa. X-päätteiset tiedostomuodot eivät voi sisältää makroja, joten ne voidaan avata ilman haitallisten koodien suorittamisen pelkoa. [26]

OOXML on kansainvälinen ISO-standardi, joka on merkittävä vahvuus kilpailijoihin nähden. Tiedostoformaattilla on myös tuki vanhoihin binäärimuotoisiin Microsoft Officen tiedostoformaatteihin [22]. Linkkien päivitys on helpoa, koska OOXML-tiedostoformaattissa linkkien kohteet määritellään erillisessä relationships-tiedostossa [26].

5.6 Rajoitukset ja heikkoudet

OOXML-standardi on vaikeasti dokumentoitu ja sen dokumentointi vaatii yhteensä noin 6000 sivua. Standardi ei hyödynnä kaikilta osin olemassa olevia ISO-standardeja, vaan siihen on luotu esimerkiksi sivujen koon ja päivämäärien määrittämiseen omat tavat. Näin ollen on luotu ristiriitoja olemassa olevien standardien kanssa. Standardi ei myöskään noudata kaikkia W3C-suosituksia. Muun muassa se käyttää omia DrawingML- ja Math-merkkauškieliään W3C:n SVG- ja MathML-suoritusten sijaan. [2]

OOXML ei tue makrojen käyttöä, mutta Microsoft Office -toimisto-ohjelmistot tukevat VBA-makrojen käyttöä. Tämä voi aiheuttaa yhteensopivuusongelmia OOXML-dokumentteja tuottavien ohjelmistojen välille, koska VBA on toimittajastandardi. Spesifikaatio ei ole täysin valmis ja Microsoft Office -toimisto-ohjelmat käyttävät ominaisuuksia, joita ei ole määritetty OOXML-spesifikaatiossa [7].

5.7 Käsittely ohjelmointikielillä ja työkaluilla

OOXML:ää pystyy käsittelemään samoilla XML-työkaluilla kuin ODF-tiedostoformaattiakin eli XLST-suotimilla sekä DOM- ja SAX-rajapinnoilla. Rice [25] käy läpi helposti lähestyttävällä tavalla artikkelissa *How to: Manipulate Office Open XML Formats Documents* keinoja muokata OOXML-dokumentteja. Microsoft on julkaissut useita omia työkaluja OOXML-dokumenttien luontiin ja hallintaan. Esimerkiksi Open XML Format SDK 2.0 tarjoaa Microsoft .NET ajeja dokumenttien laadintaan ja muokkaukseen sekä asiakas- että palvelinympäristöön [24]. Microsoftin ylläpitäältä WWW-sivustolta <http://openxmldeveloper.org/> löytyy tietoa ja työkaluja OOXML-tiedostoformaatin dokumenttien käsittelyyn.

6 Tutkimusmenetelmät

Tutkimus pohjautuu kirjallisuuskatsaukseen sekä vertailevaan tutkimukseen ja kokeelliseen tutkimukseen. Kirjallisuuskatsaus suoritettiin etsimällä julkaisutietokannoista artikkeleita ja opinnäytetöitä. Näistä koostamalla saatiin teoriaosuus kasaan. Varsinainen tutkimusosuus on tiedostoformaattien vertailussa. Vertailuun etsittiin lähinnä vuonna 2007 julkaistuja tai uudempia lähteitä, koska tiedostoformaatit ovat uusia ja jatkuvassa muutoksessa, lisäksi aikaisemmin tehdyissä vertailuissa kiinnitetään liikaa huomiota Microsoftin vanhentuneeseen binäärimuotoiseen tiedostoformaattiin. Kokeellinen tutkimus pohjautuu kirjoittajan aikaisempaan kokemukseen tiedostoformaattien tulkinnasta. Kokeellista ja vertailevaa tutkimusta ei saatu yhdistettyä kaikilta osin, koska kirjoittajan kokemus tiedostoformaateista rajoittui oikeastaan tiedostoformaattien tulkintaan. Näin ollen näiden kahden tutkimuksen yhdistämisessä on käytetty tapaustutkimusta.

Vertailu on perusteena kaikelle tutkimukselle. Vertailevassa tutkimuksessa olennaista on miettiä, miksi vertailu tehdään, eli mitä vertailulla yritetään tuoda esille sekä mitkä tapaukset vertailuun kannattaa ottaa mukaan tavoitteen saavuttamiseksi. Tavoitteena on etsiä eri ilmiöiden välinen yhteys tai selvittää eri ryhmien välillä vallitsevat erot saman ilmiön sisällä. [14]

Tutkielmassa käytetään teoriaa testaavaa vertailevaa tutkimusta. Tarkoitus on löytää tiedostoformaattien välille yhtäläisyyksiä sekä eroavaisuuksia. Tutkimuksen lopputuloksena on tarkoitus saada tiedostoformaateille niiden valintaa puoltavia ominaisuuksia.

Kokeellisessa tutkimuksessa mitataan yhden käsiteltävän muuttujan vaikutusta toiseen muuttujaan. Yleensä aluksi perehdytään tutkimusongelmaa käsittelevään kirjallisuuteen ja aikaisempiin tutkimuksiin. Tämän jälkeen kartoitetaan ongelmat ja muodostetaan niihin liittyvät hypoteesit. Kartoituksen jälkeen laaditaan koesuunnitelma ja valitaan tutkimusmenetelmä. Näiden jälkeen suoritetaan koe. Eri vaiheet raportoidaan ja vaiheen yhteydessä usein tarkastellaan luotettavuutta. [14]

Kvasikokeellinen tutkimus eroaa varsinaisesta kokeellisesta tutkimuksesta. Siinä pyrkimyksenä ei ole kontrolloida tai manipuloida kaikkia asiaan kuuluvia muuttujia. Nämä kompromissit aiheuttavat ulkoisen ja sisäisen luotettavuuden tarkempaa arviointia. Tutkimuksen vaiheet ovat samat kuin varsinaisessa kokeellisessa tutkimuksessa. [3]

Tutkielman kokeellinen tutkimus oli kvasikokeellista tutkimusta. Tutkimus suoritettiin tutustumalla vain pintapuolisesti teoriaan, jonka jälkeen tiedostoformaatteja tulkittiin. Tutkimusongelmaan verrattavana ongelmana oli saada dokumenteista tulkittua kurssin kannalta olennaiset osiot. Tähän liittyivät aikalailla tavalliset tekstinkäsittelyn osiot. Näin ollen tiedostoformaatin selkeydestä ja lähestyttävyyden vaikeudesta saatiin tietoa. Arviointi tapahtui sovelluskehittäjän näkökulmasta. Ongelmana oli kuitenkin se, ettei Parsi-projektin aikana toteutettu varsinaista tutkimusta, vaan tehtiin prototyyppiä, jolloin luotettavuuden arviointi ja raportointi jäi vähemmälle.

Tapaustutkimuksen tavoitteena on Hirsjärven ym. [14] mukaan ilmiöiden kuvailu. Tyypillisesti valitaan tutkimuskohteeksi yksittäinen tapaus, tilanne, tapahtuma tai joukko tapauksia. Aikaisempaan tutkimukseen ja teoriaan täytyy tutustua huolella. Huolellinen taustatyö tuo arviointiin erilaisia näkökulmia ja tarkempia kysymyksiä. Tutkielmassa tapaustutkimusta ei suoriteta puhtaasti ja se ei ole olennaisin tutkimusmenetelmä. Sen tarkoitus on tukea ja yhdistää muita tutkimusmenetelmiä. Tapaustutkimuksen havaintoja peilataan vertailevassa tutkimuksessa saatuihin havaintoihin. Tarkoituksena on eritellä miltä osin kokeellisen tutkimuksen tulokset tiedostoformaatin tulkinnasta poikkeavat vertailevan tutkimuksen tuloksista.

7 Vertailu

Tiedostoformaatteja on tarkoitus vertailla tarkastelemalla jo tehtyjä tutkimuksia. Lisäksi vertailuun käytetään kirjoittajan omaa kokemusta ko. formaattien tulkinnessa. Kirjoittaja on toteuttanut sovellusprojekti-kurssilla ryhmänsä kanssa prototyypin, joka tarkastaa OpenOffice.org- ja Microsoft Office -toimisto-ohjelmilla laadittuja dokumentteja.

Vertailun kohteiksi valittiin tiedostoformaattien selkeys, pysyvyys ja pitkäaikaisuus sekä tietoturva. XML-tiedostojen sisältä tarkastellaan mm. tekstin, kuvien, taulukoiden, osan vaihdon ja tyylien määrittelyä. Tyylien määrittely on merkittävässä osassa etenkin tekstinkäsittelydokumenteissa, koska jokainen dokumentin osanen on määritetty jonkun tyylimäärityksen kautta.

7.1 Kilpailutilanne

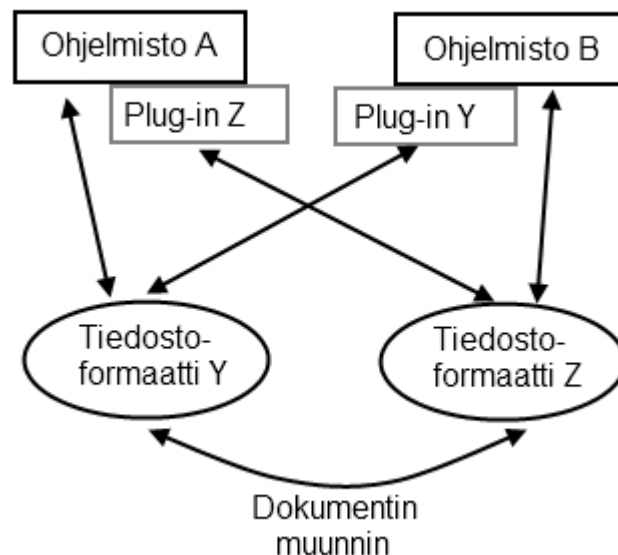
Tavallisesti on totuttu ja pyritty yhteen kansainväliseen standardiin, ns. *de jure*-standardiin. Kun ODF-standardi hyväksyttiin ISO-standardiksi, oli Microsoftin binäärimuotoinen tiedostoformaatti hallitseva markkinoilla. Se oli saavuttanut oletustalennusformaatin aseman eli sitä pidettiin standardin asemassa ns. *de facto*-standardina.

Aggarwalin ym. [1] tutkimuksen mukaan avoimet XML-pohjaiset standardit eivät ole sijoittajien mieleen. Omistusoikeudellisissa standardeissa uskotaan olevan mahdollisuus parempaan liiketoimintaan. Avoimilla standardeilla on helpompi saada osuus markkinoista, kun taas omistusoikeudellisilla standardeilla saadaan pidettyä kaikki oikeudet ja taloudelliset hyödyt itsellä.

Egyedin ja Koppenholin [8] mukaan kuluttajat hyötyvät kilpailusta klassisen käsityksen mukaan, koska kilpailu madaltaa tuotteiden hintoja, nostaa tarjontaa ja edesauttaa uusien ideoiden syntymistä eli parantaa hinta/laatusuhdetta. Kilpailu koituu näin ollen kuluttajan eduksi. Näin ei välttämättä käy kahden standardin välisessä kilpailussa, vaikka kilpailun seurauksena standardit ovatkin kehittyneet. Kahden päällekkäisen ISO-standardin olemassaolo on tuonut paljon sekaannusta. Julkishallinnon on vaikea päättää kumpaa standardia käyttää ja suosittelee käytettäväksi. Kahden hyväksytyt standardin olemassaolo on myös standardoinnin pe-

rusajatuksen, ohjata käyttäjiä ja toimittajia yhdellä muotilla toimimaan, kanssa riskitiedossa.

Microsoft perusteli oman OOXML-standardin hyväksymistä ISO-standardiksi sillä, ettei ODF tukenut Microsoftin käyttämiä binäärimuotoisia tiedostoformaateja. Kahden päällekkäisen standardin käyttö hankaloittaa dokumenttien hallintaa ja hävittää standardoinnista saatavat edut. Olemassa olevilla lisäosilla (*plug-in*) ja muuntimilla (*converter*) saadaan kummankin tiedostoformaatin dokumentit käyttöön, mutta ne tuovat kustannuksia, kuten tiedon häviämistä ja resurssien tarvetta. Kuvasta 7.1 näkee ohjelmien ja tiedostoformaattien välisen suhteen. Yhden standardin käyttö vapauttaisi ohjelmistoriippuvuudesta. [9]



Kuva 7.1: Kahden standardoidun tiedostoformaatin (X ja Y) yhteentoimivuus [9, s. 24]

Kosekin [18] mukaan OASIS sai kohtalaisen helposti standardoitua ODF:n ISO/IEC-standardiksi. Standardit vaikuttavat paljon markkinoihin, koska monet julkiset toimijat ovat velvoitettuja käyttämään ISO-standardeja. Microsoftin kilpailijat käyttivät ISO-standardin puutetta markkinoinnissa OOXML:ää vastaan. Nämä aiheuttivat sen, että markkinahallitsija Microsoftin oli kehitettävä ja ajettava oma tiedostoformaatti ISO-standardiksi. Kosekin mielestä olisi ollut parempi, että molemmat olisivat kehitelleet puutteitaan pari vuotta kauemmin ennen ISO-standardeiksi hyväksymistä ja antaneet markkinavoimien päättää kumpi tiedostoformaatti olisi tärkeämpänä hyväksytty ISO/IEC-standardiksi.

Blindin [6] mukaan laadullinen taloudellisen tehokkuuden analyysi osoittaa, että ODF- ja OOXML-standardien välistä kilpailua ISO:n suojissa ei pidä lopettaa, vaikka sitä on ehdoteltu. Epävirallinen standardi taistelisi kuitenkin virallista vastaan. Julkiselle sektorille tämä toisi epäedullisen aseman, koska siellä joudutaan usein valitsemaan virallinen standardi. Yksityisen ja julkisen sektorin yhdenvertaisten avointen kilpailutilanteiden vuoksi täytyy olla kaksi saman arvoista standardia. Microsoft aikoo julkistaa vanhan binäärimuotoisen tiedostoformaatin spesifikaation kaikkien saataville. Virallista standardointiprosessia ei pitäisi väärinkäyttää kilpailukeinona, eikä kilpailutilanteen pitäisi vaikuttaa standardointiin. Teknisesti ja taloudellisesti paremmat ratkaisut pitäisi valita standardiksi. Mikäli valittu standardi ei noudata sovittuja immateriaalioikeuksia, pitää standardointi peruuttaa.

Shahin ja Kesanin [28] tutkimuksen tulokset osoittavat, että avoimet standardit eivät aina takaa halvempia hintoja, runsaampia vaihtoehtoja ja joustavuutta. Hyödyt tulevat vasta sitten, kun on olemassa useita itsenäisiä yhteensopivia toteutuksia. Myös avoimen lähdekoodin sovellukset aiheuttavat lukittumista yhteen toimijaan. Avoimien standardien yhteensopivuus on olennaisessa asemassa, jotta avoimien standardien edut saadaan hyödynnettyä. ODF ja OOXML toteutuksilla on yhteensopivuudessa vielä parannettavaa.

7.2 Jo tehdyt vertailut

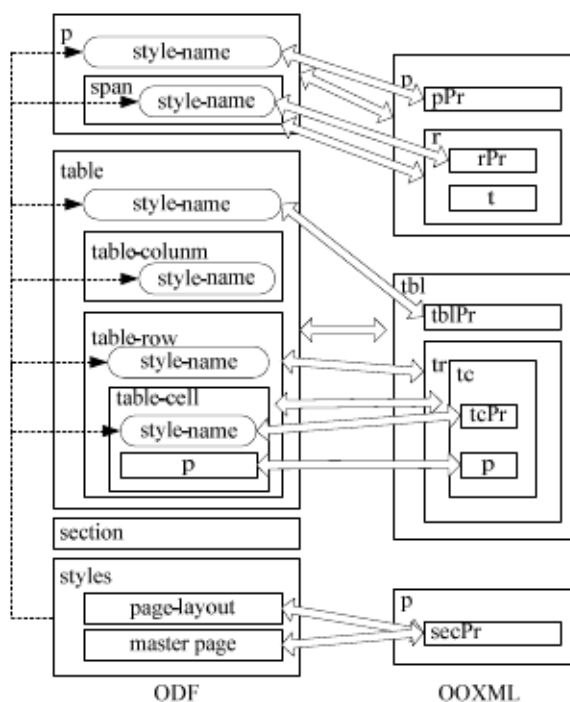
Kosekin [18] mukaan ODF- ja OOXML-tiedostoformaattit ovat yhtäläisiä monen osan suhteen. Molemmat käyttävät XML-merkkäuskieltä olennaisiin osiin, eivät määrittele isoja objekteja tiedostoissa, vaan viittaavat niihin ja pohjautuvat ZIP-pakaukseen.

7.2.1 Tiedostoformaattien selkeys

Tiedostoformaattien selkeyttä voidaan tarkastella elementtien ja attribuuttien nimeämisen kautta tai rakenteen ja viittausten kautta. Kosekin [18] mukaan ODF-tiedostoformaatti on helpompi implementoida. ODF-tiedostoformaattia tukevia laitteita ja ohjelmistoja on isompi määrä kuin OOXML-tiedostoformaattia tukevia. Useat ovat kritisoineet OOXML-tiedostoformaatin monimutkaisuutta. Monimutkaisuuden yhtenä syynä on tuki binäärimuotoisille tiedostoformaateille. Toisena syynä on Microsoft Officen ominaisuuksien runsas määrä.

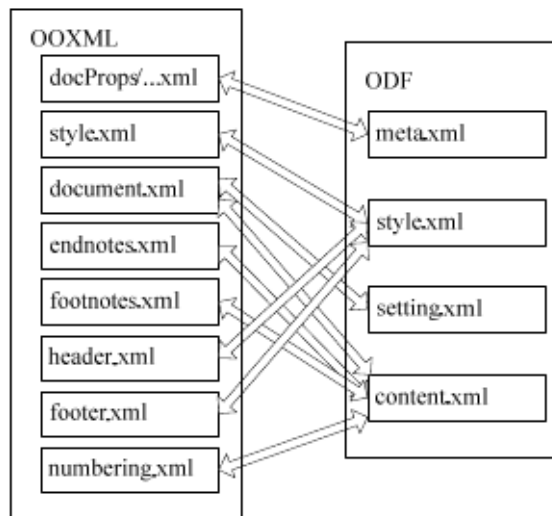
ODF- ja OOXML-standardien rakenne on erilainen, koska kummallakin on oma XML-sanasto ja semantiikka. Tiedostoformaattien loogiset rakenteet on esitetty ku-

vassa 7.2. Tiedostoformaattien rakenteet ovat saman tyyliä muttei identtisiä. Tiedostojen väliset suhteet ilmenevät kuvasta 7.3. OOXML jakaa saman tiedon useampiin tiedostoihin verrattuna ODF:ään. [36]

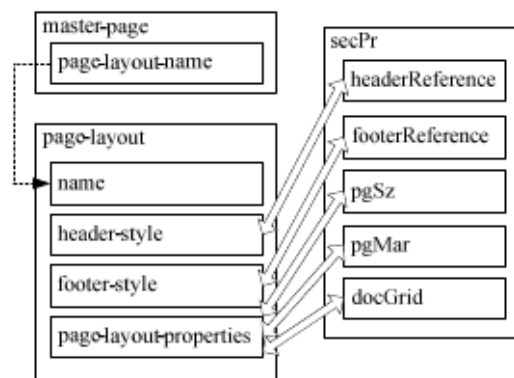


Kuva 7.2: ODF- ja OOXML-tiedostoformaattien loogiset rakenteet. Neliskulmiot kuvaavat elementtejä, ovaalit attribuutteja ja nuolet tiedostoformaattien välisiä suhteita [36, s. 298]

ODF-tiedostoformaattissa tekstin ulkoasun määrittely tapahtuu pelkästään tyylien määrittelyssä, kun taas OOXML-tiedostoformaattissa ulkoasumäärittelyä on myös lapsielementeissä, kuten kappale-elementin määrittelyä `w:pPr`-elementissä. Otsikot ja listat on määritetty omilla elementeillä ODF-tiedostoformaattissa, kun taas OOXML-tiedostoformaattissa ne on määritelty muuten samalla tavalla kuin tavalliset kappaleet, paitsi niillä on listan tai otsikon määrittävä attribuutti. Osiointi ja sivun asettelu erotetaan ODF-tiedostoformaattissa ulkoasu- (*layout*) ja isäntäsivun-tyyleillä (*master page*), kun taas OOXML-tiedostoformaattissa käytetään `w:secPr`-elementtiä. Kuvassa 7.4 esitetään tiedostoformaattien sivun asettelun rakenne ja tiedostoformaattien väliset suhteet. [36]



Kuva 7.3: ODF- ja OOXML-tiedostoformaattien tiedostojen rakenteet [36, s. 298]



Kuva 7.4: ODF- ja OOXML-tiedostoformaattien sivun ulkoasun määrittelyn rakenne [36, s. 300]

7.2.2 Pitkäaikaisuus ja pysyvyys

Hyvän pysyvyyden ja pitkäaikaisuuden takaamiseksi tiedostoformaatin täytyy olla selkeä ja hyvin dokumentoitu. Ohjelmistojen muuttuminen, uusien tuleminen tai olemassa olevien poistuminen ei saa vaikuttaa dokumentin pysyvyyteen.

Pysyvyys on merkittävässä osassa monella käyttäjällä, etenkin julkishallinnon puolella. Tiedostoformaatin pysyvyyden ohella ohjelmistojen tarjonta on tärkeässä asemassa. Avoimen lähdekoodin toimisto-ohjelmisto LibreOfficella on aktiivinen kehittäjäjoukko. Neljän vuoden aikana 664 eri kehittäjältä on tullut keskimäärin 2851 muutosta (*commit*) kuukaudessa. Myös OpenOffice.orgilla on aktiivinen kehittäjäjoukko. OOXML-tiedostoformaattia tukevia aktiivisia avoimia ohjelmistopro-

jekteja ei ole olemassa. Julkisella sektorilla käytettävään tiedostoformaattiin täytyy löytyä avoimen lähdekoodin ohjelmistototeutuksia pitkäaikaisuuden ja toimijariippumattomuuden takaamiseksi. [12]

7.2.3 Tietoturva

Dokumentin tietoturvaan liittyy haittaohjelmien, virusten tai muiden käyttäjälle haitallisten osioiden sisällyttäminen dokumenttiin. Tässä osiossa käsitellään poikkeuksellisesti myös toimisto-ohjelmistoja.

Turvallisuusriskejä on aina sisältynyt toimisto-ohjelmien tiedostoformaatteihin. Aktiiviset ominaisuudet, kuten makrot ja OLE-objektit, sekä piilottu tieto aiheuttavat riskit. Nykyään näiden riskit ovat vähentyneet, koska VBA-makrojen, OLE-objektien ja salakirjoituksen (*encryption*) käyttöä on rajattu OOXML-tiedostoformaattissa. XML-merkkauksen käyttö on vähentänyt riskejä huomattavasti. Tiedostoformaattien binäärimuotoiset kuvat ja objektit sekä itse toimisto-ohjelmisto sisältävät enemmän riskejä. [20]

ODF-tiedostoformaatti ja OpenOffice.org-ohjelmisto mahdollistavat makrojen kirjoittamisen useilla ohjelmointikielillä, jotka tarjoavat UNO-ohjelmointirajapinnan (Universal Network Objects) avulla pääsyn käyttöjärjestelmään. Virukset Stardust, Starbugs ja BadBunny ovat esimerkiksi pohjautuneet makroiin. Makrojen ajamisesta kysytään yleensä varmistus käyttäjältä. [20]

OOXML-tiedostoformaatti ja Microsoft Office-ohjelmisto sisältää samat riskit kuin ODF ja OpenOffice.org, koska tiedostoformaattit ovat niin samanlaisia. Isoimpana erona on OOXML:n makrojen rajoittaminen oletustallennusmuodossa. Pääasialliset tallennusmuodot eli X-päätteiset eivät salli makrojen käyttöä. Oletuksena ohjelmistossa on, ettei makroja suoriteta, vaikka avattaisiinkin M-päätteinen tiedosto. Makrojen puute OOXML:n määrittämisestä aiheuttaa sen, että tiedostoformaatti ei ole täysin avoin tietoturvan näkökulmasta. [20]

OLE-objektit ovat samanlaisia riskejä kummassakin tiedostoformaattissa. Etenkin OLE-paketti voi sisältää melkein mitä tahansa haitallista koodia. OLE-objektien suoritukseen kysytään yleensä käyttäjältä varmenne. [20]

Vanhoihin binäärimuotoisiin tiedostoformaatteihin verrattuna, ODF- ja OOXML-tiedostoformaattit ovat helpommin analysoitavissa ja niistä voidaan havaita vahingolliset osat helpommin. Ne kuitenkin sisältävät suunnilleen samat riskit kuin vanhatkin. Tietoturvariskien näkökulmasta toimisto-ohjelmistot ovat merkittävämmässä osassa verrattuna tiedostoformaatteihin. Toimisto-ohjelmistojen päivittäminen sekä asetusten muuttaminen makrojen ja OLE-objektien suorittami-

sen osalta kielteiseksi ovat merkittävät teot, jotka käyttäjä voi tehdä. OOXML- ja ODF-tiedostoformaateilla ei ole merkittäviä eroja tietoturvan suhteen. [20]

Myös Bhattacharyan ym. [5] tutkimuksessa todettiin, että OOXML-tiedostoformaatti on avoimuuden takia turvallisempi kuin binäärimuotoiset, koska dokumenttien sisältö pystytään näkemään helposti.

7.2.4 Tiedostoformaatin valinta

Ruotsissa tehdyn tutkimuksen mukaan tiedostoformaattia ei yleensä huomioitu kunnolla valittaessa toimisto-ohjelmistoa julkisella sektorilla. Tutkimuksen mukaan myös toimisto-ohjelman ja tiedostoformaatin välistä eroavaisuutta ei tiedostettu. [12]

Kosekin [18] mielestä organisaatio valitsee tiedostoformaatin käytettävän toimisto-ohjelmiston mukana. Kosek painottaa, että toimisto-ohjelmat eivät ole kuitenkaan ainoita ohjelmia, jotka vaikuttavat valintaan, koska toiminnan ohjausjärjestelmät (ERP) ja monet muut järjestelmät tuottavat raportteja suoraan toimisto-ohjelmien tiedostoformaateihin. Kummankin tiedostoformaatin käyttö tai ainakin tukeminen on suositeltavaa organisaatioille, joilla on paljon viestintää yhteistyökumppaneiden kesken.

Muuntimien ja laajennusten käyttö tiedostoformaatin muuttamisessa toiseen saattaa hävittää tietoa [18]. Xian ym. [36] jakavat muuntamiseen vaikuttavat asiat kolmeen ryhmään: Helposti muutettavat sisältävät osat, jotka ovat samanlaisia tai lähellä toisiaan. Tällaisia ovat esimerkiksi taulukot. Keskitasoiset voidaan muuttaa, koska ne sisältävät loogisella tasolla yhtäläisyyksiä, vaikka niiden määrittely eroaa merkittävästi. Tällaisia ovat esimerkiksi sivun ulkoasun määrittelyt. Muuttaminen tosin sisältää tiettyjä haasteita. Vaikean tyyppiset sisältävät paljon haasteita eikä niitä voida aina muuntaa laisinkaan. Tämä johtuu täysin erilaisesta suunnittelusta ja lähestymistavasta. Osaa toiminnallisuuksista ei ole edes toteutettu toisessa tiedostoformaattissa. Tekstinkäsittely-, esitysgraafiikka- ja taulukkolaskentadokumenttien toiminnallisuuksista 80 % voidaan muuttaa toiseen tiedostoformaattiin ODF:n ja OOXML:n kesken.

Jelliffe [16] tukee kahden kansainvälisen standardin olemassaoloa, koska standardoinnista päättävä komitea ei pysty arvaamaan kaikkea käyttöä. Näin ollen kulluttajat voivat valita käyttöönsä sopivan. Standardit eivät tue kaikkia maakohtaisia ominaisuuksia, joten on ristiriitaista, että olisi yksi kansainvälinen standardi. Jelliffe muistuttaa, että ODF:n versiossa 1.2 ja OOXML:n tiukassa versiossa on paljon korjauksia. Niiden tuloa kannattaa odottaa ennen valintaa. OOXML:n siirtymäversios-

ta pitäisi päästä eroon, eivätkä eri maat saisi julistaa sitä kansalliseksi standardiksi. Valinnat vähentäisivät Microsoftin painetta siirtyä lopulliseen tiukkaan versioon.

Bhattacharyan ym. [5] tutkimuksessa tarkastellaan OOXML-tiedostoformaatin ehdotetun valinnan yhteiskunnallisia vaikutuksia Intiassa. Epäilyksenä on, että valinta kasvattaisi kustannuksia. Valinta pitäisi tehdä tieto- ja viestintäteknologian (TVT) kehitykseen pyrkien ja välttämättä lukittumista tiettyyn tekniikkaan. TVT-sovellukset ovat tarjonneet etenkin köyhälle väestölle suurta hyötyä tarjoamalla pääsyn terveyteen, markkinoihin ja koulutukseen. Avointen standardien tarkoitus on estää lukkiutumiset tiettyyn teknologiaan tai toimijaan. Vaikka avoin standardi olisikin monopoli asemassa, silti kyseiseen teknologiaan tukeutuvilla tuotteilla on täysi kilpailu.

Intian hallituksella on paljon projekteja, joilla Intian TVT-tasoa on tarkoitus nostaa etenkin maaseudulla. Teknologiat, joita käytetään avuksi, pitäisi olla mahdollisimman halpoja ja yhteensopivia. OOXML:n valinta kansalaisten ja hallituksen väliseen yhteydenpitoon loisi sen toimittajalle yksinoikeuden. OOXML:n valinta aiheuttaisi kustannuksia myös laitteiston uusimiseen. Valinta rajoittaisi pyrkimyksiä Intian informaatioyhteiskunnan rakentamisesta. Kehitysmaiden pitäisi vaatia ehdotonta avointa standardia takaakseen kansalaistensa riippumattomuuden yhdestä teknologiasta ja toimijasta. Yhteisön tukeman teknologian valinta pitää hankinta- ja ylläpitokustannukset alhaisimina verrattuna yhden toimijan tukemaan teknologiaan. [5]

Antikaisen [2] tutkielmassa käsiteltiin luvuissa 5 ja 6 julkishallintojen suosituksia toimistodokumenttistandardin valintaan sekä käyttöönottopäätöksiä perusteineen. Tietoa saatiin kolmesta suuresta liitosta, 29 valtiosta ja 10 pienemmästä hallintoalueesta. ODF-tiedostoformaatti on hyväksytty 2 liitossa, 25 valtiossa ja 10 alueellisessa hallinnossa. OOXML-tiedostoformaatti on hyväksytty 8 valtioon ja 2 alueelliseen hallintoon, kuitenkin jokainen näistä tukee ODF-tiedostoformaatin käyttöä. Tiedostoformaatin todellista käyttöä on kuitenkin vaikea arvioida, koska käyttöönottosuunnitelmissa ja aikatauluissa on usealla maalla ollut vaikea pysyä. Tilanne ei kuitenkaan ole loppuun käsitelty ja monet saattavat päätyä kummankin standardin tukeen. Käyttöönottooperusteisiin on vaikuttanut EU:n ja YK:n suositukset avoimien standardien käytöstä. Laaja yhteensopivuus ja halvat kustannukset sekä valinta ensimmäisenä ISO-standardiksi vaikuttivat ODF:n valintaan.

Ditch [7] suosittelee organisaatioille ja laitoksille, joilla on korkeat vaatimukset turvallisuuden, pitkäaikaisuuden, arkistoinnin ja säilytyksen suhteen, tehtäväksi erillinen asiantuntijoiden laatima kartoitus valinnan suhteen. Väliaikaisena ratkaisuna voi pitkäaikaisen säilytyksen suhteen tutustua Adoben PDF/A-tiedosto-

formaattiin. Tällöin täytyy kuitenkin säilyttää myös alkuperäinen muutettava dokumentti.

Shahin ja Kesanin [28] tekemän tutkimuksen mukaan ainoastaan OpenOffice.org tarjosi täyden tuen ODF-tiedostoformaattiin ja Microsoft Office OOXML-tiedostoformaattiin tekstinkäsittelyn osalta. StarOffice tosin tarjosi käytännössä myös täyden tuen ODF-tiedostoformaattiin. Tulokset ovat huolestuttavia ja viittaavat avoimien standardien ympärillä olevaan holtittomuuteen. Tällainen epävarmuus aiheuttaa lukkiutumista tiettyyn ohjelmistoon ja toimijaan. Hallitusten pitäisi pyrkiä täyden tuen yhteensopivuuteen kahdesta syystä. Ensinnäkin hallitusten tuki avoimiin standardeihin johtuu luottamuksesta dokumentteihin pysyvästä pääsystä ja toimijariippumattomuudesta, koska kuka tahansa voi implementoida standardin. Toiseksi tiedostoformaatin hyöty kasvaa mitä useampi sitä käyttää. Hallitusten pitäisi luoda toimittajille painetta yhteensopivuuden parantamiseksi painottamalla sen tärkeyttä tiedostoformaattia ja ohjelmistoja valittaessa. Valitettavasti tiedostoformaatin valinta painottuu liikaa ohjelmistoihin yhteensopivuuden puutteen takia.

7.3 Empiirinen vertailu

Luvussa vertaillaan ODF- ja OOXML-tiedostoformaatteja toisiinsa tutkielman kirjoittajan kokemukseen pohjautuen.

7.3.1 Tausta

Tutkielman kirjoittaja osallistui vuoden 2011 kevätlukukaudella Jyväskylän yliopiston tietotekniikan laitoksen järjestämälle *TIES405 Sovellusprojekti* -opintojaksolle. Tutkielman kirjoittaja oli osallisena Parsi-nimisessä projektissa tekijänä. Luku perustuu kirjoittajan omaan kokemukseen tiedostoformaateista. Luvussa ei ole käytetty Parsi-projektin [4] ulkopuolisia lähteitä, joten asiat perustuvat hyvin pitkälle kirjoittajan omakohtaiseen käyttökokemukseen. Tarkoituksena on ollut, ettei tutkielmassa käytetyt tutkimukset olisi vaikuttanut empiiriseen vertailuun ja tämän luvun kirjoittamiseen.

Parsi-projekti suunnitteli ja toteutti Jyväskylän yliopiston tietotekniikan laitokselle prototyypin sovelluksesta, jolla voidaan tarkastaa Microsoft Office- ja OpenOffice.org-tekstinkäsittelyohjelmistoilla laadittujen XML-pohjaisten dokumenttien muotoiluja ja rakennetta. Sovellus on kehitetty kurssin Tietokone ja tietoverkot työvälineenä opiskelijoiden ja opettajien käyttöön siten, että harjoitustöistä

tarkastettavat kohteet ja virheistä annettavat palautteet on määritelty XML-tiedostoissa. [4]

Kurssilla opetellaan dokumenttien laadintaan liittyvät olennaisimmat asiat. Kurssi suoritetaan laatimalla joko Microsoft Office- (versio 2007 tai uudempi) tai OpenOffice.org- (versio 3.3 tai uudempi) toimisto-ohjelmalla harjoitustyödokumentit. Vaadittuihin dokumentteihin kuuluvat tekstinkäsittely-, taulukkolaskenta- ja esitysgrafiikkadokumentit toimisto-ohjelmien osalta. Harjoitustyöhön liittyy kurssin yleisiä ja henkilökohtaisia vaatimuksia. Dokumenteista piti saada tarkastettua kurssin suoritukseen liittyvät vaatimukset.

Projektin aikana allekirjoittanut tutustui ODF- ja OOXML-tiedostoformaatteihin käytännössä. Projektin varsinaisesta toteutuksesta iso osa oli näiden tiedostoformaattien tulkintaa. Tiedostoformaatteihin piti ensin tutustua, jonka jälkeen aloitettiin toteuttamaan sovellusta, joka tulkitsee dokumentteja käyttäen avuksi DOM-ohjelmointirajapintaa. DOM:ssa dokumentin sisältö on kuvattuna puumaiseen tietorakenteeseen. Tietorakenteesta voidaan hakea elementtejä, attribuutteja ja niiden sisältöjä. Tietorakenteen eri solmujen suhteet ovat myös hallittavissa.

Allekirjoittanut keskittyi enemmän ODF-tiedostoformaatin tulkintaan kuin OOXML-tiedostoformaattiin. Formaateihin tutustuminen suoritettiin tekstinkäsittelyn, taulukkolaskennan ja esitysgrafiikan osalta. Näistä käytännön toteutukseen päästiin vain tekstinkäsittelyn osalta. Eri tiedostot ovat kuitenkin hyvin paljon samankaltaisia saman ohjelmiston sisällä. Näistä eniten määrittäviä sisältää kuitenkin juuri tekstinkäsittely. Esitysgrafiikka ja taulukkolaskenta ovat hieman pelkistetympiä dokumentteja verrattuna tekstinkäsittelyyn.

Ennen projektin alkua kokemus XML-merkkauskielstä oli aika vähäistä. Tiedostojen ulkonäkö oli tiedossa, eli elementit, attribuutit ja tiedoston rakenne oli ennestään tuttuja. DOM-ohjelmointirajapinta ja XML-tiedostojen tulkinta ohjelmallisesti tuli uutena asiana. Edellä mainituista seikoista johtuen käsiteltyjen tiedostoformaattien tulkinta ja niihin tutustuminen alkoivat lähestulkoon tyhjistä. Teoreettiseen tutustumiseen ei käytetty paljoa aikaa. Käytännön tutustuminen aloitettiin etsimällä tiedostoformaateista helposti lähestyttäviä kohteita.

7.3.2 Tiedostoformaattien selkeys

Tiedostoformaattien helppolukuisuus vaikutti käytännön kautta lähestyttäessä siten, että tiedostoformaattia oli nopeampi tulkita. Tiedostoformaattiin pystyi tutustumaan selailemalla lähdekoodia. Etenkin elementtien ja attribuuttien nimeämiskäytännöt sekä viitteiden käyttö vaikuttivat lähestymiseen. ODF-tiedostoformaattissa

elementit ja attribuutit ovat selkeästi nimettyjä. Yksikäsitteisten ja helposti lähestyttävien nimeämiskäytänteiden takana on avoin ja laaja formaatin kehitysorganisaatio.

Seuraavassa on esitetty "Hello, World!" tekstin lähdekoodi tallennettuna ODF-tiedostoformaatin tiedostoon `content.xml`:

```
<office:body>
  <office:text>
    <text:p text:style-name="Standard">
      Hello World!
    </text:p>
  </office:text>
</office:body>
```

Seuraavassa on esitetty "Hello, World!" tekstin lähdekoodi tallennettuna OOXML-tiedostoformaatin tiedostoon `document.xml`:

```
<w:body>
  <w:p w:rsidR="00BF081C" w:rsidRPr="00AB1924" w:rsidRDefault="00AB1924">
    <w:pPr>
      <w:rPr>
        <w:lang w:val="fi-FI"/>
      </w:rPr>
    </w:pPr>
    <w:r>
      <w:rPr>
        <w:lang w:val="fi-FI"/>
      </w:rPr>
      <w:t>
        Hello World!
      </w:t>
    </w:r>
  </w:p>
</w:body>
```

Edellä esitetty ODF-esimerkki on kirjoitettu käyttäen OpenOffice.org 3.1 -ohjelmiston Writer-tekstinkäsittelyohjelmaa ja OOXML Microsoft Office 2007 -ohjelmiston Word-tekstinkäsittelyohjelmaa. Esimerkit on kirjoitettu avaamalla kyseiset ohjelmat ja kirjoittamalla teksti tekemättä mitään muutoksia dokumentin tai ohjelman asetuksiin. Kuten esimerkistä näkee, on ODF-nimeämiskäytännöt helpommin lähestyttäessä. Esimerkiksi tyylin määrittäminen on ODF-formaatissa elementin `text:p` attribuutin `text:style` arvona selkokielisesti "Standard". Sama OOXML-tiedosto-

formaatissa määrittäminen on elementin `w:p` attribuutin `w:rsidr` arvona viitteenä. Viite "00BF081C" viittaa `styles.xml` tiedostossa olevaan tyyliin, josta löytyy samalla viitteellä "Normal"-tyyli. Huomioitavaa on se, että "Normal" ja "Standard" eivät näy käyttäjälle välttämättä ohjelmistossa kyseisillä nimillä, vaan ohjelmisto voi näyttää tyylin nimen käyttäjälle esimerkiksi suomennettuna.

OOXML-tiedostoformaateissa on myös XML-tiedostoja enemmän. Esimerkiksi ylä- ja alatunnisteet ovat omina XML-tiedostoina, kun taas ODF-tiedostoformaateissa ne ovat sisällytetty `content.xml`-tiedostoon. OOXML-tiedostoformaateissa käytetään myös paljon enemmän viittauksia, eli tietoa, jota käytetään, määritellään eri paikassa. ODF-tiedostoformaateissa viitteitä käytetään hyvin vähän. Kuviin ja muuhun vastaavaan tietoon viitataan, mutta muuten tieto on pyritty määrittelemään käytetyssä kohdassa. Viittausten määrä vaikeuttaa tiedostoformaatin tulkintaa ihmisillä.

Mittayksiköinä ODF-tiedostoformaatti käyttää yleensä senttimetrejä tai tuumia. OOXML-käyttää pisteitä (*point*, *pt*) ja niiden osia. Yleensä käytetään twippejä (*twip*), joka on pisteen kahdeskymmenesosa. Yksi tuuma on 1440 twippiä. OOXML:n tapaan käyttää pelkkiä twippejä ympäristöstä riippumatta on hivenen selkeämpi kuin ODF-tiedostoformaatin kahden eri yksikön käyttäminen.

Metatiedot ovat aikalailla samalla tavalla määritetty kummassakin tiedostoformaateissa, koska kumpikin käyttää osittain Dublin Corea ja tiedostoformaattien omat määrittelyt muistuttavat myös Dublin Corea.

ODF-tiedostoformaatti on selkeämpi ja helpommin lähestyttävä. OOXML-tiedostoformaatin kehittäminen suljettuna yhden organisaation takana yhdelle ohjelmistolle on aiheuttanut sen, että siitä on tullut vaikealukuisempi. Nimeämiskäytänteinä on ollut helppo käyttää lyhyitä nimiä. Myös viitteiden ja useiden XML-tiedostojen käyttö ovat aiheuttaneet monimutkaisemman tiedostorakenteen verrattuna ODF-tiedostoformaattiin.

7.3.3 Pitkäaikaisuus ja pysyvyys

Parsi-projektin aikana tuli kiinnitettyä huomiota ODF-tiedostoformaatin puolella olevaan laajaan tukeen. Eri valmistajien toimisto-ohjelmistoja, työkaluja ja muuntimia on paljon tarjolla. Myös dokumentointi on parempaa ja kattavampaa. Nämä seikat luovat sovelluskehittäjille paremmat työskentelymahdollisuudet yhdessä ODF-tiedostoformaatin avoimuuden ja rajoittamattomuuden kanssa. ODF:n kehitystyöstä paistaa kaikessa yhteistyö ja yhteisen hyvän rakentaminen. Vahva, laaja ja monipuolinen kehitystyö luo edellytykset pitkäaikaisuudelle ja pysyvyydelle.

Microsoftin omat esittelyt olivat usein kalvoesittelyjä. Näillä saadaan kyllä nopeasti pintapuolinen tieto, mutta se ei riitä sovelluskehittäjälle. Lähdekoodia ja muita esimerkkejä löytyy ODF-tiedostoformaatile paljon enemmän kuin OOXML-tiedostoformaatile. OOXML-tiedostoformaatti on käytännössä pelkästään Microsoftin varassa. Tosin Microsoftilla on vahvat intressit ja resurssit pitää tiedostoformaatti pitkäaikaisena. Se myös pyrkii yleensä uusilla formaateilla yhteensopivuuteen vanhojen formaattien kanssa.

7.4 Pohdintaa tiedostoformaattien vertailusta

Tutkielman tavoitteena oli selvittää kumpi tiedostoformaateista soveltuu paremmin dokumenttien tallennukseen. Tyhjentävää vastausta kysymykseen ei saatu, koska valintaan vaikuttaa niin moni asia. Tiedostoformaattien keskeneräisyys vaikuttaa olennaisesti valinnan vaikeuteen. OOXML-tiedostoformaatin oikealle eli tiukalle versiolle ei ole tukea missään ohjelmistossa. Tämä yhdessä ODF-tiedostoformaatin uusimman version tuoreuden kanssa ovat syynä ettei tutkimuksia ole tehty paljoa. Tutkimusten vähyys luo epävarmuutta ja niiden keskeinen sanoma on odotellussa. Tehdyt tutkimukset käsitelivät enemmän tiedostoformaatin valintaa ja standardoinnin vaikutuksia kuin varsinaista standardien teknistä vertailua. Tästä syystä tutkielmassa on käsitelty paljon valintaan vaikuttavia seikkoja, joiden huomioon ottaminen helpottaa myöhemmin valinnan tekoa. Omat käyttötarkoitukset ja mielitymukset kartoittamalla helpotetaan valintaa.

Aiemmin tehdyt tutkimukset ja kokeellisen tutkimus yhdistyvät toisiinsa aika kapealta alalta. Tiedostoformaatin selkeys on ainut, mistä saadaan aikaiseksi kunnon vertailua. Monessa tutkimuksessa painotettiin OOXML:n sekavuutta ja pitkää määrittelyä. Tämä todettiin usein myös Parsi-projektin aikana. ODF-tiedostoformaatin tulkinta oli nopeampaa ja helpompaa. OOXML-tiedostoformaatisa lisäksi tuli usein ongelmia poikkeusten kanssa. Sama määrittely saattaa olla useassa eri paikassa ja muutosten selvittämiseen meni aikaa. Myös eri standardien käyttö pohjalla lisäsi työtä. Mikäli kumpikin käyttäisi kaikkia mahdollisia valmiita standardeja, olisivat tiedostoformaattit vielä enemmän päällekkäisiä. Pitkäaikaisuuden ja pysyvyyden osalta vertailua ei kannata tehdä, koska kokeellisen tutkimuksen osalta aihetta käsiteltiin vain pintapuolisesti.

Tiedostoformaatin valintaan vaikuttaa monta asiaa. Käytettävä toimisto-ohjelmisto on olennaisin tekijä tiedostoformaattia valittaessa, koska tiedostoformaattia on työlästä ja vaikeaa käyttää ilman sitä. Lisäksi sen valinta rajoittaa tiedostoformaatin valintaa. Tällä hetkellä toimisto-ohjelmat eivät takaa luotettavaa tallennus-

ta kilpailijan tiedostoformaattiin. Ohjelmiston valintaan vaikuttavat maksuhaluus, mieltymys, käyttöjärjestelmä, olemassa olevat dokumentit ja tulevaisuuden tarpeet. Mikäli tiedostoformaatin valinnassa ei oteta huomioon ohjelmistoa, eroavaisuudet pienenevät, koska molemmat ovat ilmaisia ja avoimia sekä perustuvat XML-merkkaukseen. Sovelluskehittäjän näkökulmasta ODF-tiedostoformaatin valinta on järkevämpää, koska sen käyttöön ei sisälly rajoitteita, tukiyhteisö on laajempi sekä dokumentointi on selkeämpää ja kattavampaa. Sovelluskehittäjille avoimet XML-pohjaiset tiedostoformaattit tuovat paljon hyötyä, helpotusta ja mahdollisuuksia valittiin sitten kumpi tahansa tiedostoformaatti.

OOXML-tiedostoformaatin suurimpina vahvuuksina ovat käytettävyys ja toimivuus Microsoft Officella sekä iso käyttäjämäärä. Käytettävyuteen ovat vaikuttaneet Microsoftin pitkäaikainen kehitystyö sekä OOXML-tiedostoformaatin ja Office-ohjelmiston kehitys ainoastaan toisiaan varten. Microsoftin halu tukea vanhoja binäärimuotoisia tiedostoformaatteja on lykännyt lopulliseen OOXML-tiedostoformaattiin siirtymistä.

OOXML ei täytä täysin avoimen standardin vaatimuksia. Näyttäisi myös siltä, että Microsoft haluaa pitää OOXML:n ainakin osittain toimittajariippuvaisena. OOXML-tiedostoformaattiin tallentaminen onnistuu luotettavasti ainoastaan Microsoft Officella, joka puolestaan vaatii alustakseen Microsoft Windowsin tai Mac OS:n. Näin ollen OOXML aiheuttaa lukittumista yhteen toimijaan. Mahdollisesti ja toivottavasti avoimen lähdekoodin toimisto-ohjelmat tarjoaisivat tulevaisuudessa mahdollisuuden luoda OOXML-tiedostoformaatin dokumentteja. [5]

ODF-tiedostoformaattilla on laaja tuki. Tuki koostuu isoista organisaatioista yksittäisiin käyttäjiin. Tiedostoformaattia tukevia ohjelmistoja on myös saatavilla paljon, joista moni on ilmaisia. ODF-tiedostoformaatti on hieman valmiimpi kuin OOXML-tiedostoformaatti, johtuen tiedostoformaatin pitkäaikaisesta kehityksestä ja uuden 1.2-version julkaisemisesta. ODF-tiedostoformaatin valinta ei luo riippuvuutta yhteen kaupalliseen toimijaan.

Tiedostoformaatin vaihtaminen toiseen ei ole tällä hetkellä suositeltavaa. ODF- ja OOXML-tiedostoformaattit ovat suhteellisen uusia ja niiden kehitys ei ole vielä valmis. ODF-tiedostoformaatin puolella ohjelmistot kehittyvät nopeasti. Vaihtamisen seurauksena vanhojen dokumenttien luotettava käyttäminen vaikeutuu. Kehitys on kuitenkin menossa suuntaan, jossa vaihtaminen ja ristiinkäyttö on mahdollista. Kun saadaan kehitettyä toimisto-ohjelmisto tai muunnin, joka tukee molempia tiedostoformaatteja täysin, voidaan olla valinnan jälkeen varmoja, että dokumentit saadaan aina oikein tulkittua. Suurilla organisaatioilla ja niillä, joilla on paljon dokumentteja säilöttävänä, vaihtamisen kynnyks on korkealla.

Toisen kansainvälisen standardin hyväksymisestä on ollut paljon haittaa. Microsoft olisi ihan hyvin voinut ottaa ODF:n Officeen tallennusmuodoksi ja pyrkiä hyötymään ohjelmiston tuottamasta yhteensopivuudesta binäärimuotoiseen ja ODF-tiedostoformaatin välillä. Tosin yhteensopivuuden täydellinen toimiminen olisi voinut olla vaikea toteuttaa. Kilpailuun menee resursseja sekä kehittäjäorganisaatioilta että käyttäjiltä. Kiireellä julkaistut tiedostoformaatit ovat hieman keskennäisiä, jotka pitäisi saada viimeistelyä mahdollisimman pian. Monen asiantuntijan mukaan kumpikaan ei ole täysin käyttökunnossa. Standardien mainostamisen sijaan ja julkishallintoihin vaikuttamisen sijaan voitaisiin enemmän keskittyä kehitystyöhön, mikäli olisi vain yksi standardi. Microsoftia on kritisoitu paljon myös siitä, että se ei ole pysynyt lupauksissa. Välillä vaikuttaa siltä, ettei sen aikomuksena ole edes siirtyä OOXML-tiedostoformaatin käytössä tiukkaan versioon, vaan pysytellä siirtymäversiossa. Tiukkaan versioon siirtyminen oli ISO-standardoinnin ehto. Mikäli Microsoft ei pidä lupauksia, pitäisi ISO-standardin peruuttamista miettiä.

Käytetyllä ohjelmistolla on suuri merkitys käyttäjän kokemukseen. Microsoftin OOXML-tiedostoformaatti toimii hyvin Microsoft Officessa, joka on käyttökokemukseltaan varsin hyvä verrattuna kilpailijoihin. Monet käyttäjät ovat myös tottuneet Microsoft Officen käyttöön. Tiedostoformaatin merkitys huomataan usein vasta siinä kohtaa, kun avataan dokumentti ohjelmistossa, joka ei tue täysin kyseistä tiedostoformaattia, jolloin aseteltu voi poiketa odotetusta.

Verkossa toimivien toimisto-ohjelmien käyttö tulee lisääntymään entisestään. Esimerkiksi Google Docs & Spreadsheets -toimisto-ohjelmistoa pystyy käyttämään ilmaiseksi internetselaimella. Ilmaisuuden lisäksi hyötyinä ovat varmennettu tallennus, joka luo dokumentille luotettavan tallennuksen, hajautettu käyttö, jonka avulla dokumentteihin ja ohjelmistoon päästään käsiksi eri paikoista ja eri koneilta ilman erillisiä toimeenpiteitä, sekä jaettu käyttö, jonka avulla useat käyttäjät voivat muokata samaa dokumenttia samanaikaisesti. Lisäksi ohjelmistoa ei tarvitse asentaa paikallisesti. Google Docs & Spreadsheets tukee ODF-tiedostoformaattia ja lukemisen suhteen myös OOXML-tiedostoformaattia. Isoimpana rajoituksena on internetyhteyden tarve. Microsoftilla on vastaavana palveluna Office Web Apps -ohjelmisto, joka otettiin käyttöön vähän Office 2010 -version julkaisun jälkeen.

7.5 Tutkimuksen luotettavuuden tarkastelu

Tutkimuksen luotettavuuteen vaikuttaa vahvasti käytetyt lähteet ja allekirjoittaneen oma kokemus. Allekirjoittaneen kokemus rakenteisista dokumenteista, XML-merkkauksesta ja tarkastelluista dokumenteista on sen verran rajallinen, että sen avul-

la lähteiden luotettavuutta on vaikea tarkastella. Tutkielmaa kirjoittaessa empiirinen vertailu kirjoitettiin ennen tehtyjen tutkimusten tarkastelua. Tällä pyrittiin estämään tutkimusten vaikutus tutkijan omaan käsitykseen ja Parsi-projektin aikana havaittuihin näkemyksiin. Tämän jälkeen tutustuttiin tehtyihin tutkimuksiin tarkemmin ja kirjoitettiin luku 7.2. Vasta luvussa 7.4 yhdistyvät kirjoittajan oma kokemus ja tehdyt tutkimukset.

Tutkimuksen lähteiksi pyrittiin valitsemaan neutraaleja lähteitä. Lähteissä on useita eri yliopistoissa tehtyjä tutkielmia sekä tunnetuissa ja arvostetuissa julkaisuissa julkaistuja artikkeleita. Tämän tarkoituksena oli, että tiedostoformaattien vertailuun ei käytetty lähteitä, joilla oli selvä yhteys jompaan kumpaan tiedostoformaattiin. Tiedostoformaattien määrittelyyn käytettiin niiden kehittäjäorganisaatioiden omia määrittelyitä ja artikkeleita. Tällä pyrittiin saamaan mahdollisimman oikea kuvaus tiedostoformaattien määrittämisestä. Näitä lähteitä ei käytetty apuna vertailussa.

Kaikilla eri vertailun osa-alueilla ei päästy kovin hyvään luotettavuuteen, koska käytettyjä lähteitä ei ollut tarpeeksi. Lähteiden valinnassa onnistuttiin kuitenkin hyvin, koska lähteinä käytettiin luotettavina pidettyjen julkaisijoiden materiaaleja. Useat valintapäätöstä sekä tiedostoformaattien etuja ja rajoitteita koskevat tutkimukset antoivat saman suuntaisia tuloksia. Kahden kansainvälisen tiedostoformaatin olemassaolon eduista ja haitoista oli eniten ristiriitaista tietoa. Osan mielestä kahden kilpailevan standardin malli on hyvä ja osa näki siinä paljon huonoja puolia. Kahden olemassa olevan standardin vaikutusten arviointi on kuitenkin vaikeaa.

8 Johtopäätökset ja yhteenveto

Avoimet XML-pohjaiset tiedostoformaattit ovat vaikuttaneet paljon toimisto-ohjelmiin ja niillä työskentelyyn. Avoimien tiedostoformaattien myötä käyttäjät voivat varmemmin luottaa dokumenttinsa pysyvyyteen ja pitkäaikaisuuteen. Avoimuuden myötä toimisto-ohjelmien ja niihin liittyvien sovellusten määrä on kasvanut huomasti, etenkin ODF-tiedostoformaatin myötä.

ODF- ja OOXML-tiedostoformaattit ovat ISO/IEC-standardeja sekä perustuvat XML-merkkaukieleen ja ZIP-pakkaukseen. XML-merkkaukieli tuo paljon etuja rakenteisten dokumenttien tallentamiseen. XML-pohjaiset tiedostot pakataan yhteen, jolloin ne näkyvät käyttäjälle yhtenä tiedostona. Kummallakin tiedostoformaattilla on omat edut ja rajoitteet, joiden pohjalta käyttäjän täytyy tehdä valinta. Tosin käytettävä toimisto-ohjelmisto vaikuttaa myös olennaisesti päätökseen.

ODF-tiedostoformaatin vahvuuksina on laaja kehitysympäristö niin standardin kuin ohjelmistojen osalta. Se on myös helpommin lähestyttävä, eikä se luo riippuvuutta yhteen toimijaan. OOXML-tiedostoformaatin vahvuuksina on Microsoft Office -toimisto-ohjelmien myötä todella laaja käyttäjäkunta ja vahva markkina-asema sekä hyvä käytettävyys. Kummallekin tiedostoformaattille on tarjolla kehitysyhteisöjä ja työkaluja helpottamaan kolmansien osapuolten työtä. Tosin ODF:n puolella ulkopuolinen kehitystyö on huomattavasti aktiivisempaa, johtuen avoimen lähdekoodin toimisto-ohjelmistoprojekteista. OOXML:n puolen kehitys tapahtuu suurimmaksi osaksi Microsoftin sisällä.

Tiedostoformaatin valinta on vaikeaa. Jokaisen käyttäjän ja organisaation on tehtävä valintansa koko käyttötarkoitusta pitkällä aikavälillä arvioiden. Julkisella sektorilla erilaiset suositukset puoltavat ODF-tiedostoformaatin valintaa. Julkisella puolella monet toimijat ovat tehneet valinnan, mutta käyttöönottosuunnitelmissa ei kuitenkaan olla pysytty kovin hyvin. Jotkut ovat päättäneet ottaa käyttöön molemmat. Tämä tosin tuo ylimääräisiä kustannuksia ja vaivaa.

Kahden kansainvälisen standardin hyväksyminen on aiheuttanut paljon keskustelua sekä puolesta että vastaan. Kahden standardin päällekkäisyys on ristiriidassa standardoinnin perusajatuksen kanssa. Osa perustelee kahden standardin hyväksymistä lisääntyneen kilpailun tuomilla eduilla ja julkisen sektorin saattamisella tasarvoiseen tilanteeseen kuluttajiin nähden, joilla on vapaus valita oman harkinnan

mukaan. Toisten mielestä kaksi standardia sekoittaa käyttäjiä ilman mitään merkittävää hyötyä.

Tiedostoformaateihin siirtyminen on vielä kesken. ODF:n puolella ollaan hieman pidemmällä, kuin OOXML:n puolella, johtuen Microsoftin halusta pitää kiinni yhteensopivuudesta vanhaan binäärimuotoiseen tiedostoformaattiin. Microsoft käyttää tuotteissaan OOXML-tiedostoformaatin väliaikaiseksi tarkoitettua siirtymäversiota. Mikäli Microsoft siirtyy OOXML-standardin tiukkaan versioon, selkeytyy kyseisen tiedostoformaatin kanssa työskentely. Mikäli tiedostoformaatin valinnalle ei ole kiirettä, kannattaa tilanteen antaa tasaantua ennen lopullista päätöstä.

Lähteet

- [1] Aggarwal Nitin, Dai Qizhi ja Walden Eric A., *Are open standards good business?*, Springer, 2012, saatavilla PDF-muodossa <URL: <http://www.springerlink.com/content/9294t50m803h4220/fulltext.pdf>>.
- [2] Antikainen Tuukka, *XML-pohjaiset toimistodokumenttistandardit julkishallinnossa*, Jyväskylän yliopisto, Tietojenkäsittelytieteiden laitos, 2010, saatavilla PDF-muodossa <URL: <http://urn.fi/URN:NBN:fi:jyu-201101131023>>.
- [3] Anttila Pirkko, *Kokeelliset ja testaukseen perustuvat menetelmät*, Virtuaaliammattikorkeakoulu, Ylemmän AMK-tutkinnon metodi-foorumi, viitattu 12.2.2012, saatavilla WWW-muodossa <URL: <https://www.amk.fi/opintojaksot/0709019/1193463890749/1193464131489/1194289356644/1194290120703.html>>.
- [4] Auvinen Vili, Kauppinen Olli ja Tammela Juho, *Parsi-projekti*, Jyväskylän yliopisto, Tietotekniikan laitos, 2011, saatavilla WWW-muodossa <URL: <http://sovellusprojektit.it.jyu.fi/parsi/>>.
- [5] Bhattacharya Jaijit, Ilavarasan P. Vigneswara ja Gupta Shantanu, *Open standards and accessibility to information: a critical analysis of OOXML in India*, ACM, ICEGOV '07, s. 151–154, 2007, saatavilla PDF-muodossa <URL: <http://doi.acm.org/10.1145/1328057.1328089>>.
- [6] Blind Knut, *An economic analysis of standards competition: The example of the ISO ODF and OOXML standards*, Elsevier, Telecommunications Policy 35, s. 373–381, 2011, saatavilla PDF-muodossa <URL: <http://www.sciencedirect.com/science/article/pii/S0308596111000218>>.
- [7] Ditch Walter, *XML-based Office Document Standards*, JISC: Bristol, UK, 2007, saatavilla PDF-muodossa <URL: <http://www.jisc.ac.uk/media/documents/techwatch/tsw0702pdf.pdf>>.

- [8] Egyedi Tineke M. ja Koppenhol Aad, *The Standards War Between ODF and OOXML*, IGI Global, 2010, saatavilla PDF-muodossa <URL: <http://www.igi-global.com/viewtitlesample.aspx?id=39086>>.
- [9] Egyedi Tineke M., *On the Implications of Competing Standards, The Pros and Cons of Standard Setting*, Swedish Competition Authority, 2010, s. 12–33, saatavilla PDF-muodossa <URL: http://www.kkv.se/upload/Filer/Trycksaker/Rapporter/Pros&Cons/rap_pros_and_Cons_standard_setting.pdf>.
- [10] Ehrli Erika, *Walkthrough: Word 2007 XML Format*, MSDN Library, viitattu 28.1.2012, saatavilla WWW-muodossa <URL: <http://msdn.microsoft.com/en-us/library/bb266220.aspx>>.
- [11] Eisenberg J. David, *OASIS OpenDocument Essentials*, Friends of OpenDocument, 2005, saatavilla PDF-muodossa <URL: http://books.evc-cit.info/OD_Essentials.pdf>.
- [12] Gamalielsson Jonas ja Lundell Björn, *Open Source communities for long-term maintenance of digital assets: what is offered for ODF & OOXML?*, University of Skövde, 2011, saatavilla PDF-muodossa <URL: <http://tutopen.cs.tut.fi/sos11/papers/cr6.pdf>>.
- [13] Grönroos Marko ja Karjalainen Martti, *OpenDocument-standardi asiakirjojen tallennusmuotona*, Oikeusministeriö, 2007, saatavilla PDF-muodossa <URL: http://arkisto.victoriamedia.net/pdftiedostot/internetoppaat/om_opas_opendocument.pdf>.
- [14] Hirsjärvi Sirkka, Remes Pirkko ja Sajavaara Paula, *Tutki ja kirjoita*, 13. osin uudistettu painos, Helsinki, Tammi, 2007.
- [15] Honkaranta Anne, *From Genres to Content Analysis Experiences from Four Case Organizations*, Jyväskylän yliopisto, 2003, väitöskirja.
- [16] Jelliffe Rick, *Should OOXML be a national standard?*, O'Reilly, 2009, viitattu 5.2.2012, saatavilla WWW-muodossa <URL: <http://broadcast.oreilly.com/2009/08/should-ooxml-be-a-national-sta.html>>.

- [17] Jones Brian ja Rajabi Zeyad, *History of office XML formats (1998-2006)*, MSDN, 2007, saatavilla WWW-muodossa <URL: http://blogs.msdn.com/b/brian_jones/archive/2007/01/25/office-xml-formats-1998-2006.aspx>.
- [18] Kosek Jirka, *From the Office Document Format Battlefield*, IT Professional, vol. 10, no.3, 2008, s. 51-55, saatavilla PDF-muodossa <URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4525543&isnumber=4525527>>.
- [19] Krieger Stephanie, *Advanced Microsoft Office Documents 2007 Edition Inside Out.*, Microsoft, 2007, kappale 22, saatavilla WWW-muodossa <URL: [http://msdn.microsoft.com/en-us/library/dd721893\(v=office.12\).aspx](http://msdn.microsoft.com/en-us/library/dd721893(v=office.12).aspx)>.
- [20] Lagadec Philippe, *OpenDocument and Open XML security (OpenOffice.org and MS Office 2007)*, Springer Paris, 2008, s. 115-125, saatavilla PDF-muodossa <URL: <http://dx.doi.org/10.1007/s11416-007-0060-2>>.
- [21] Microsoft, *Overview of the XML file formats in Office 2010*, Microsoft TechNet, Product release, 2010, viitattu 1.2.2012, saatavilla WWW-muodossa <URL: <http://technet.microsoft.com/en-us/library/cc179190.aspx>>.
- [22] Ngo Tom, *Office Open XML Overview*, ECMA International, 2006, saatavilla PDF-muodossa <URL: http://www.ecma-international.org/news/TC45_current_work/OpenXML%20White%20Paper.pdf>.
- [23] OASIS, *Open by design: The Advantages of the OpenDocument Format (ODF)*, OASIS ODF Adoption TC, 2006, saatavilla PDF-muodossa <URL: http://www.oasis-open.org/committees/download.php/21450/oasis_odf_advantages_10dec2006.pdf>.
- [24] Rajabi Zeyad ja Rice Frank, *Creating Documents by Using the Open XML Format SDK 2.0*, Microsoft Corporation, 2009, viitattu 3.2.2012, saatavilla WWW-muodossa <URL: <http://msdn.microsoft.com/en-us/library/dd440953%28v=office.12%29.aspx>>.
- [25] Rice Frank, *How to: Manipulate Office Open XML Formats Documents*, Microsoft Corporation, 2006, viitattu 3.2.2012, saatavilla WWW-

- muodossa <URL: <http://msdn.microsoft.com/en-us/library/aa982683%28v=office.12%29.aspx>>.
- [26] Rice Frank, *Introducing the Office (2007) Open XML File Formats*, Microsoft Corporation, 2006, saatavilla WWW-muodossa <URL: <http://msdn.microsoft.com/en-us/library/ms406049.aspx>>.
- [27] Salminen Airi, *Building Digital Government By XML*, Jyväskylän yliopisto, 2005, saatavilla PDF-muodossa <URL: <http://users.jyu.fi/~airi/papers/HICSS-2005.pdf>>.
- [28] Shah Rajiv ja Kesan Jay, *AI interoperability challenges for open standards: ODF and OOXML as examples*, Digital Government Society of North America, dg.o '09, s. 56–62, 2009, saatavilla PDF-muodossa <URL: <http://dl.acm.org/citation.cfm?id=1556176.1556191>>.
- [29] Suhonen Juha, *XML-dokumenttien tuottaminen*, Jyväskylän yliopisto, Tietojenkäsittelytieteiden laitos, 2008, saatavilla PDF-muodossa <URL: <http://urn.fi/URN:NBN:fi:jyu-200810205816>>.
- [30] Weir Rob, *An ODF/OOXML File Format Timeline*, 2007, saatavilla WWW-muodossa <URL: <http://www.robweir.com/blog/2007/06/file-format-timeline.html>>.
- [31] Weir Rob, *OpenDocument Format The Standard for Office Documents*, IEEE Computer Society, 2009, saatavilla PDF-muodossa <URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=4797941>>.
- [32] Weir Rob, *OASIS Open Document Format for Office Applications (OpenDocument) TC*, OASIS, viitattu 3.2.2012, saatavilla WWW-muodossa <URL: http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=office>.
- [33] Wikipedia, *OpenDocument*, Wikimedia Foundation, viitattu 28.1.2012, saatavilla WWW-muodossa <URL: <http://en.wikipedia.org/wiki/OpenDocument>>.
- [34] Wikipedia, *OpenDocument Format Alliance*, Wikimedia Foundation, viitattu 28.1.2012, saatavilla WWW-muodossa <URL: http://en.wikipedia.org/wiki/OpenDocument_Format_Alliance>.

- [35] W3C, *Extensible Markup Language (XML) 1.0 (Fifth Edition)*, 2008, saatavilla WWW-muodossa <URL: <http://www.w3.org/TR/xml>>.
- [36] Xia Hou, Ning Li, Hong-bo Yang ja Qi Liang *Comparison of Wordprocessing Document Format in OOXML and ODF, Semantics Knowledge and Grid (SKG)*, 2010, s. 297-300, saatavilla PDF-muodossa <URL: <http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=5663529&isnumber=5663480>>.