

Tuomas Komulainen

LUOVA LOMAKE ANALYSOINTITYÖKALU

Opinnäytetyö
Kajaanin ammattikorkeakoulu
Tradenomikoulutus
Syksy 2007



**Kajaanin
ammattikorkeakoulu**

**OPINNÄYTETYÖ
TIIVISTELMÄ**

Koulutusala Luonnontieteiden ala	Koulutusohjelma Tietojenkäsittelyn koulutusohjelma
Tekijä(t) Komulainen Tuomas	
Työn nimi Luova Lomake Analysointityökalu	
Vaihtoehtoiset ammattiopinnot Ohjelmointi	Ohjaaja(t) Piirainen Veli-Pekka
	Toimeksiantaja Luovaliiike Oy
Aika syksy 2007	Sivumäärä ja liitteet 41 + 3
<p>Tämä opinnäytetyö tehtiin Luovaliiike Oy:n toimeksiannosta. Toimeksiantona oli kehittää Luova Lomake -ohjelmistoon kyselytutkimuksella kerätyn aineiston analysointityökalu. Luova Lomake Analysointityökalu on tehty osaksi Luova Lomake -ohjelmistoa, joka tarjoaa käyttäjälleen mahdollisuuden muun muassa kyselytutkimusten tekoon, ilmoittautumisprosessien hoitamiseen sekä kilpailujen järjestämiseen.</p> <p>Luova Lomake Analysointityökalulla on mahdollista luoda aiemmin tehdyn kyselytutkimuksen pohjalta raporttipohjia. Raporttipohjan perusteella on mahdollista tulostaa reaaliaikainen raportti kyselytutkimuksen tuloksista. Raportti voi sisältää yksittäisiä muuttujia kuvailevaa tietoa ja muuttujien välisien riippuvuuksien esittämistä ristiintaulukoinnin avulla.</p> <p>Luova Lomake Analysointityökalu kehitettiin Microsoft ASP .NET 2.0 -alustalla, Microsoft Visual Studio 2005 -ohjelmointiympäristössä. Ohjelmointikielenä käytettiin Visual Basic .NET:iä. Analysoitava tieto on tallennettuna joko MySQL- tai Microsoft Server 2005 -tietokantaan. Työkalulla tehdyt raporttipohjat tallennetaan XML-tiedostoon, jonka avulla ohjelma generoi halutun muotoisen raportin reaaliaikaisena käyttäen tietokannassa olevaa kyselytutkimuksen vastausmateriaalia.</p> <p>Opinnäytetyön teoriaosuudessa käsitellään yleisesti XML:n eri osa-alueita sekä tilastollista tutkimusta kvantitatiivisen tutkimuksen kannalta, siltä osin kuin se liittyy opinnäytetyön käytännönosuuden toteuttamiseen.</p> <p>Luova Lomake Analysointityökalu täyttää hyvin sille määritetyt vaatimukset ja vastaa toiminnallisuudeltaan alkuperäisiä tavoitteitaan. Ongelmana ilmeni ohjelman hitaus generoitaessa raporttia, joka sisältää monia ristiintaulukoita. Tämä ongelma on kuitenkin ratkaistavissa helposti, jos luovutaan raportin reaaliaikaisesta generoimisesta ja se toteutetaan ajastetusti.</p>	
Kieli	Suomi
Asiasanat	XML, kysely, kvantitatiivinen tutkimus, raportti, ASP .NET
Säilytyspaikka	<input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun Kaktus-tietokanta <input checked="" type="checkbox"/> Kajaanin ammattikorkeakoulun kirjasto

School Business	Degree Programme Data Processing
Author(s) Komulainen Tuomas	
Title Luova Lomake Analyzing tool	
Optional Professional Studies Programming	Instructor(s) Piirainen Veli-Pekka
	Commissioned by Luovaliiike Oy
Date Autumn 2007	Total Number of Pages and Appendices 41 + 3
<p>This thesis was commissioned by Luovaliiike Oy. The aim of the thesis was to develop an analyzing tool for questionnaire material which is collected by Luova Lomake program. Among other things Luova Lomake offers possibility for users to make questionnaires, handle registrations for different kind of events and host competitions.</p> <p>The Luova Lomake Analyzing tool makes it possible to draft report forms based on questionnaires. Using the report form the program can print a real time report from the results of the questionnaire in the Internet browser. The report can include data on individual variables, as well as correlation between the variables using cross tabulation.</p> <p>The program was built by using the Visual Basic .NET programming language and the programming platform Microsoft ASP .NET 2.0. The programming environment was Microsoft Visual Studio 2005 and the database software was Microsoft SQL Server 2005 which is used to save the results of questionnaires. The report forms made by the Luova Lomake Analyzing tool are saved in the XML file and are used by the program to analyze the questionnaire results from the database to generate a real time report.</p> <p>The theory parts of the thesis focuses on XML and statistical analysis using the quantitative method. Both parts of the theory are very large so they are limited only to those topics which are related to the practical part of the thesis.</p> <p>The Luova Lomake Analyzing tool works well meeting the original functionality targets. The only problem in this tool is that it is slow when generating a versatile report. This problem is easy to solve by scheduling.</p>	
Language of Thesis	Finnish
Keywords	XML, questionnaire, quantitative method, report, ASP .NET
Deposited at	<input checked="" type="checkbox"/> Kaktus Database at University of Applied Sciences Library <input checked="" type="checkbox"/> Library of University of Applied Sciences

SYMBOLILUETTELO

API	Application Programming Interface, ohjelmointirajapinta, jonka avulla eri sovellukset tai niiden osat voivat kommunikoida keskenään.
CSS	Cascading Style Sheets, tyyliohje jota käytetään erityisesti WWW-dokumenteissa.
GUID	Globally Unique Identifier, 128-bittinen tunniste, joka koostuu 32-merkistä (esimerkiksi 9464899F-BA09-426C-AD60-F49C70F6AC42).
MySQL	SQL tietokannan hallintajärjestelmä.
SGML	Standard Generalized Markup Language (ISO 8879), vuonna 1986 kehitetty rakenteisten dokumenttien merkintäkieli, jonka pohjalta esimerkiksi XML on kehitetty.
SPSS	Statistical Package for Social Sciences, tuoteperhe joka tarjoaa erilaisia ratkaisuja tiedon keräämiseen ja analysointiin.
Unicode	Merkistöstandardi, joka kattaa suurimman osan maailman eri kielten käyttämistä merkeistä.
URI	Uniform Resource Identifier, merkkijono, jonka muotoisia esimerkiksi Internet-osoitteet (URL) ovat.
W3C	World Wide Web Consortium, kehittää erilaisia suosituksia ja teknologioita, tavoitteenaan ohjata Webin kehittymistä.
XLink	Määrittely XML-dokumenttien välisten linkitysten luomiseen.

XML eXtensible Markup Language, metakieli, jolla voidaan määritellä tiedon sisältö- ja rakennemuotoja.

XSL XML Stylesheet Language, tyylikieli XML-dokumenteille.

XSLT Extensible Stylesheet Language Transformations, muunnoskieli, joka mahdollistaa dokumenttien muuntamisen eri muotoihin.

TIIVISTELMÄ
ABSTRACT
SYMBOLILUETTELO

SISÄLLYS

1 JOHDANTO	1
2 XML	3
2.1 Taustaa	4
2.2 Tiedon rakenteistaminen ja rakenteiset dokumentit	5
2.3 XML:n käyttökohteet	6
2.4 XML-dokumentti	7
2.5 XML-jäsenin ja -proessori	8
2.6 Nimiavaruus	10
2.7 XML-dokumentin tyyppimäärittely	11
2.7.1 DTD	11
2.7.2 XML-Skeema	15
2.8 Ohjelmointirajapinnat	16
2.8.1 DOM	17
2.8.2 SAX	17
2.9 XML .NET -ympäristössä	18
3 TILASTOLLINEN TUTKIMUS	20
3.1 Tutkimusprosessi	21
3.2 Kerätyn aineiston analysointi	22

4 LUOVA LOMAKE ANALYSOINTITYÖKALUN TOTEUTTAMINEN	25
4.1 Ohjelmointikielet ja työkalut	25
4.2 Vaatimusmäärittely	26
4.3 Työprosessi	28
4.3.1 Suunnittelu	28
4.3.2 Ohjelmointi	28
4.3.3 Testaus	29
4.4 Toteutunut ohjelmisto	30
4.4.1 Koodi	30
4.4.2 Käyttöliittymät	31
5 POHDINTA	38
LÄHTEET	40
LIITTEET	42

1 JOHDANTO

Opinnäytetyönä oli Luova Lomake -ohjelmistoon tehtävä analysointityökalu, jonka tehtävänä on Internet-kyselytutkimuksen vastaustietojen analysointi. Tutkimusmateriaalin perusteella Luova Lomake Analysointityökalulla käsitellään vastaustietoja ja niistä muodostetaan käyttäjän tekemän raporttipohjan perusteella reaaliaikainen HTML-pohjainen raportti. Luova Lomake Analysointityökalun tarkoituksena on keventää käyttäjän työmäärää analysoitaessa tehdyn kyselytutkimuksen vastausmateriaalia ja mahdollistaa tietojen esittäminen raporteissa erilaisten esitystapojen avulla.

Luova Lomake Analysointityökalun toteuttamiseen käytettiin Microsoft Visual Studio 2005 .NET -ohjelmointiympäristöä. Ohjelmointikielenä oli Visual Basic .NET ja ohjelmointialustana oli Microsoft'in ASP .NET 2.0. Kyselytutkimusten vastausmateriaali voi olla tallennettuna joko MySQL- tai Microsoft SQL Server 2005 -tietokantaan ja raporttipohjat tallennettiin XML-dokumentteihin.

Opinnäytetyön toimeksiantaja on Luovaliike Oy, jonka toimialana on ohjelmistotuotanto erilaisiin verkkoympäristöihin. Tämän työn toiminta-alustana on Internet-selain ja ohjelmaa käytetään Internet'in välityksellä. Luovaliike Oy:llä on erilaisia ohjelmistoja, joista voidaan asiakaskohtaisesti koota juuri oikeanlainen paketti.

Opinnäytetyön teoriaosuudessa käsitellään pääasiassa XML-tekniikoita, mutta oma lukunsa on myös tilastolliselle tutkimukselle. XML-tekniikat sisältävät erittäin laajan kokonaisuuden erilaisia tekniikoita, joten kaikkien osa-alueiden käyminen läpi opinnäytetyön puitteissa on mahdotonta. Pääasiassa XML:n teoriaosuudessa keskitytäänkin kaikkein olennaisimpiin ja

tähän työhön liittyviin asioihin. Tilastollisen tutkimuksen teoriaosuuden tarkoituksena on esitellä kyseisestä aiheesta perustietoja, jotta voidaan ymmärtää paremmin Luova Lomake Analysointityökalun tarkoitus ja toiminta.

XML sisältää suuren määrän erilaisia tekniikoita, joista tunnetuimpia lienee XML-dokumentti, XSL sekä dokumentin tyyppimäärittelyyn käytettävät DTD ja XML-Skeema. XML-dokumentti on luultavasti koko XML-perheen tunnetuin osa-alue. Tutkimalla XML-dokumenttia (Liite 1.) näkee XML:n ydinajatuksen eli tiedon rakenteisuuden sekä hierarkisen rakenteen.

XML on W3C:n kehittämä joukko erilaisia suosituksia, joita on useita eri osa-alueille. Kaiken perustana on suositus Extensible Markup Language (XML) 1.0, jossa määritellään koko XML:n perusta. Kaikki W3C:n XML:ään liittyvät suositukset löytyvät Internet-osoitteesta <http://www.w3.org/XML/Core/#Publications>.

2 XML

XML on rakenteisten dokumenttien merkintäkieli. Se on metakieli, jolla on mahdollista määritellä tiedon sisältö- ja rakennemuotoja. XML on tarkoitettu vain tallennettavan tiedon kuvaamiseen, eikä siinä määritellä tiedon esitysmuotoa. Seuraavassa muutamia XML:n perusideoita:

- XML on kehitetty rakenteisen tiedon esittämiseen.
- XML:n sisältämä tieto ei ole tarkoitettu sellaisenaan luettavaksi.
- XML sisältää monia eri teknologioita.
- XML mahdollistaa nimiavaruuksien käyttämisen.
- XML:n käyttö ei vaadi lisensoijaa ja se on järjestelmäriippumaton.
- XML tukee Unicode ja ISO/IEC 10646 -merkkijärjestelmiä. (Tuikka & Kanala 2001, 13 - 14; W3C Communications Team, 2003; Extensible Markup Language (XML) 1.0 (Fourth Edition), 2006.)

XML sisältää erilaisia osamäärittelyjä, joilla jokaisella on omat tehtävänsä. XML määrittelee syntaksin, XSL on XML:ssä käytettävä tyylikieli, joka määrittelee XML-dokumentin esitystavan ja XLink määrittelee eri dokumenttien väliset linkitykset. (Tuikka & Kanala 2001, 14.)

2.1 Taustaa

XML lähdettiin kehittämään aiemmin käytössä olleen SGML:n seuraajaksi. Syynä kokonaan uuden merkkaukielen tarpeeseen voidaan pitää SGML:n monimutkaisuutta ja sitä kautta liiallista ”jäykkyyttä” pieniin ohjelmistokehitysprojekteihin. Tavoitteeksi XML:n kehitykseen otettiin muun muassa:

- käyttö Internet:in yli
- laitteisto- ja ohjelmistoriippumattomuus
- SGML yhteensopivuus
- ohjelmoinnin helppous
- valinnaisten ominaisuuksien pieni määrä
- syntaksin luettavuus ja selkeys
- ei pyritä merkkauksen lyhyteen. (W3C a, Nykänen, O. 2001, 10 - 12; Extensible Markup Language (XML) 1.0 (Fourth Edition), 2006.)

XML on suunniteltu pääasiallisesti verkossa välitettävän tiedon esittämiseen, mutta se on käyttökelpoinen myös muuhun tiedonsiirtoon ja -tallentamiseen erilaisissa ohjelmistoissa. XML ei ole varsinaisesti standardi, vaan sen kehittänyt W3C puhuu suosituksista (recommendation). XML koostuu useista alateemoista, esimerkiksi XML-syntaksi, dokumenttien rakennemäärittelyt, nimiavaruudet ja XML-Skeemat. (Nykänen, O. 2001, 1, 8; Holzner, S. 2001, 13.)

Extensible Markup Language (XML) 1.0 -määrittely hyväksyttiin standardin mukaiseksi suositukseksi (W3C recommendation) helmikuussa 1998 ja sen jälkeen on määrittelystä ilmestynyt useita päivitettyjä versioita. Määrittelystä on kokonaan erillinen versio, Extensible Markup Language (XML) 1.1, jossa on kuitenkin vain pieniä eroavaisuuksia versioon 1.0 verrattuna. Eroavaisuudet ovat lähinnä korjauksia Unicode:n suhteen. Uusin versio määrittelystä on Extensible Markup Language (XML) 1.0 (Fourth Edition), joka on julkaistu elokuussa 2006. Tämä ydinmäärittely luo perustan muille XML-standardiperheen määrittelyille (esimerkiksi XML Namespaces, XML Path Language (XPath) ja Extensible Stylesheet Language Transformations (XSLT)). XML -määrittely ja kaikki muut W3C:n tekemät suositukset löytyvät Internet:stä osoitteesta <http://www.w3.org>. (Nykänen, O. 2001, 8; W3C a; W3C b; Extensible Markup Language (XML) 1.0 (Fourth Edition); Extensible Markup Language (XML) 1.1 (Second Edition).)

2.2 Tiedon rakenteistaminen ja rakenteiset dokumentit

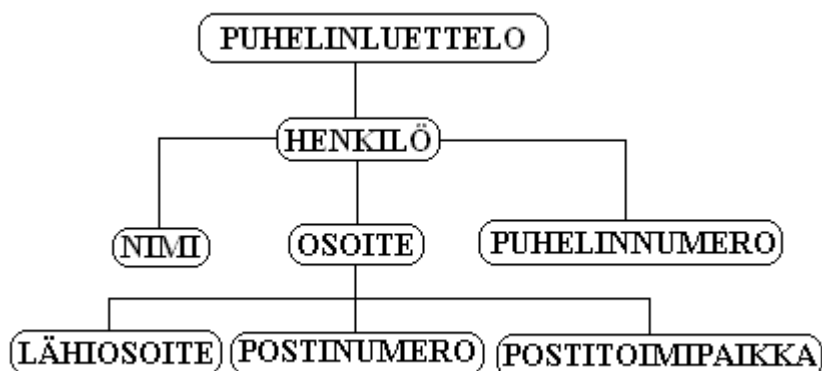
Tiedon rakenteistaminen on tärkeää, kun halutaan etsiä tietoa entistä suuremmista tietomassoista. Tiedon rakenteistamisessa dokumentin sisältö tallennetaan dokumentin rakenneosiin, joita käsitellään myöhemmin tässä työssä. Tässä yhteydessä dokumentilla tarkoitetaan tietyssä esitysmuodossa olevaa tietokokonaisuutta. Jokaisella dokumentin rakenneosista on oma tunnistemerkintä, joka on tietoa tiedosta eli metatietoa. (Tuikka & Kanala 2001, 3 - 5.)

Dokumentti voidaan jakaa eri osiin, jolloin sisältö, rakenne ja esitystapa saadaan eroteltua, ja muun muassa dokumentin ulkoasun määrittely ja muokkaaminen helpottuvat. Sisältö tarkoittaa dokumentin sisältämää tietoa, esimerkiksi tekstiä tai kuvia. Rakenne määrittelee dokumentin sisällön sijainnin dokumentissa. Esitystapa määrittää dokumentin visuaalisen ulkoasun. (Tuikka & Kanala 2001, 3 - 5; Lipitsäinen 2001.)

Tiedon rakenteistaminen mahdollistaa sen, että tietoa voidaan helposti hakea esimerkiksi XML-dokumentista tai HTML-sivuilta rakenneosan tunnistemerkinnän avulla. Lisäksi rakenteiset dokumentit ovat laitteisto- ja käyttöjärjestelmäriippumattomia. Tämä mahdollistaa sen, että saman rakenteisen dokumentin tietoja voidaan käyttää eri järjestelmistä, jolloin sama tieto voi olla usean järjestelmän käytössä samanaikaisesti. (Tuikka & Kanala 2001, 5; Lipitsäinen 2001.)

Rakenneosat muodostavat hierarkisen puurakenteen, joka sisältää erilaisia elementtejä. Dokumentissa luodaan juurielementti, jonka sisällä voi olla useita muita elementtejä, joita kutsutaan ali- tai lapsielementeiksi. Juurielementtejä ei voi olla dokumentissa kuin yksi ja kaikki muut elementit alkavat ja päättyvät sen sisällä. (Tuikka & Kanala 2001, 7.)

Tiedon rakenteistaminen voidaan tehdä kaikelle tiedolle, mutta parhaiten se soveltuu luetteloidulle tiedolle, esimerkiksi puhelinluettelotiedolle (Kuvio 1.).



Kuvio 1. Hierarkinen puurakenne

2.3 XML:n käyttökohteet

XML:n käyttökohteita on monenlaisia. XML:ää käytetään erilaisten tietojen tallentamiseen, joista parhaiten XML soveltuu luettelotyypisen tiedon tallentamiseen. XSL-tyylikielellä voidaan määrittää XML:n esitystapa.

XML:ää voidaan käyttää esimerkiksi Internet-sivustoilla, jolloin esitettävä tieto voidaan tallentaa esimerkiksi XML-dokumenttiin ja tiedon esitystapa määritellään XSL- tai CSS-tyylikielellä, joista XSL on nimenomaan XML:ää varten kehitetty. XSL-tyylikieli on monipuolisempi kuin CSS, joka on tarkoitettu lähinnä HTML-dokumenttien ulkoasun kuvaamiseen. XSL:llä voidaan myös laatia taulukoita sekä suodattaa ja lajitella XML-dokumentin sisältämää tietoa. XML:ää käytetään myös sovellusten rajapinnoissa, jolloin tieto lähetetään XML-muodossa vastaanottavalle sovellukselle esimerkiksi Internetin kautta. (Tuikka & Kanala 2001, 37.)

XML-dokumenttia voidaan myös käyttää pelkästään tiedon tallennusyksikkönä, jolloin sen sisältämää tietoa käytetään esimerkiksi siten, että se luetaan ohjelmallisesti ja käytetään vain XML-dokumentin sisältämää tietoa. Esimerkiksi Microsoft SQL Server 2005 tietokannassa on mahdollista tallentaa tietokannan kenttään tietoa myös XML-muodossa.

2.4 XML-dokumentti

XML-dokumentti (Liite 1) sisältää tekstipohjaista tietoa, joka voidaan avata ja muokata esimerkiksi tekstieditorilla. XML-dokumentti jaetaan rakenteeltaan kahteen osaan, loogiseen ja fyysiseen. Looginen rakenne edellyttää, että dokumentin rakenne toteutettu oikein. Se sisältää elementit, kommentit, merkkiviittaukset ja prosessointiohjeet. Dokumentin tulee sisältää yksi juurielementti, jonka sisällä muut elementit alkavat ja päättyvät. Fyysinen rakenne puolestaan koostuu entiteeteistä, jotka ovat käytännössä tiedonsäilytysyksiköitä. (Extensible Markup Language (XML) 1.0 (Fourth Edition), 2006; Tuikka & Kanala 2001, 15 - 16.)

XML-dokumentti sisältää aina vähintään yhden elementin, joka on juurielementti. Kaikki muut elementit alkavat ja päättyvät sen sisällä. Jokainen elementti alkaa ja päättyy saman emoelementin sisällä. Elementti alkaa alkutunnisteella, `<elementinnimi>` ja päättyy lopputunnisteeseen, `</elementinnimi>`. Näiden tunnisteiden väliin tulee kaikki elementtiin kuuluva tieto. Tyhjä elementti voidaan myös merkitä, `<elementinnimi/>`, jolloin ei tarvita erillistä alku- ja lopputunnistetta. Elementti voi sisältää toisia elementtejä, attribuutteja ja tekstiä. (Extensible Markup Language (XML) 1.0 (Fourth Edition), 2006; Tuikka & Kanala 2001, 18 - 20; Lipitsäinen, A. 2001.)

Attribuutit antavat lisätietoa elementistä tai sen sisällöstä. Attribuutti koostuu nimi-arvo-parista ja se sijaitsee elementin alkutunnisteessa, `<elementinnimi attribuutinnimi="arvo">`. On olemassa myös yleisattribuutteja, esimerkiksi dokumentin kielen määrittelyä varten. Yleisattribuutin `xml:lang` avulla voidaan määrittää dokumentin sisällön ja attribuuttien arvojen kieli. (Tuikka & Kanala 2001, 18 - 20; Lipitsäinen, A. 2001; Holzner, S. 2001, 85 - 86.)

Entiteetti on virallisessa XML-kielen määrittelyssä säilytysobjekti. Entiteetit voivat olla ulkoisia, jolloin samoja entiteettejä voidaan käyttää useissa dokumenteissa. Tärkein entiteetti on dokumenttientiteetti, tai toiselta nimeltään juurientiteetti. Entiteetti voi olla esimerkiksi ulkoinen kuvatiedosto, johon vain viitataan XML-dokumentista. Erikoimerkkientiteetit sisältävät merkit `<`, `>`, `&`, `'` ja `"`. Nämä merkit esiintyvät XML-syntaksissa, joten siksi niitä ei voi käyttää muuhun tarkoitukseen, vaan ne on korvattava

erilaisella merkintätavalla. (Tuikka & Kanala 2001, 15 - 16, 20 - 21; North & Hermans 2000, 33, 44.)

XML-dokumentin tulee olla hyvin muodostettu (well-formed) tai validi (valid), jotta se ylipäättään voi olla XML-dokumentti. Hyvin muodostettu XML-dokumentti on XML-syntaksin mukainen ja täyttää XML-määrittelyn ehdot. Hyvin muodostettu XML-dokumentti ei välttämättä ole validi, mutta jos se on validi, sen täytyy olla myös hyvin muodostettu. Validin dokumentin täytyy noudattaa sille DTD:ssä tai XML-Skeemassa määritettyjä rakennemäärittelyjä. (Extensible Markup Language (XML) 1.0 (Fourth Edition), 2006; Nykänen, O. 2001, 122; Tuikka & Kanala 2001, 15; Lipitsäinen, A. 2001.)

XML-dokumentin sisältö voi näyttää esimerkiksi seuraavanlaiselta:

```
<?xml version="1.0" standalone="yes" encoding="ISO-8859-1" ?>
<JuuriElementti>
    <Elementti1 attribuutti1="arvo">
        Elementin sisältö...
    </Elementti1>
</JuuriElementti>
```

Esimerkissä annetaan ensimmäisellä rivillä määritelmä. Määritelmä ei ole pakollinen XML-dokumentissa. Määritelmä on prosessointiohje, jonka erottaa kysymysmerkistä rivin alussa ja lopussa. Attribuutti *version* kertoo mitä versiota XML-standardista käytetään. Attribuutti *standalone* voi olla arvoltaan *yes* tai *no*. Mikäli dokumentissa ei viitata ulkoisiin entiteetteihin, arvona tulee olla *yes*. Attribuutti *encoding* kertoo dokumentissa käytettävän kielikoodauksen. (North & Hermans, 2000, 28 - 29; Holzner, S. 2001, 76.)

2.5 XML-jäsennin ja -proessori

XML-jäsennin, toiselta nimeltään parseri on ohjelma, joka lukee XML-dokumentin, tarkistaa sen ja raportoi havaitsemansa virheet. Jäsentimet voivat olla hyvinkin erilaisia. Yksinkertaisimmillaan jäsennin tunnistaa XML-dokumentin kielioppivirheet, kun taas monipuolisemmat jäsentimet edellä mainitun lisäksi validoivat dokumentin rakenteen

annetun tyyppimäärityksen (DTD tai Skeema) mukaisesti. Taulukossa 1. on erilaisia jäsentimiä jaoteltuna XML-termien mukaisesti. (Nykänen, O. 2001, 52 - 53.)

Tyyppi	Kuvaus
ei-validoiva jäsenin	Tunnistaa XML-kielioppivirheet.
nimiavaruudet tunnistava jäsenin	Tunnistaa XML-kielioppivirheet sekä osaa nimiavaruudet huomioiden jäsentää XML-dokumentin.
validoiva jäsenin	Tunnistaa XML-kielioppivirheet ja validoi dokumentin loogisen rakenteen annetun tyyppimäärityksen (DTD) mukaisesti.
skeema-validoiva jäsenin	Tunnistaa XML-kielioppivirheet ja validoi dokumentin loogisen ja leksaalisen rakenteen annetun tyyppimäärityksen (Skeema) mukaisesti.

Taulukko 1. Erilaisia XML-jäsentimiä. (Nykänen, O. 2001, 53.)

XML-prosessorin tehtävänä on XML-dokumentin lukeminen ja tiedon jäsentäminen seuraavalle ohjelma moduulille, eli XML-sovellukselle (Kuvio 2.). Extensible Markup Language (XML) 1.0 -määrittely esittelee myös XML-prosessorin käyttäytymisen ja virhetilanteiden kuvailun. Erona XML-jäsentimellä ja -prosessorilla on se, että XML -määrittely ei kerro kuinka jäsenitys prosessoriohjelman sisällä teknisesti toteutetaan, eikä myöskään sitä minkälainen tietorakenne jäsentimessä valmistellaan. XML -määrittelyssä esitellään erityisesti peruuttamattomat virhetilanteet, joista XML-prosessori ei saa toipua. Tämä on tärkeää erityisesti siksi, että XML-dokumenttien hyvin muodostuneisuus säilyisi. (Extensible Markup Language (XML) 1.0 (Fourth Edition), 2006; North & Hermans, 2000, 31; Nykänen, O. 2001, 93 - 95.)

XML-prosessori lähdedokumentti	<i>jäsenitys</i> standardi rajapinta	XML-sovellus käyttö
--	---	-------------------------------

Kuvio 2. XML-prosessorin ja -sovelluksen työnjako. (Nykänen, O. 2003.)

2.6 Nimiavaruus

Nimiavaruus tarkoittaa kokonaisuutta, jonka sisällä ei voi olla päällekkäisiä nimiä. Nimiavaruuksia käytetään, jotta esimerkiksi erilaisissa muuttujien nimeämisissä voitaisiin käyttää mahdollisimman selkeitä ja lyhyitä nimiä. Nimiavaruuksien erottamiseen toisistaan käytetään jonkinlaista teknistä tunnistetta, jonka avulla ohjelmat ymmärtävät mitä nimiavaruutta milloinkin käytetään. (Walkama & Laakkonen 2004, 9; Nykänen, O. 2001, 182.)

XML-nimiavaruudet on luotu ehkäisemään erilaisia tunnistamisongelmia ja ristiriitoja. XML-dokumentin nimiavaruuden tulee olla URI-syntaksin mukainen merkkijono. Tämä ei kuitenkaan tarkoita, että nimiavaruuden osoittamassa kohdassa verkkoa tulisi olla mitään. XML-nimiavaruus esitellään XML-dokumentissa elementin aloitustunnisteessa `xmlns`-attribuutilla, esimerkiksi `<elementti xmlns="http://www.kajak.fi/nimiavaruus">`. Edellä esitetyllä tavalla määritetään elementin oletusnimiavaruus joka periytyy myös kaikille elementin lapsielementeille. Haluttaessa käyttää muuta kuin oletusnimiavaruutta käytetään kvalifioituja nimiä, jolloin nimiavaruudelle annetaan lyhennemerkintä, jonka avulla kyseiseen nimiavaruuteen viitataan, `<es:elementti xmlns:es="http://www.kajak.fi/es">`. Esimerkissä kvalifioitu merkkijono on `es:elementti`, joista ennen kaksoispistettä oleva osa on nimiavaruuden prefiksi, eli nimiavaruuden lyhenne ja kaksoispisteen jälkeen on nimen lokaaliosa, eli elementin nimi. Määriteltäessä nimiavaruus `xmlns:es` -attribuutilla perivät kaikki elementin lapsielementit saman nimiavaruuden. Yleensä nimiavaruuksien prefiksit on selkeintä esitellä XML-dokumentin juurielementissä. (Tuikka & Kanala 2001, 34 - 36; Walkama & Laakkonen 2004, 18 - 19; Nykänen, O. 2001, 181 - 183, 186 - 188.)

W3C on antanut XML-nimiavaruuksista oman suosituksensa. Uusimmat versiot ovat Namespaces in XML 1.0 (Second Edition) ja Namespaces in XML 1.1 (Second Edition), jotka vastaavat versionumeroiltaan samansuuruisia Extensible Markup Language (XML) -spesifikaatioita. (W3C b.)

2.7 XML-dokumentin tyyppimäärittely

Yksi rakenteisen dokumentin ominaisuuksista on mahdollisuus määrittellä dokumentin rakenne tietyn mallin mukaiseksi. Tällöin dokumenttiin liitetään dokumenttityypin määrittely, joka voidaan toteuttaa esimerkiksi DTD:llä tai XML-Skeemalla. Dokumenttityypin määrittely pakottaa dokumentin tiettyyn muotoon, jolloin sitä on helpompi käsitellä. Dokumentti on validi, kun se noudattaa sille DTD:ssä tai XML-Skeemassa annettua rakennemäärittelyä. (Tuikka & Kanala 2001, 23; Holzner, S. 2001, 109; Nykänen, O. 2001, 122.)

XML:n rakenneosia ovat elementit, attribuutit ja entiteetit. Entiteetit määritellään DTD-kuvauksissa, ja XML-Skeemaa käytettäessä ne eivät ole käytössä, muuten kuin erikoismerkkientiteettien muodossa. (Tuikka & Kanala 2001, 18 - 20; Walkama & Laakkonen 2003, 237 - 238.)

2.7.1 DTD

DTD (Document Type Definition), eli dokumenttityypimäärittely oli yleisin XML-dokumenttien rakenteen kuvaukseen käytetty tekniikka ennen XML-Skeemaa. DTD-kuvaukset on kirjoitettu SGML-kielillä. Ne kuvaavat dokumentin rakenteen, elementtien väliset suhteet sekä elementtien nimet ja sisällöt. DTD soveltuu parhaiten tekstidokumentin rakenteen määrittelyyn, koska sen ominaisuudet eivät ole riittävät kaikkiin XML:n käyttöalueisiin. DTD voi olla joko sisäinen tai ulkoinen. Sisäinen DTD on kirjoitettu XML-dokumenttiin ja ulkoinen DTD on puolestaan kokonaan eri tiedostossa. (Lipitsäinen, A. 2001, 7; Tuikka & Kanala 2001, 23, 29; Nykänen, O. 2001, 123.)

DTD:ssä dokumentin rakenne ja sisällöt määritellään tarkasti, jolloin kaikki dokumenttiin tulevat elementit ja attribuutit ovat mukana määrittelyssä. Määrittelyssä käy ilmi jokaisen elementin sisältö eli sisältääkö elementti toisia elementtejä, attribuutteja vai sisältääkö se pelkkää tekstiä. Lisäksi DTD kertoo mitkä elementit ja attribuutit ovat pakollisia. (Holzner, S. 2001, 109.)

Esimerkki sisäisestä DTD:stä:

```
<?xml version="1.0" standalone="yes" encoding="ISO-8859-1"?>
<!DOCTYPE JuuriElementti [
    <!ELEMENT JuuriElementti (Elementti)>
    <!ELEMENT Elementti (#PCDATA)>
    <!ENTITY PVM "30.4.2007">
] >
<JuuriElementti>
    < Elementti>Tekstiä... </ Elementti>
    < Elementti &PVM;>Tekstiä... </ Elementti>
</JuuriElementti>
```

Edellisessä esimerkissä on DTD:llä määritetty dokumentin rakenne. !DOCTYPE -elementti luo rakenne-esittelyn. Hakasulkujen välissä on varsinainen rakennemäärittely. Esimerkissä rakennemäärittelyssä luodaan juurielementin sisään elementti, joka voi sisältää merkkipohjaista dataa, eli tekstiä. !ENTITY -elementissä esitellään entiteettimääritelmä, joka korvaa dokumentissa olevat entiteettiviittaukset. (Holzner, S. 2001, 110 - 118, 156 - 157.)

DTD:n dokumentin tyyppimäärittely voi sisältää seuraavia osa-alueita; notaatiomäärittely (notation declaration), elementin tyyppimäärittely (element type declaration), attribuutilistan tyyppimäärittely (attribute-list declaration), entiteettimäärittely (entity declaration). Edellä mainittujen osa-alueiden lisäksi tyyppimäärittely voi sisältää myös muita merkkauksia, esimerkiksi kommentteja ja prosessointiohjeita. (Nykänen, O. 2001, 124 - 125.)

Notaatio tarkoittaa merkintää, joka liitetään dataentiteetin määrittelyn yhteyteen joko notaatio-tyyppisen attribuutin arvoksi tai prosessointiohjeen tueksi. Notaatiomäärittely kirjoitetaan aina DTD:n sisään ja se sisältää joko prosessointikohtaisen tai julkisen notaatioarvon. Notaatio määritellään seuraavan esimerkin mukaisesti, *<!NOTATION konverteri SYSTEM "converter.exe">*. (Nykänen, O. 2001, 125.)

Elementin tyyppimäärittelyssä ilmoitetaan elementin nimi ja tyyppi. Elementtityyppejä on neljä erilaista; tyhjä, elementtisisältöinen, yhdistelmä ja rajoitteeton. Tyhjä elementti voidaan määrittää seuraavan esimerkin mukaisesti, *<!ELEMENT tyhjä EMPTY>*, tällainen

elementti ei voi sisältää mitään tietoa. Elementtisisältöinen elementti voidaan määrittellä esimerkiksi, `<!ELEMENT juuri (lapsi1, lapsi2, lapsi3)>`, tämä elementti voi sisältää määrittelyn mukaisia lapsielementtejä. Yhdistelmäsisisältöinen elementti voidaan määrittellä esimerkiksi, `<!ELEMENT kappale (#PCDATA|termi|huomautus)*>`, tällainen elementti voi sisältää vapaata tekstisisältöä sekä vapaan määrän lapsielementtejä termi tai huomautus. Rajoitteeton elementti voi sisältää mitä tahansa dokumentissa sallittuja elementtejä tai merkkejä ja se voidaan määrittellä seuraavasti, `<!ELEMENT rajoitteeton ANY>`. (Nykänen, O. 2001, 126 - 132.)

Attribuutit ovat elementtien määreitä. Niiden tyyppimäärittely tehdään samoin kuin elementtienkin, mutta attribuuteilla ei ole hierarkista rakennetta. Attribuuttien tyyppimäärittely voidaan tehdä yksittäin tai attribuutilistana. Attribuutilistan tyyppimäärittely voi olla esimerkiksi seuraavanlainen, `<!ATTLIST elementti kieli (suomi/ruotsi/englanti) "suomi">`. Esimerkki kertoo mihin elementtiin attribuutti kuuluu, attribuutin nimen ja arvo vaihtoehdot sekä attribuutin oletusarvon. Attribuutilista voidaan myös määrittää eri tarpeisiin, `<!ATTLIST elementti kieli (suomi/ruotsi/englanti) #REQUIRED>`, pakollinen attribuutti, `<!ATTLIST elementti kieli (suomi/ruotsi/englanti) #IMPLIED>`, ei pakollinen, mutta XML-prosessori kuitenkin välittää tiedon attribuutin olemassa olostasovellukselle, `<!ATTLIST elementti kieli CDATA #FIXED "suomi">`, pakottaa attribuutin kiinteäksi. Esimerkeissä käytetty attribuutin tyyppi on ns. lueteltu attribuutti eli attribuutin mahdolliset arvot on annettu tyyppimäärittelyssä. Attribuuteilla on myös muita tyyppisiä (Taulukko 2.). (Nykänen, O. 2001, 131 - 134.)

Tyyppinimi	Kuvaus arvosta
CDATA	merkkidataa
lueteltu (enumerated)	sallitut arvot on lueteltu tyyppimäärittelyssä
notaatio	notaation nimi
ID	yksikäsitteinen XML-nimi
IDREF	viittaus ID-tyyppisen attribuutin arvoon
IDREFS	lista viittauksia ID-tyyppisten attribuuttien arvoihin
ENTITY	entiteetin nimi
ENTITIES	lista entiteettien nimiä
NMTOKEN	tunnistemerkkijono
NMTOKENS	lista tunnistemerkkijonoja

Taulukko 2. XML DTD:n attribuuttien tyytit. (Nykänen, O. 2001, 134.)

Entiteetit ovat tietosäiliöitä, joiden avulla XML-dokumentteihin liittyvä teksti ja data esitetään. Entiteetit jaetaan kahteen pääluokkaan; tekstientiteetit ja dataentiteetit. Tekstientiteetit sisältävät XML-tekstiä. Dataentiteetit voivat sisältää mitä tahansa dataa, jota dokumentissa ei kuitenkaan jäsennetä. XML:ssä on kuusi erilaista entiteettityyppiä:

- dokumenttientiteetti (document entity)
- sisäinen tekstientiteetti (internal general entity)
- ulkoinen tekstientiteetti (external general entity)
- dataentiteetti (unparsed entity)
- sisäinen parametrientiteetti (internal parameter entity)
- ulkoinen parametrientiteetti (external parameter entity). (Nykänen, O. 2001, 141.)

Dokumenttientiteettiä ei koskaan määritellä erikseen, vaan se liittyy suoraan dokumentin määrittelyyn. Entiteeteillä on aina ominaisuudet nimi ja sisältö samaan tapaan kuin elementeilläkin. Entiteettien tyyppimäärittelyn perusteella entiteetti saa jonkin yhdistelmän eri ominaisuuksista:

- sisäinen tai ulkoinen (internal vs external)
- yleinen tai parametri (general vs parameter)
- jäsennetty tai jäsentämätön (parsed vs unparsed). (Nykänen, O. 2001, 142.)

Sisäisiä tekstientiteettejä käytetään esimerkiksi vakiomerkkijonojen määrittelyyn. Määrittely voi olla esimerkiksi, `<!ENTITY oppilaitos "Kajaanin amk">`, jolloin dokumentin esiintymäosassa voidaan entiteettiviittauksella `&oppilaitos;` saada korvaustekstinä entiteetin literaaliarvo (esimerkin literaaliarvo on Kajaanin amk). Ulkoiset tekstientiteetit sijaitsevat fyysisesti kokonaan eri tiedostossa, jonka määrittely voi olla esimerkiksi, `<!ENTITY johdanto SYSTEM "esim.xml">`. Entiteettiviittaus on ulkoisessa tekstientiteetissä samanlainen kuin sisäisessä tekstientiteetissäkin. (Nykänen, O. 2001, 143 - 144.)

Dataentiteetti on ulkoinen, yleinen ja jäsentämätön entiteetti, joka on aina jokin ulkoinen resurssi. Dataentiteetille merkitään vain URI-muotoinen viittaus ja siihen liittyy aina notaatio. (Nykänen, O. 2001, 145 - 146.)

Sisäisiä ja ulkoisia parametrientiteettejä käytetään dokumentin tyyppimäärittelyn sieventämiseen. Parametrientiteetit kirjoitetaan DTD:hen ja ne käsitellään siellä. Sisäisiä parametrientiteettejä käytetään makrotyyppisten operaatioiden toteuttamiseen. Niitä

käytetään myös ehdollisten DTD-lohkojen soveltamisessa. Ulkoiset parametrititeetit mahdollistavat DTD:n kokoamisen useammasta kuin kahdesta osasta. (Nykänen, O. 2001, 146 - 148.)

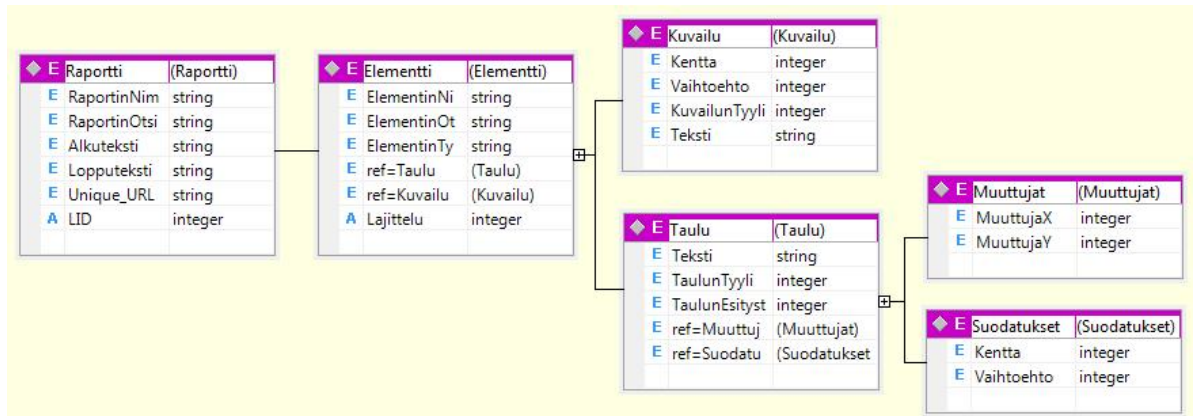
2.7.2 XML-Skeema

XML-Skeema on W3C:n kehittämä skeemakieli, jolla on mahdollista tehdä tarkempi rakennemäärittely kuin DTD:llä. XML-Skeema noudattaa XML:n syntaksia, joten se helpottaa ohjelmistokehittäjien työtä, koska rakennemäärittelyä varten ei tarvitse hallita eri ohjelmointikieltä. XML-Skeeman avulla on mahdollista määrittellä elementtien välille suhteita ja käyttää haluttuja tietotyyppisiä. W3C on tehnyt myös XML-Skeemalle määrittelyjä, joita kolme erilaista; XML Schema Part 0: Primer, XML Schema Part 1: Structures ja XML Schema Part 2: Datatypes. Näistä ensimmäinen kuvaa skeemojen perusteet ja jälkimmäiset kuvaavat kuinka skeemoja tulee käyttää. Seuraavassa luettelossa on esitelty XML-Skeeman etuja verrattuna DTD:hen:

- XML-Skeemassa voidaan käyttää tehokkaasti nimiavaruuksia.
- XML-Skeema kuvaukset ovat laajennettavissa, jolloin voidaan käyttää olio-ohjelmoinnissa käytettävää perintää.
- XML-Skeema mahdollistaa tietotyyppien käytön, jolloin sen käyttö perinteisissä ohjelmointikielissä on helpompaa.
- XML-Skeema mahdollistaa dokumentin oletussisällön määrittämisen.
- XML-Skeemassa on mahdollista esitellä suoraan myös yksittäisiä elementtejä. (Tuikka & Kanala 2001, 29 - 30; Lipitsäinen, A. 2001, 7 – 8; Nykänen, O. 2001, 150 - 151.)

XML-Skeema määrittelyssä elementin sisältömalli voi olla tyhjä, yksinkertainen tai monimutkainen. Tyhjä sisältömalli tarkoittaa, ettei elementti oleteta sisältävän tekstiä tai lapsielementtejä. Yksinkertainen sisältömalli voi sisältää tekstisolmuja ja monimutkainen

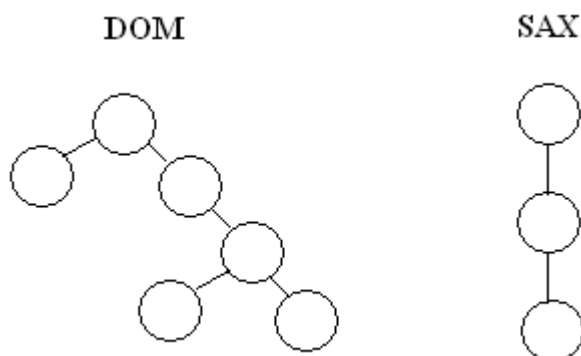
sisältömalli voi sisältää lapsielementtejä, mutta ei tekstiä. Näiden lisäksi on olemassa sekoitettu sisältömalli, joka voi sisältää sekä tekstiä että lapsielementtejä. Liite 1:stä nähdään millainen XML-dokumentti luodaan Kuvio 3:n mukaisesta XML-Skeemasta. (Walkama & Laakkonen 2004, 67.)



Kuvio 3. Microsoft Visual Studio 2005:lla määritetty XML-Skeema.

2.8 Ohjelmointirajapinnat

XML-dokumentti on yhteydessä pelkästään rajapintaan ja kaikki tiedonsiirto suoritetaan vain sen kautta. Rajapinnat mahdollistavat sen, että XML on ohjelmisto- ja laitteistoriippumaton. Rajapinta on käytännössä ohjelmistoluokka, joka sisältää erilaisia metodeja XML-dokumentin käsittelyyn. Tärkeimmät XML-ohjelmointirajapinnat ovat DOM (Document Object Model) ja SAX (Simple Api for XML) (Kuvio 4.). (Tuikka & Kanala 2001, 91.)



Kuvio 4. DOM ja SAX ohjelmointirajapintojen rakenne (Inkinen 2003, 186)

2.8.1 DOM

DOM on W3C:n määrittämä oliomalli, jota käytettäessä XML-dokumentista muodostetaan hierarkinen puurakenne. Tällöin XML-dokumentin käsittely on helppoa, koska silloin voidaan ohjelmallisesti viitata mihin tahansa kohtaan puurakennetta. DOM:in huonopuoli on se, että se käyttää paljon resursseja, koska koko XML-dokumentti tallennetaan muistiin. (Nykänen, O. 2001, 60, 65; Holzner, S. 2001, 307; Tuikka & Kanala 2001, 91; Inkinen 2003, 186.)

W3C:llä on kolme eri tasoa DOM suositusta koskien. Taso 1 määrittelee rajapinnan HTML- ja XML-dokumenttien käsittelyyn. Taso 2 laajentaa taso 1:n suositusta muun muassa tyyliedostoihin sekä HTML:ään liittyen. Taso 3 määrittää tarkemmin lataamisen ja tallentamisen sekä validoinnin. Kaikki DOM:iin liittyvät suositukset ja dokumentit löytyvät W3C:n Internet-sivuilta osoitteesta <http://www.w3.org/DOM/DOMTR>. (Holzner, S. 2001, 308 – 309; W3C c.)

DOM tekee dokumentista solmupuun, jossa jokainen tietoalkio on solmu. Elementin sisältämistä solmuista tulee elementin alisolmuja. Dokumentin jokaisesta osasta tulee solmu. W3C:n määrittämiä mahdollisia DOM:in solmuja ovat muun muassa:

- elementti
- attribuutti
- teksti
- CDATA-lohko
- entiteetti viittaus
- entiteetti
- prosessointiohje. (Holzner, S. 2001, 307 - 308.)

2.8.2 SAX

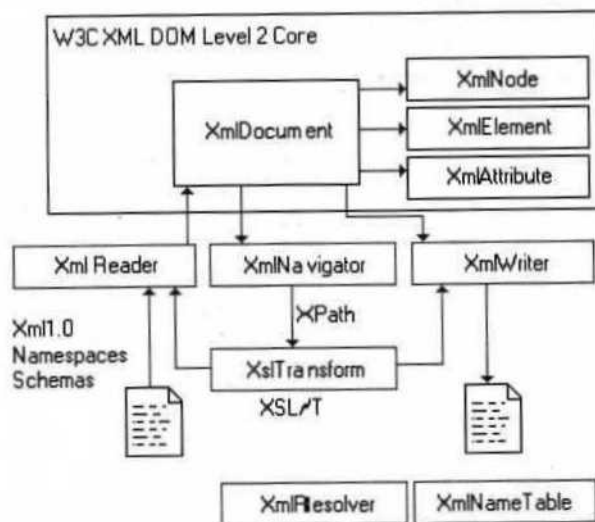
SAX on toinen yleisesti käytetty rajapinta. SAX ei ole virallinen suositus eikä standardi. Se on syntynyt erään postituslistan jäsenten tekemänä. SAX on suunniteltu alkujaan Java:lle, mutta SAX2:den myötä se on laajentunut myös muille ohjelmointikielille. SAX -projektin

viralliset Internet-sivut löytyvät osoitteesta <http://www.saxproject.org>. (Megginson Technologies; SAX project.)

SAX ei tallenna koko XML-dokumenttia muistiin puuna, vaan tietoja käsitellään tapahtuma kerrallaan. Se on siis tapahtumaperustainen rajapinta. SAX:in etuna on, että se vaatii vähemmän resursseja ja on yksinkertaisempi käyttää kuin DOM. Toisaalta SAX:ssa dokumenttia voidaan käsitellä vain kulkemalla yksisuuntaisesti eteenpäin, joka tekee vaikeaksi varsinkin monimutkaisten hakujen käsittelyn. (Tuikka & Kanala 2001, 98 - 99; Holzner, S. 2001, 581; Inkinen 2003, 186.)

2.9 XML .NET -ympäristössä

.NET-ympäristöön huomioi XML:n käsittelyn vaatimat erityispiirteet ja sinne on lisätty luokkakokoelmat XML:ää varten. .NET tukee sekä DOM että SAX ohjelmointirajapintoja. Kaikki tärkeimmät XML:n käyttöön tarvittavista luokista on sijoitettu System.XML -perusluokkaan. System.XML -perusluokka sisältää muun muassa luokat XML-dokumentin lukemiseen ja kirjoittamiseen. Kuviossa 5. käy ilmi .NET:in XML-arkkitehtuuri. (Inkinen, V. 2003, 186 - 187; Walkama & Laakkonen 2004, 160.)



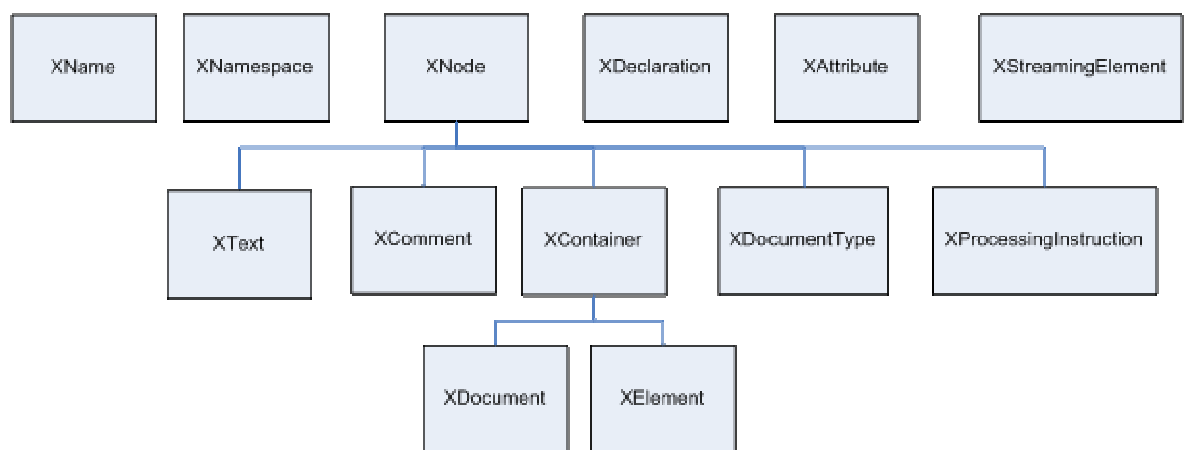
Kuvio 5. .NET:in XML-arkkitehtuuri (Walkama & Laakkonen 2004, 161)

Yleisimmin käytettyjä System.XML:n sisältämiä luokkia ovat:

- XmlDocument, luokka XML-dokumenttien käsittelyyn
- XmlTextReader, metodikokoelma XML-dokumentin sisällön lukemiseen
- XmlTextWriter, metodikokoelma XML-dokumentin sisällön kirjoittamiseen
- XmlElement, yksittäisen DOM-puun elementin perusluokka
- XmlNode, esittää DOM-puun solmun (voi sisältää alisolmuja)
- XmlNodeReader, DOM-mallin XML-solmun lukeminen
- XmlAttribute, perusluokka attribuuttien käsittelemiseen
- XmlTransform, XSLT-muunnoksien tekemiseen. (Inkinen, V. 2003, 187 - 188.)

XML-sisältöä ohjelmallisesti käsiteltäessä on ensin valittava tapa, jolla käsittely aiotaan suorittaa. Valinta DOM:in ja SAX:in välillä on kuitenkin yleensä helppo tehdä, mikäli tiedetään sovelluksen kulku ja XML-sisällön käyttötapa. Esimerkiksi mikäli koko XML-tiedoston sisältö aiotaan lukea, on SAX parempi valinta, koska sitä käyttäen saadaan tiedoston lukeminen suoritettua nopeammin ja tehokkaammin. (Inkinen, V. 2003, 189.)

Lähiaikoina julkaistava uusi Visual Studio tulee muuttamaan XML:n ja tietokantojen käsittelyä .NET -ympäristössä huomattavasti, kun Microsoft tuo uuden XQuery -luokan (Language Integrated Query for XML) korvaamaan aiemman System.XML:n. XQuery sisältää esimerkiksi luokat XDocument, XElement ja XAttribute (Kuvio 6.). (Microsoft Project LINQ 2006.)



Kuvio 6. XQuery luokkarakenne (Microsoft Project LINQ 2006, 7.)

3 TILASTOLLINEN TUTKIMUS

Tieteellinen tutkimus voidaan toteuttaa teoreettisena tai empiirisenä tutkimuksena. Empiirisen tutkimuksen toteutusvaihtoehdot ovat kvantitatiivinen eli määrällinen tai kvalitatiivinen eli laadullinen (Taulukko 3.). Kvantitatiivista tutkimusta voidaan kutsua myös tilastolliseksi tutkimukseksi. Tutkimusmenetelmä määrää yleensä kuinka tietoa kerätään, ja kuinka se käsitellään. (Heikkilä, T. 2004, 13 - 22.)

KVANTITATIIVINEN	KVALITATIIVINEN
vastaa kysymyksiin: Mikä? Missä? Paljonko? Kuinka usein?	vastaa kysymyksiin: Miksi? Miten? Millainen?
numeerisesti suuri, edustava otos	suppea, harkinnanvaraisesti koottu näyte
ilmiön kuvaus numeerisen tiedon pohjalta	ilmiön ymmärtäminen ns. pehmeän tiedon pohjalta

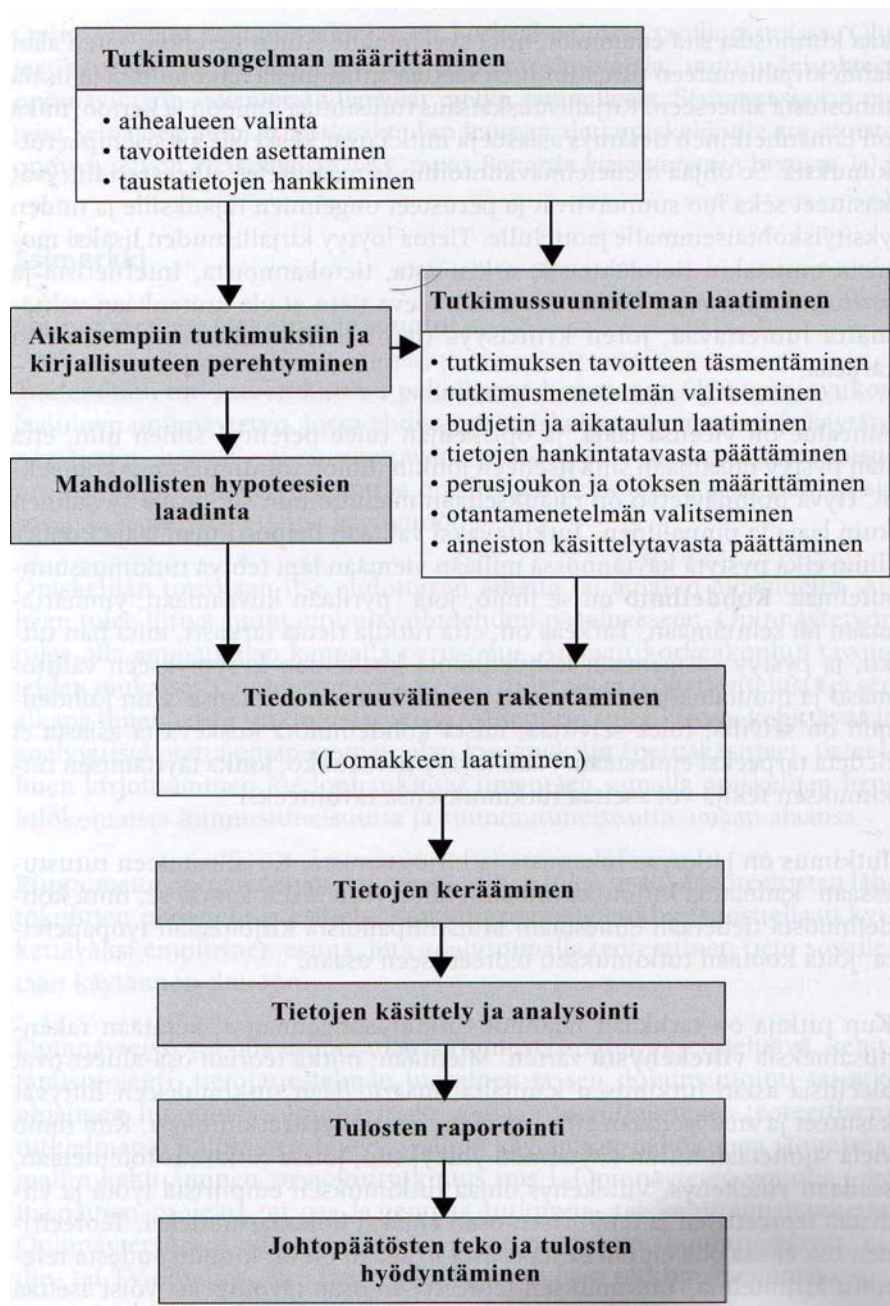
Taulukko 3. Kvantitatiivisen ja kvalitatiivisen tutkimuksen oleelliset erot (Heikkilä, T. 2004, 17.)

Kvantitatiivisen tutkimuksen avulla selvitetään lukumääriin ja prosentteihin liittyviä kysymyksiä sekä tutkitaan asioiden välisiä riippuvuuksia. Kvantitatiivisen tutkimuksen tulokset esitetään yleensä prosenttein ja taulukoin. Tämän tutkimusmenetelmän heikkoutena on, ettei se pysty välttämättä riittävästi selvittämään asioiden syitä, mutta toisaalta se mahdollistaa selkeän ja helppolukuisen tavan esittää tutkimustuloksia. (Heikkilä, T. 2004, 16 - 17.)

3.1 Tutkimusprosessi

Tarve tutkimukselle ilmenee jonkin kysymyksen / ongelman kautta. Tutkimusongelma voidaan esittää esimerkiksi seuraavanlaisilla kysymyksillä:

- Kuinka hyvin ammattikorkeakoulun opiskelijat työllistyvät opintojen jälkeen?
- Millaiset ovat yrittäjien tulevaisuuden näkymät?
- Mitkä tekijät vaikuttavat opiskelijoiden motivaatioon ammattikorkeakoulussa?
- Millaiset odotukset opiskelijoilla on opiskelujen jälkeisestä elämästään? (Heikkilä, T. 2004, 22 - 24.)



Kuvio 7. Kvantitatiivisen tutkimusprosessin vaiheet (Heikkilä, T. 2004, 25.)

Yleensä tutkimusongelma täytyy jakaa pienempiin osiin, alaongelmiin, joiden avulla saadaan selkeämpi kuva koko tutkimusongelmasta. Osa- tai alaongelmien kysymykset voidaan muotoilla väittäviksi eli hypoteeseiksi, joita voidaan käyttää myös tutkimuslomakkeella. (Heikkilä, T. 2004, 22 - 24.)

Tutkimusprosessi (Kuvio 7.) etenee tutkimusongelman selvittämisen jälkeen, tutkimussuunnitelman mukaisesti tutkimuslomakkeen laadinnalla, aineiston keräämisellä, tietojen käsittelyllä ja analysoinnilla sekä tulosten raportoinnilla. (Heikkilä, T. 2004, 24.)

3.2 Kerätyn aineiston analysointi

Kerättyä aineistoa voidaan analysoida monin eri tavoin, mutta tässä opinnäytetyössä keskitytään ainoastaan asioiden ilmaisemiseen muuttujia kuvailemalla ja kahden muuttujan välisien yhteyksien käsittelemiseen ristiintaulukoinnin avulla. Ristiintaulukoinnin tulokset voidaan ilmaista joko numeerisena tietona tai graafisessa esitysmuodossa.

Muuttuja tarkoittaa mitattavaa ominaisuutta, joka voi olla luokitteluasteikollinen (esimerkiksi sukupuoli), järjestysasteikollinen (esimerkiksi ikäluokka), välimatkaasteikollinen (esimerkiksi lämpötila) tai suhteasteikollinen (esimerkiksi bruttotulot).

Analysointimenetelmien kannalta on käyttökelpoista jakaa muuttujat seuraavasti:

- kategoriset muuttujat (luokittelu- ja järjestysasteikolliset)
- asteikolla mitatut muuttujat (esimerkiksi 5-portainen tyytyväisyysasteikko)
- määrälliset muuttujat (välimatka- ja suhteasteikolliset). (Taanila, A. 2006, 17 - 18.)

Muuttujien kuvailussa tyydytään tuomaan esille vain yksittäisissä vastauksissa olevaa tietoa, esimerkiksi prosenttimäärien avulla. Tällöin saadaan tutkittua vastaustietoa koskien tiettyä kyselyn kysymystä. Esimerkiksi ”*Kyselyyn vastanneista 44 % oli miehiä.*”.

Tietoa kuvailemalla tuodaan selkeästi ja ymmärrettävästi esille kuinka vastaajat ovat vastanneet tiettyyn tutkimuslomakkeen kysymykseen. Kuvailevalla tasolla tehty tulosten esittäminen on kuitenkin monesti liian suppea, ja esimerkiksi ristiintaulukoinnilla voidaan tuoda tutkimuksen tulokset monipuolisemmin esille.

Muuttujien välisiä yhteyksiä voidaan tutkia muun muassa ristiintaulukoinnin ja korrelaatiokertoimen avulla. Luova Lomake Analysointityökalun toteuttamisessa käytetään näistä ainoastaan ristiintaulukointia. (Heikkilä, A. 2004, 203, 210.)

Ristiintaulukoinnissa (Taulukko 4.) tutkitaan miten kaksi muuttujaa vaikuttavat toisiinsa. Muuttujista toinen on sarakemuuttuja ja toinen rivimuuttuja. Lisäksi oikeassa reunassa voivat olla rivisummat, ja alhaalla sarakesummat. (Heikkilä, T. 2004, 210.)

	Perhesuhde		
Palkkatyytyväisyys	Perheetön	Perheellinen	Kaikki
Erittäin tyytymätön	54,8 %	31,4 %	40,2 %
Tyytymätön	19,4 %	25,5 %	23,2 %
Ei tyytymätön muttei tyytyväinenkään	16,1 %	27,5 %	23,2 %
Tyytyväinen	9,7 %	13,7 %	12,2 %
Erittäin tyytyväinen	0,0 %	2,0 %	1,2 %
Kaikki	100,0 %	100,0 %	100,0 %

Taulukko 4. Esimerkki ristiintaulukoinnista (Taanila, A. 2007, 38).

Ristiintaulukointi toteutetaan yleensä siten, että sarakemuuttujaksi asetetaan riippumaton muuttuja (syy), esimerkiksi sukupuoli tai ikä ja rivimuuttujaksi asetetaan riippuva muuttuja (seuraus). Esimerkiksi taulukossa 1 riippumaton muuttuja on perhesuhde ja riippuva muuttuja on palkkatyytyväisyys. Aina ei kuitenkaan ole selvää kumpi muuttujista on riippumaton ja kumpi riippuva, näin on esimerkiksi jos verrataan kahden mielipidekysymyksen välistä suhdetta. Tällöin on harkittava, kumpi muuttuja on sarakemuuttuja ja kumpi rivimuuttuja. (Heikkilä, T. 2004, 210.)

Ristiintaulukoinnin tarkoitus on tuoda esille muuttujien välisiä yhteyksiä. Tulkittaessa muuttujien välisiä riippuvuuksia voidaan kuitenkin helposti tehdä vääriä johtopäätöksiä. Ristiintaulukointia tulkittaessa onkin erittäin tärkeää ymmärtää muuttujien tarkoitukset ja taustat. Muuttujia määriteltäessä tulee olla tarkkana, kumpi muuttujista on selittävä muuttuja. Väärin määritetty muuttuja voi aiheuttaa täysin päinvastaisen johtopäätöksen taulukosta kuin asia todellisuudessa on. (Taanila, A. 2007, 39.)

Korrelaatiokertoimen avulla tutkitaan onko tiettyjen muuttujien välillä kausaalista suhdetta (syy-seuraus-suhdetta). Korrelaatiokerroin voidaan laskea eri menetelmillä, Pearsonin korrelaatiokerrointa voidaan käyttää määrällisille muuttujille, järjestysasteikollisille muuttujille voidaan käyttää Spearmanin tai Kendallin järjestyskorrelaatiokertoimia.

Korrelaatiokerroin on $-1:n$ ja $1:n$ välillä jos korrelaatiokerroin on 0 , ei muuttujien välillä ole lineaarista riippuvuutta. Korrelaatiokertoimen etumerkki osoittaa riippuvuuden suunnan (pieneneko vai suureneeko toisen muuttujan arvo toisen suurentuessa). Korrelaatiokertoimen osoittaessa muuttujien välistä riippuvuutta (arvo lähes -1 tai 1) ei vielä voida olla varmoja, onko niiden välillä syy-seuraussuhde. Tästä johtuen korrelaatiokertoimen merkitsevyys tulee vielä testata. Testausta varten on olemassa taulukoita, joilla korrelaatiokerroin on helpompi testata kuin itse laskemalla. (Taanila, A. 2007, 39 - 43; Heikkilä, T. 2004, 203 - 204; Holopainen & Pulkkinen 2002, 198 - 210.)

4 LUOVA LOMAKE ANALYSOINTITYÖKALUN TOTEUTTAMINEN

Luovaliike Oy on vuonna 2004 perustettu kajaanilainen ohjelmistoalan yritys, joka toteuttaa tietokantapohjaisia ohjelmistoratkaisuja sekä web- että mobiili-ympäristöön. Luovaliike käyttää ratkaisuissaan Microsoft .NET -teknologiaa.

Luova Lomake -ohjelmistolla on muun muassa mahdollista suorittaa kyselytutkimuksia, hoitaa ilmoittautumisprosesseja sekä järjestää kilpailuja Internet-sivuilla. Ohjelmiston kehitystarpeina olivat kyselytutkimusten vastaustietojen käsittely ja analysointi. Luova Lomake -ohjelmistolla luotuja lomakkeita voidaan hallita ja vastaustietoja voidaan käsitellä selainpohjaisella työkalulla, johon kirjautuminen tapahtuu käyttäjätunnuksen ja salasanan avulla. (Luovaliike a. 2006)

Luova Lomake Analysointityökalu mahdollistaa vastaustietojen käsittelyn ja analysoinnin. Analysointityökalulla on mahdollista luoda raporttipohjia, joiden avulla saadaan kyselytutkimuksen vastaustiedoista generoitua reaaliaikainen raportti. Raportit auttavat loppukäyttäjää analysoimaan tekemiensä kyselytutkimusten tuloksia, ja esimerkiksi luomaan käyttäjäprofiileja sivustolla kävijöistä.

4.1 Ohjelmointikielien ja työkalut

Projekti toteutettiin Microsoftin ASP .NET 2.0 -alustalla. Ohjelmointiympäristönä toimi Microsoft Visual Studio .NET 2005, johon oli myös asennettu Visual Sourcesafe 2005. Ohjelmointikielenä käytettiin Visual Basic .NET:iä ja raporttipohjien tallentamiseen

käytettiin XML-dokumentteja, joiden rakenne määritettiin XML-Skeemassa. Raporteissa analysoitavat vastaustiedot voivat olla tallennettuina joko MySQL- tai Microsoft SQL Server 2005 -tietokantaan.

4.2 Vaatimusmäärittely

Luova Lomake Analysointityökalulle oli asetettu toimeksiantajan puolelta erilaisia vaatimuksia, joita käsitellään seuraavassa. Lyhyesti sanottuna tavoitteena oli toteuttaa työkalu, joka mahdollistaa nopean ja tehokkaan tavan käsitellä kyselytutkimuksen vastaustietoja.

Seuraavassa luettelossa on esitetty Luova Lomake Analysointityökalun vaatimukset:

- Voidaan muodostaa aiemmin Luova Lomake -ohjelmistolla luodun kyselyn perusteella raporttipohja, joka voi sisältää useita graafisia esityksiä / numeerisia taulukoita.
- Raportointisäännöt pohjautuvat kyselytutkimuksen kysymyksiin.
- Raportointisäännöt voivat sisältää:
 - o Suodatusehtoja, jotka mahdollistavat esimerkiksi vastaajien rajaamisen halutulla perusteella.
 - o Tulokset, eli mitä muuttujia näytetään raportissa (enintään kaksi muuttujaa per numeerinen taulukko / graafinen esitys).
 - o Voidaan valita kyselystä muuttujia, joiden mukaan tulostetaan numeerinen taulukko / graafinen esitys kaikista eri vaihtoehdoista.
- Raportin sisältämien graafisten esitysten / numeeristen taulukoiden järjestystä voidaan vaihtaa.

- Raporttiin voidaan kirjoittaa otsikko sekä aloitus- ja lopetusteksti.
- Raporttipohjia voi olla useita jokaisella kyselytutkimuksella.
- Kun raporttipohja on muodostettu, voidaan sen avulla generoida kyselytutkimuksesta reaaliaikainen raportti.
- Raportti on HTML-pohjainen, ja se voidaan tulostaa.
- Raportilla on yksilöllinen web-osoite, jolloin sitä voidaan jakaa sähköpostilla tai linkittää Internet-sivuille.

Luova Lomake Analysointityökalulle ei tarvitse erikseen suunnitella tietokantaa, koska se tehdään jo olemassa olevaan Luova Lomake -ohjelmistoon ja näin ollen tietokanta on jo valmiina (Liite 2.). Tietokantaa käytetään työssä ainoastaan tietojen lukemiseen ja sen sisältämien tietojen avulla lasketaan raporttipohjan raportointisääntöjen mukaiset tulokset kyselytutkimuksen vastausmateriaalista. Tietokannasta työssä käytetään vain seuraavia tauluja; lomake, kenttä, vaihtoehto ja vastaus.

Käyttöliittymiä ei tarvitse suunnitella, vaan tarkoituksena on noudattaa aiemmin tehdyn Luova Lomake -ohjelmiston käyttöliittymälogiikkaa ja visuaalista ulkoasua. Tietysti koko ajan ohjelmistoa tehtäessä on muistettava ajatella käyttöliittymän käytettävyyttä loppukäyttäjän kannalta, jottei sorruta liian monimutkaisiin ja vaikeakäyttöisiin ratkaisuihin.

XML-dokumenttien rakenteen määrittely on suurin suunnittelualaue koko työssä. Tämän tekee vaikeammaksi se, että koko XML-teknologia on suhteellisen uusi ja laaja ympäristö. XML-rakenteen suunnittelu täytyy tehdä kokeilemalla ja opettelemalla erilaisia toteutustapoja ja mahdollisuuksia, tätä kautta myös henkilökohtainen oppiminen on monipuolisempaa.

4.3 Työprosessi

Ohjelmistoprojekti koostuu eri osa-alueista. Kaikista selkein jaottelu voidaan suorittaa jakamalla se kolmeen eri osa-alueeseen, joita ovat suunnittelu, ohjelmointi ja testaus. Tällaisella jaottelulla osa-alueet sisälsivät Luova Lomake Analysointityökalun teossa seuraavia toimintoja.

- Suunnittelu; vaatimusmäärittely sekä ohjelmistorakenteen suunnittelu / tutkiminen.
- Ohjelmointi; ohjelmakoodin kirjoittaminen ja käyttöliittymien tekeminen.
- Testaus; ei erillistä testausta vaan jatkuvaa testausta ohjelmoinnin ohessa.

4.3.1 Suunnittelu

Luova Lomake Analysointityökalun suunnittelu alkoi tutustumisella Luova Lomake -ohjelmistoon, ja siihen kuinka se on toteutettu. Ohjelmistoon tutustumisen jälkeen alettiin tutkia XML-tekniologiaa ja sitä kuinka sen käyttäminen olisi järkevää tässä projektissa.

Luova Lomake -ohjelmiston rakenne ja toteutustapa oli suhteellisen nopeasti omaksuttavissa, mutta suurin haaste oli itse XML:ssä, joka havaittiin erittäin laajaksi ja monipuoliseksi erilaisten teknologioiden joukoksi. Lopulta päädyttiin käyttämään apuna XML-dokumenttien määrittelyssä XML-Skeemaa, jonka avulla XML-dokumentin rakenne saadaan validoitua, jolloin voidaan varmistaa dokumentin oikea muoto.

4.3.2 Ohjelmointi

Ohjelmointi- ja suunnitteluvaiheen raja oli vaikeasti erotettavissa Luova Lomake Analysointityökalu -projektissa, koska koko ajan ohjelmaa toteutettaessa tutkittiin myös XML-tekniologiaa ja sen parasta hyödyntämistapaa tässä projektissa.

Ohjelmointi tehtiin vaihe kerrallaan aloittamalla raporttipohjan luomissivusta, joka toteutettiin hieman wizard -tyylisenä, mutta kuitenkin siten että kaikki raporttipohjan tiedot ovat käyttäjän näkyvissä koko raporttipohjan luomisprosessin ajan. Toisena tehtiin

raporttipohjan muokkaussivu, joka noudattaa samaa logiikkaa kuin raporttipohjan luominen. Luova Lomake Analysointityökalun etusivulle tehtiin listaus luoduista raporttipohjista sekä mahdollisuus valita näytetäänkö listauksessa kaikki raporttipohjat vai pelkästään tiettyyn kyselytutkimukseen liittyvät raporttipohjat. Lisäksi etusivun listaukseen laitettiin mahdollisuus poistaa luotuja raporttipohjia.

Raportit tehtiin siten että ne generoidaan reaaliaikaisesti omalle Internet-sivulle. Kun käyttäjä menee raporttisivulle, haetaan tietokannasta raporttipohjan tarvitsemat tiedot ja analysoidaan ne samalla kun sivu ladataan.

4.3.3 Testaus

Varsinaista testaussuunnitelmaa ja -raporttia ei Luova Lomake Analysointityökalulle tehty, vaan testaus suoritettiin jatkuvana koko ohjelmistonkehityksen aikana. Tietokannan rasiustestaus olisi kuitenkin ollut tällaisessa projektissa tärkeää suorittaa ennen varsinaisen työn toteuttamista, koska tietokannan rakenne aiheutti suuria ongelmia generoitaessa raportteja Internet-sivulle. Ongelmat ilmenivät, kun raporttiin oli määritetty ristiintaulukoita, joissa on paljon muuttujia. Tällöin ohjelma meni ”jumiin” jopa pariiksi minuutiksi, joka on aivan liian pitkä aika käyttäjälle odotettavaksi.

Tietokannan rasiustestauksessa olisi saatu heti selville tietokannan rakenteelliset puutteet. Tietokanta olisi tällöin havaittu heti huonosti suunnitelluksi tällaiselle tietokantaa paljon rasittavalle työkalulle. Tietokannan pahimpia puutteita lienee se, että siinä oli käytetty paljon tekstimuotoisia tietotyyppisiä tiedontallentamiseen, kun olisi pitänyt käyttää erilaisia indeksejä. Indeksit olisivat olleet numeraalista tietoa, jolloin tietokantakyselyt olisivat olleet huomattavasti tehokkaampia eikä ohjelman ”jumiutumista” olisi niin todennäköisesti tapahtunut raporttia generoitaessa.

Luultavasti helpoin ratkaisu ohjelman ”jumiutumisen” korjaamiseksi tässä tilanteessa olisi ollut tehdä raporttien generointi ajastetusti. Tällöin jokainen raportti olisi päivitetty esimerkiksi kerran vuorokaudessa eikä joka kerta sivua ladattaessa. Opinnäytetyöhön käytettävän ajan rajallisuuden vuoksi tämä muutos jäi myöhemmin tehtäväksi.

4.4 Toteutunut ohjelmisto

Luova Lomake Analysointityökalu saatiin suhteellisen toimivaksi lukuun ottamatta ristiintaulukoinnissa ilmennyttä hitautta. Raporttipohjien käsittely ja raporttien generointi saatiin suhteellisen toimintavarmaksi ja helppokäyttöiseksi. Analysointityökalun käyttöliittymien selkeyttä olisi saatu vielä paremmaksi säätämällä lisää HTML-tyylejä, mutta ajanpuutteen vuoksi se jäi tässä opinnäytetyössä tekemättä.

4.4.1 Koodi

Luova Lomake Analysointityökalua ohjelmoitaessa noudatettiin aiemmin tehdyn Luova Lomake -ohjelmiston rakennetta ja koodaustapoja. Raporttipohjien XML-tiedostot tallennetaan yhteen kansioon. Jokainen XML-tiedosto sisältää yhden raporttipohjan, ja tiedosto on nimetty GUID-tunnisteella, joka toimii myös parametrina raportin osoitteessa, jolloin kukin raportti saadaan helposti yksilöityä.

Suurin osa työhön käytetystä ajasta meni raporttipohjan tietojen tallentamista varten luotavan XML-dokumentin rakenteen määrittelyssä (Liite 1.) ja XML-rakenteiden opettelussa.

Koko ohjelmisto on toteutettu yhdellä Visual Studio -projektilla. Projektirakenne Luova Lomake Analysointityökalun osalta on jaettu erillisiin kansioihin hallinnan ja raportin generoinnin osalta, jolloin käyttöoikeuksien perusteella on helppo antaa oikeuksia eri osaluille. Samalla jaolla on tehty koko Luova Lomake -ohjelmiston hallinta ja muiden osien erottelu.

Lisäksi ohjelmakoodit ovat omassa App_Code -kansiossa, joka on ASP .NET -projektien vakiokansio ohjelmakoodeille. Ohjelmiston ulkoasu on toteutettu CSS-tyylitiedostoissa olevilla määrittelyillä. Käytettävä tietokanta, Microsoft SQL Server 2005 tai MySQL asetetaan asiakaskohtaisessa konfiguraatitiedostossa. Tällöin käytettävä tietokanta voi olla eri asiakkailta erityyppinen. SQL-lauseet on koottu omaan luokaan, jossa on myös huomioitu SQL Server:in ja MySQL:n erot ja luokka osaa käyttää kulloinkin oikeaa SQL-lausetta.

Konfiguraatitiedostoa käytetään muutenkin sivustokohtaisten konfiguraatitietojen määrittelyyn, jolloin käytännössä muita ASP .NET -projektin osia ei eri sivustoja varten tarvitse muuttaa. Toki myös tyyli- eli CSS-tiedostoja muuttamalla saadaan toteutettua erilainen ulkoasu eri sivustoille.

Kirjautuneella käyttäjällä voi olla joko pää- tai peruskäyttäjäoikeudet. Käyttäjän perustiedot tallennetaan kirjautumisen yhteydessä sessio-muuttujiin, jolloin esimerkiksi tieto käyttöoikeuksista on helposti saatavilla koko istunnon ajan. Tällöin saadaan esimerkiksi hallintaan liittyvien sivujen latauksessa tarkistettua, onko kirjautuneella käyttäjällä pääkäyttäjäoikeuksia ja jos ei ole, voidaan käyttäjä ohjata pois kyseiseltä sivulta.

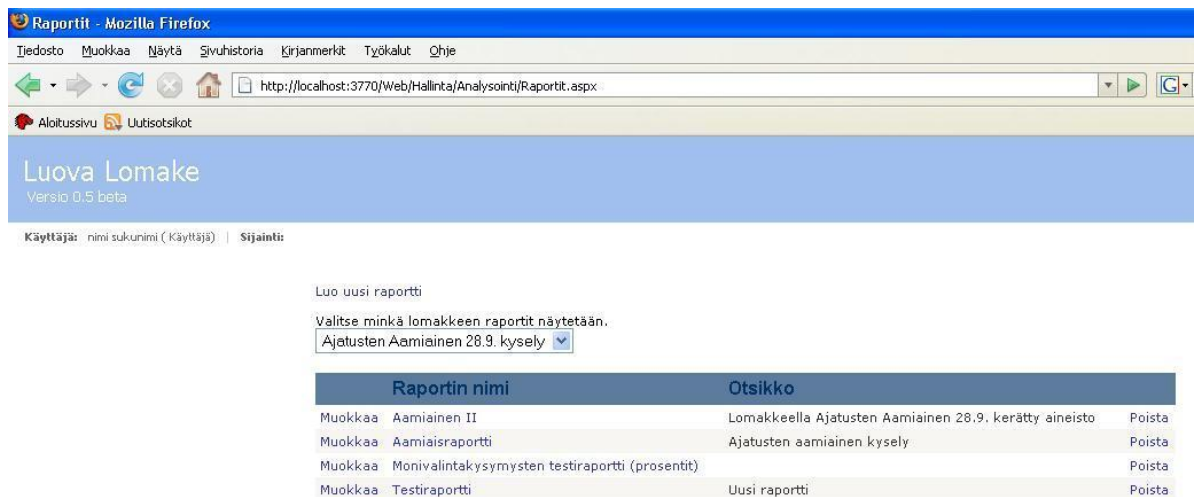
4.4.2 Käyttöliittymät

Käyttöliittymiä suunniteltaessa on pyrittävä tekemään niistä käytettävyydeltään mahdollisimman hyviä. Mikäli käytettävyys on huono, ei idealtaan hyvä ohjelmakaan saa käyttäjiä kiinnostumaan, jolloin koko ohjelma jää helposti käyttämättä.

Luova Lomake Analysointityökalussa on käyttöliittymistä ja toimintalogiikasta pyritty tekemään mahdollisimman yksinkertainen, jotta ohjelman käyttäminen ei tulisi liian monimutkaiseksi ohjelmiston käyttäjille. Seuraavilla sivuilla olevilla kuvankaappauksilla tuodaan esille ohjelmiston toimintaa ja käyttöliittymää.

Etusivu

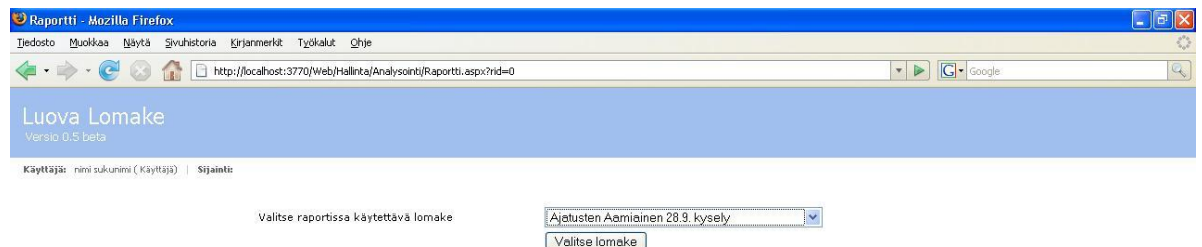
Luova Lomake Analysointityökalun etusivulle (Kuvio 9.) on koottu kaikki käyttäjän luomat raportit. Raporttilistaus tehdään käyttäjän valinnan mukaan joko kaikista tai vain tiettyyn kyselyyn liittyvistä raporteista. Raporttilistauksessa on linkit jokaisen raporttipohjan muokkaussivulle sekä raporttipohjan poistamiseen.



Kuvio 9. Luova Lomake Analysointityökalun etusivu (raporttien listaus).

Raporttipohjan luominen

Lisättäessä uutta raporttipohjaa Luova Lomake Analysointityökalulla täytyy käyttäjän ensin valita mitä kyselytutkimusta raporttipohjassa käytetään (Kuvio 10.). Kyselytutkimuksen valinnasta riippuu mitä tietoja tietokannasta haetaan käyttäjälle raporttipohjan luomisen eri vaiheisiin.



Kuvio 10. Raportissa käytettävän kyselytutkimuksen valinta.

Toisena vaiheena raporttipohjan luomisen yhteydessä käyttäjä antaa raportin perustiedot (Kuvio 11.), joita ovat raportin nimi, otsikko, alku- ja lopputeksti. Näitä tietoja käytetään raportissa selkeyttämään raportin sisältöä ja tarkoitusta.

Raportti - Mozilla Firefox
Tiedosto Muokkaa Näytä Sivuhistoria Kirjanmerkit Työkalut Ohje
http://localhost:3770/Web/Hallinta/Analysointi/Raportti.aspx?nid=0
Google

Luova Lomake
Versio 0.5 beta

Käyttäjä: nimi sukunimi (Käyttäjä) | Sijainti

Valitse raportissa käytettävä lomake: Ajustusten Aamiainen 28.9. kysely

Peruuta raportin luominen

Raportin perustiedot

Raportin nimi:

Raportin otsikko:

Raportin alkuteksti:

Raportin lopputeksti:

Tallenna

Kuvio 11. Raportin perustiedot.

Raporttiin voidaan lisätä elementtejä rajoittamaton määrä. Elementin tyyppi voi olla joko muuttujan kuvailu (Kuvio 12.) tai ristiintaulukointi (Kuvio 13.). Yhteisiä tietoja molemmille elementtityypeille ovat nimi ja otsikko. Muuttujan kuvailu -elementille valitaan yhteisten tietojen lisäksi seliteteksti ja muuttuja jota käsitellään. Ristiintaulukointi -elementille puolestaan valitaan esitystapa (kappalemäärä / prosentit), Y- ja X-akselin muuttujat, taulukon seliteteksti sekä mahdolliset suodatus ehdot. Suodatus ehdoiksi voidaan antaa esimerkiksi sukupuoli, jolloin vain valitun sukupuolen vastaajien vastaukset huomioidaan.

Raportti - Mozilla Firefox
Tiedosto Muokkaa Näytä Sivuhistoria Kirjanmerkit Työkalut Ohje
http://localhost:3770/Web/Hallinta/Analysointi/Raportti.aspx?nid=0
Google

Luova Lomake
Versio 0.5 beta

Käyttäjä: nimi sukunimi (Käyttäjä) | Sijainti

Valitse raportissa käytettävä lomake: Ajustusten Aamiainen 28.9. kysely

Pilote perustiedot Peruuta raportin luominen

Raportin perustiedot

Raportin nimi: Uusi raportti

Raportin otsikko:

Raportin alkuteksti:

Raportin lopputeksti:

Tallenna Muokkaa perustietoja

Elementin lisääminen

Elementin nimi:

Elementin otsikko:

Valitse elementin tyyli: Kuvaileva taso Taulukointi

Anna seliteteksti: Merkitse ## -merkeillä kohta johon haluat tuloksen.

Valitse muuttuja: Valitse-

Tallenna elementti Peruuta

Miten tieto näytetään?: Kappalemäärä Prosentit

Kuvio 12. Muuttujan kuvailu elementin lisääminen raporttiin.

Raportti - Mozilla Firefox

Tiedosto Muokkaa Näytä Sivuhistoria Kirjanmerkit Työkälu Ohje

http://localhost:3770/Web/Hallinta/Analysointi/Raportti.aspx?nid=0

Luova Lomake
Versio 0.5 beta

Käyttäjäs: nimi sukunimi (Käyttäjäs) | Sijainti:

Valitse raportissa käytettävä lomake: Ajustusten Aamiainen 28.9. kysely

Pilota perustiedot Peruuta raportin luominen

Raportin perustiedot

Raportin nimi: Uusi raportti

Raportin otsikko:

Raportin alkuteksti:

Raportin lopputeksti:

Tallenna Muokkaa perustietoja

Elementin lisääminen

Elementin nimi:

Elementin otsikko:

Valitse elementin tyyli: Kuvallinen taso Taulukointi

Valitse taulukoinnin tyyli: Numeerinen taulukko Graafinen esitys

Esitystapa: Kappalemäärä Prosentit

Valitse muuttuja (x-akseli): -Valitse-

Valitse muuttuja (y-akseli): -Valitse-

Tallenna elementti Peruuta

Suodatukset

Lisää suodatusehto

Kuvio 13. Ristiintaulukointi elementin lisääminen raporttiin.

Raportti - Mozilla Firefox

Tiedosto Muokkaa Näytä Sivuhistoria Kirjanmerkit Työkälu Ohje

http://localhost:3770/Web/Hallinta/Analysointi/Raportti.aspx?nid=0

Luova Lomake
Versio 0.5 beta

Käyttäjäs: nimi sukunimi (Käyttäjäs) | Sijainti:

Valitse raportissa käytettävä lomake: Ajustusten Aamiainen 28.9. kysely

Pilota perustiedot Peruuta raportin luominen

Raportin perustiedot

Raportin nimi: Uusi raportti

Raportin otsikko:

Raportin alkuteksti:

Raportin lopputeksti:

Tallenna Muokkaa perustietoja

Elementit

		Nimi	Otsikko	
↑	↓	Näytä elementti	Elementti 2	Poista
↑	↓	Näytä elementti	Elementti 1	Poista
↑	↓	Näytä elementti	Elementti 3	Poista

Lisää elementti

Tallenna raportti

Kuvio 14. Tietojen listaus ennen raportin tallentamista.

Lisätyt elementit listataan raporttipohjan loppuun (Kuvio 14.). Listauksesta näkyvät elementin nimi ja otsikko. Listauksessa elementtien järjestystä voidaan muuttaa ja elementtejä päästään muokkaamaan tai poistamaan.

Raportti

Raporttipohjan tallentamisen jälkeen sen avulla saadaan generoitua reaaliaikainen raportti kyselytutkimuksen tiedoista (Kuvio 15.). Jokaisella raportilla on yksilöllinen tunniste, jonka avulla tunnistetaan, mikä raportti halutaan generoida.

http://localhost:3770/Web/Analysointi/Raportti.aspx?rid=509968b0-02f3-4a70-a945-c8f3f614bac6

Aarniainen II 21.4.2007

Lomakkeella Ajatusten Aarniainen 28.9. kerätty aineisto

Kyselyyn vastanneita 5605 kpl

Raporttiin on kerätty tieto kyselyn Ajatusten Aarniainen 28.9. pohjalta.

Asema
Asema ja työpaikka
Suodatusehdot

Sukupuoli Mies

	Johtoryhmän jäsen / johtaja	Esimies / päällikkö	Tiiminvetäjä / työnjohtaja	Asiantuntija	Työntekijä	Yhteensä
Etelä-Suomen lääni	50,46 % (0,98 %)	41,38 % (0,64 %)	45,00 % (0,32 %)	51,61 % (1,43 %)	39,39 % (0,70 %)	4,07 %
Länsi-Suomen lääni	23,85 % (0,46 %)	27,59 % (0,43 %)	37,50 % (0,27 %)	27,74 % (0,77 %)	33,33 % (0,59 %)	2,52 %
Itä-Suomen lääni	11,01 % (0,21 %)	11,49 % (0,18 %)	12,50 % (0,09 %)	8,39 % (0,23 %)	13,13 % (0,23 %)	0,94 %
Oulun lääni	8,26 % (0,16 %)	10,34 % (0,16 %)	0,00 % (0,00 %)	8,39 % (0,23 %)	10,10 % (0,18 %)	0,73 %
Lapin lääni	4,59 % (0,09 %)	5,75 % (0,09 %)	0,00 % (0,00 %)	3,87 % (0,11 %)	2,02 % (0,04 %)	0,33 %
Ahvenanmaan maakunta	1,83 % (0,04 %)	3,45 % (0,05 %)	5,00 % (0,04 %)	0,00 % (0,00 %)	2,02 % (0,04 %)	0,17 %
Yhteensä	100 % (1,94 %)	100 % (1,55 %)	100 % (0,72 %)	100 % (2,77 %)	99,99 % (1,78 %)	
Kysymykseen vastanneet ja suodatusehdot täyttävät: 490 kpl	X: Oma asema Y: Työpaikkari sijainti Suhissa olevat tulokset on laskettu vertaamalla ristintaulukointiehtoja kaikkien kyselyyn vastanneiden vastauksiin.					

Miesten työpaikan sijainnin ja aseman suhde.

Kiitos mielenkiinnosta raporttia kohtaan!

[Tulosta raportti](#) [Sulje raportti](#)

Kuvio 15. Valmis raportti.

Ristiintaulukointi

Raportin sisältämät ristiintaulukoinnit voivat olla esitetty kappalemäärin (Kuvio 16.) tai prosenttein (Kuvio 17.). Ristiintaulukoinnissa ylärivillä olevat tekstit tulevat elementin X-akselille valitun muuttujan vaihtoehdoista, ja vasemmalla olevat tekstit elementin Y-akselille valitun muuttujan vaihtoehdoista. Ristiintaulukoinneissa on yhteensä tieto (rivi- tai sarakesumma) sekä X- että Y-akselin muuttujalle, johon on laskettu summa kyseisen rivin tai sarakkeen arvosta. Ristiintaulukoinnin alalaidassa on tieto kuinka monta vastaajaa kyseisen ristiintaulukoinnin tulokseen on vaikuttanut sekä tiedot mikä muuttuja on X- ja Y-akselilla. Lisäksi prosenttein esitettävässä ristiintaulukoinnissa on jokaisessa ristiintaulukointisolussa sulkuihin laitettu arvo, josta käy ilmi ristiintaulukoinnin tulos ilman suodatusehtoja.

	Alle 25	25-30	31-35	36-45	46-55	55 tai yli	Yhteensä
Johtoryhmän jäsen / johtaja	20 kpl	15 kpl	21 kpl	92 kpl	100 kpl	97 kpl	345 kpl
Esimies / päällikkö	15 kpl	28 kpl	38 kpl	158 kpl	142 kpl	124 kpl	505 kpl
Tiiminvetäjä / työnjohtaja	7 kpl	32 kpl	30 kpl	73 kpl	67 kpl	52 kpl	261 kpl
Asiantuntija	46 kpl	102 kpl	120 kpl	319 kpl	231 kpl	209 kpl	1027 kpl
Työntekijä	302 kpl	245 kpl	234 kpl	827 kpl	743 kpl	565 kpl	2916 kpl
Yhteensä	390 kpl	422 kpl	443 kpl	1469 kpl	1283 kpl	1047 kpl	
	Sukupuoli	Nainen					
Kysymykseen vastanneet ja suodatusehdot täyttävät: 5054 kpl	X: Ikä Y: Oma asema						

Kuvio 16. Ristiintaulukointi vastausten lukumäärillä.

	Nainen	Mies	Yhteensä
Johtoryhmän jäsen / johtaja	6,83 % (6,16 %)	22,24 % (1,94 %)	8,1 %
Esimies / päällikkö	9,99 % (9,01 %)	17,76 % (1,55 %)	10,56 %
Tiiminvetäjä / työnjohtaja	5,16 % (4,66 %)	8,16 % (0,71 %)	5,37 %
Asiantuntija	20,32 % (18,32 %)	31,63 % (2,77 %)	21,09 %
Työntekijä	57,70 % (52,02 %)	20,20 % (1,77 %)	53,79 %
Yhteensä	100 % (90,17 %)	99,99 % (8,74 %)	
Kysymykseen vastanneet ja suodatusehdot täyttävät: 5544 kpl	X: Sukupuoli Y: Oma asema Suluissa olevat tulokset on laskettu vertaamalla ristiintaulukointiehtoja kaikkien kyselyyn vastanneiden vastauksiin.		

Kuvio 17. Ristiintaulukointi prosenteilla.

5 POHDINTA

Opinnäytetyön tavoitteena oli toteuttaa tehokas ja selkeä työkalu kyselytutkimusten tietojen analysointiin osaksi Luova Lomake -ohjelmistoa. Luova Lomake -ohjelmistolla voidaan muun muassa toteuttaa kyselytutkimuksia Internet:ssä, jolloin on myös tarpeellista pystyä analysoimaan kerättyä materiaalia. Työn teoriaosuus käsittelee XML:ää ja tilastollista tutkimusta. Kummatkin näistä osa-alueista olivat tekijälle uusia asioita, joka aiheutti sen, että kaikki tieto joka tuli teoriaosuuteen oli myös uuden oppimista.

Luova Lomake Analysointityökalu oli melko vapaasti toteutettavissa annettujen vaatimusten mukaisesti. Työ toteutettiin Luovaliike Oy:n tiloissa ja laitteilla. Microsoft Visual Studio 2005 ja Microsoft SQL Server 2005 olivat molemmat jo tuttuja työkaluja, mutta käytettävä ohjelmointikieli Visual Basic .NET sekä toinen tietokantaohjelma MySQL olivat tekijälle suhteellisen tuntemattomia. Kokemusta Visual Basic .NET:stä oli vain yhden kurssin verran ja MySQL:ää ei tekijä ollut aiemmin käyttänyt. Näin ollen työtä tehtäessä Internet oli hyvä väline tiedonhankinnassa.

Käytännön osuutta tehtäessä tutustuttiin aluksi Luova Lomake -ohjelmistoon ja sen rakenteeseen, jonka jälkeen alettiin suunnitella kuinka opinnäytetyönä oleva Luova Lomake Analysointityökalu kytkeytyisi ohjelmiston muihin osioihin. Tässä ei mitään vaikeuksia ollut, koska opinnäytetyö ei suoranaisesti liittynyt aiemmin toteutettuihin osioihin, vaan käytännössä kaikki käytettävä data on tietokannassa. Raporttipohjan luomista varten tietokannasta haettiin valittuun kyselytutkimukseen liittyvät tiedot, joiden pohjalta käyttäjä luo haluamansa muotoisen raporttipohjan. Raporttipohja tallennetaan XML-tiedostoon. Raporttia generoitaessa Luova Lomake Analysointityökalu lukee XML-tiedoston ja sen tekee pohjalta tarvittavat kyselyt tietokantaan ja luo reaaliaikaisen raportin.

Teoriaosuuteen tuli kaksi suurempaa osa-aluetta, XML ja tilastollinen tutkimus. Kumpikin aihealue oli tekijälle uusia, joten kaikki hankittu tieto oli uuden opiskelua. XML:ssä yllätti alueen laajuus ja ongelmaksi muodostuikin se, mitkä osiot ovat keskeisimpiä kokonaisuuden kannalta ja mitkä olennaisesti liittyvät opinnäytetyön käytännönsioon. Tilastollinen tutkimus otettiin teoriaosuuteen mukaan, jotta se helpottaisi työn kokonaisuuden ymmärtämistä. Sen osuus on kuitenkin hyvin rajattu koskemaan ainoastaan sellaisia asioita, jotka olennaisesti liittyvät käytännön työhön.

XML:n teorian opiskelu ja sen soveltaminen käytännön työssä olivat erittäin opettavaisia ja hyödyllisiä tulevaisuutta varten. XML on erittäin hyödyllinen ja vartenotettava menetelmä erilaisissa ohjelmointiprojekteissa, mutta kannattaa myös muistaa että se ei välttämättä ole paras ratkaisu kaikkiin kohteisiin missä sitä voisi käyttää. Lähiaikoina julkaistavan uuden Microsoft Visual Studio:n mukana tuleva XLINQ muuttanee XML:n käsittelyä .NET ympäristössä huomattavasti. Tulevaisuus näyttää kuinka XML jatkaa kehittymistään ja sitä mukaa yleistymistään erilaisissa sovellusympäristöissä.

Kokonaisuutena opinnäytetyö onnistui hyvin. Ainoa takaisku oli tietokantarakenteesta johtuva raporttien generoinnin hitaus. Senkin ongelman saisi ratkaistua helposti, kun muuttaisi analysointityökalua siten, ettei raportteja generoitaisi reaaliaikaisesti vaan esimerkiksi kerran vuorokaudessa generoitaisiin raportti sen hetkisestä tilanteesta. Käyttäjän pyytäessä raporttia ohjelmalta, haetaan viimeksi generoitu versio raportista, jolloin raportin lataamisaika saataisiin minimoitua.

LÄHTEET

Extensible Markup Language (XML) 1.0 (Fourth Edition). W3C Recommendation. 2006. Web-dokumentti. Saatavilla: <http://www.w3.org/TR/2006/REC-xml-20060816>. (Luettu 16.9.2007).

Extensible Markup Language (XML) 1.1 (Second Edition). W3C Recommendation. 2006. Web-dokumentti. Saatavilla: <http://www.w3.org/TR/2006/REC-xml11-20060816>. (Luettu 18.9.2007).

Heikkilä, T. 2004. Tilastollinen tutkimus. Helsinki: Edita.

Holopainen, M. & Pulkkinen, P. 2002. Tilastolliset menetelmät. Helsinki: WSOY.

Holzner, S. 2001. Inside XML. Jyväskylä: Edita.

Inkinen, V. 2003. ASP.NET. Jyväskylä: Docendo.

Lipitsäinen, A. 2001. XML – Extensible Markup Language. Helsingin liiketalouden ammattikorkeakoulu. <http://my.helia.fi/~atk42d/Xml/xmlopas.html> (Luettu 17.5.2006).

Luovaliike a. <http://www.luovaliike.fi/luovalomake.aspx> (Luettu 20.11.2006).

Megginson Technologies. <http://www.megginson.com/downloads/SAX> (Luettu 3.11.2007)

Microsoft Project LINQ, 2006. X.Linq .NET Language Integrated Query for XML Data.

North, S. & Hermans, P. 2000. XML Trainer Pro. Helsinki: Edita.

Nykänen, O. 2001. XML, Jyväskylä: Docendo.

Nykänen, O. 2003 XML-peruskurssi koulutusmateriaali, W3C Suomen toimisto.
<http://www.w3c.tut.fi/training/2003/0219-xml-1/johdanto-xmlpros.html> (Luettu 22.9.2007)

SAX project. <http://www.saxproject.org/sax1-roadmap.html> (Luettu 3.11.2007)

Taanila, A. 2007. Aineiston esittäminen ja kuvailu. HAAGA-HELIA ammattikorkeakoulu.

Taanila, A. 2006. Kvantitatiivisen tutkimuksen suunnitteluvaihe. HAAGA-HELIA ammattikorkeakoulu.

Tuikka, T. & Kanala, S. 2001. XML Ohjelmoinnin perusteet. Helsinki: Edita.

Walkama, P. & Laakkonen, A. 2004. Inside XML-Skeema. Helsinki: IT Press.

W3C a. <http://www.w3.org/XML> (Luettu 15.9.2007).

W3C b. <http://www.w3.org/XML/Core> (Luettu 15.9.2007).

W3C c. <http://www.w3.org/DOM/DOMTR> (Luettu 3.11.2007).

W3C Communications Team, 2003. XML in 10 points.

<http://www.w3.org/XML/1999/XML-in-10-points.html.en> (Luettu 28.1.2007).

LIITTEET

LIITE 1: XML-dokumentti

LIITE 2: Luova Lomake tietokannan taulut

```

- <Raportti xmlns="http://luovaliike.fi/lomake/Raportti.xsd" LID="109">
  <RaportinNimi>Testiraportti</RaportinNimi>
  <RaportinOtsikko>Uusi raportti</RaportinOtsikko>
  <Alkuteksti>Tähän tulee raportin alussa oleva teksti.</Alkuteksti>
  <Lopputeksti>Tähän tulee raportin lopussa oleva teksti.</Lopputeksti>
  <Unique_URL>16606309-dba2-413b-bf37-1e376d2f792a</Unique_URL>
- <Elementti Lajittelu="1">
  <ElementinNimi>Elementti 1</ElementinNimi>
  <ElementinOtsikko>Kuvailu elementti</ElementinOtsikko>
  <ElementinTyyli>0</ElementinTyyli>
  - <Kuvailu>
    <Kentta>693</Kentta>
    <Vaihtoehto>17</Vaihtoehto>
    <KuvailunTyyli>1</KuvailunTyyli>
    <Teksti>Kyselyyn vastanneista ## on miehiä.</Teksti>
  </Kuvailu>
</Elementti>
- <Elementti Lajittelu="2">
  <ElementinNimi>Elementti 2</ElementinNimi>
  <ElementinOtsikko>Taulukointi elementti</ElementinOtsikko>
  <ElementinTyyli>1</ElementinTyyli>
  - <Taulu>
    <Teksti>Tähän tulee taulukon seliteteksti.</Teksti>
    <TaulunTyyli>0</TaulunTyyli>
    <TaulunEsitystapa>1</TaulunEsitystapa>
    - <Muuttujat>
      <MuuttujaX>693</MuuttujaX>
      <MuuttujaY>691</MuuttujaY>
    </Muuttujat>
    - <Suodatukset>
      <Kentta>-1</Kentta>
      <Vaihtoehto>-1</Vaihtoehto>
    </Suodatukset>
  </Taulu>
</Elementti>
</Raportti>

```

kayttaja	
PK	<u>UID</u>
	LisaysPvm Etunimi Sukunimi Email Tunnus Salasana Tyyppi LisaysOikeus IpRajaus

kentta	
PK	<u>KID</u>
	LID Lajittelu Tyyppi Pakollinen

lomake	
PK	<u>LID</u>
	UID LID_URL LisaysPvm MuokkausPvm spvm epvm Salasana Otsikko Alkutieto Lopputieto MaaraMax TallennusTeksti VahvistusTapa VahvistusViestiOtsikko VahvistusViestiTeksti OrganisaatioVasEmail OrganisaatioIImEmail Tid EID VahvistusEmail

vastaaja	
PK	<u>VID</u>
	LID ip spvm epvm Kasittelija KasittelyTila

kyselyntulostuskentat	
PK	<u>Kid</u>
	TulRyhma TulKnt_kysTxltd TulKnt_kys_txt

lomakekayttooikeus	
PK	<u>UID</u>
PK	<u>LID</u>
	OikeusKatselu OikeusMuokkaus OikeusKasittely

tyyli	
PK	<u>Tid</u>
	Tyylitiedosto Oletus

vastaus	
PK	<u>VID</u>
PK	<u>KID</u>
	Vastaus

vaihtoehto	
PK	<u>VAID</u>
	KID Lajittelu Teksti