

Ville Miettinen

# Sisällönhallintajärjestelmä pienyrityksen asiakaskontaktien hallinnan välineenä

Metropolia Ammattikorkeakoulu  
Insinööri (AMK)  
Tietotekniikka  
Insinöörityö  
9.5.2012

Tekijä(t) Otsikko Sivumäärä Aika	Ville Miettinen Sisällönhallintajärjestelmä pienyrityksen asiakaskontaktien hallinnan välineenä 38 sivua 9.5.2012
Tutkinto	Insinööri (AMK)
Koulutusohjelma	Tietotekniikka
Suuntautumisvaihtoehto	Ohjelmistotekniikka
Ohjaaja(t)	Lehtori Vesa Ollikainen Toimitusjohtaja Pirjo Silius
<p>Tämä insinöörityö tehtiin PTSILIUS-asiantuntijayritykselle. Työssä selvitettiin vapaan ohjelmakoodin sisällönhallintajärjestelmän sopivuutta yritykselle asiakashallinnan apuvälineenä. Selvityksen lisäksi toteutettiin valitulla sisällönhallintajärjestelmällä yritykselle kotisivut, jotka helpottavat asiakaskontaktien hallintaa, sekä testattiin ja vertailtiin valittuja sisällönhallintajärjestelmiä.</p> <p>Työn alkupuolella käydään läpi syyt vapaan ohjelmakoodin ohjelmistojen käyttöön ja sitä, millaisessa testiympäristössä kehitystä tehtiin. Lisäksi esitellään sisällönhallintajärjestelmien ymmärtämisen kannalta tärkeimmät ohjelmointikielet.</p> <p>Työn tavoitteena oli uusia PTSILIUS-yrityksen kotisivut niin, että sivustoa voidaan päivittää ilman suurta ohjelmointiosaamista. Tähän soveltuivat parhaiten sisällönhallintajärjestelmät, joista tutkittiin kuutta erilaista. Tarkemmin paneuduttiin Drupal- ja Joomla!-sisällönhallintajärjestelmiin, joista yrityksen käyttöön parhaiten soveltuvaksi paljastui Joomla!.</p> <p>Joomla! soveltuu yrityskäyttöön mainiosti, ja siihen on sen levinneisyyden vuoksi tarjolla hyvin paljon kaikenlaisia lisäosia. Yrityksen näkökulmasta hyödyllisiä lisäosia ovat esimerkiksi sopivat kielipaketit, joilla sivustoista saadaan helpommin muokattavia. Ulkoasun muokkaamiseen soveltuvat sivupohjat sekä sisällön tuottamista helpottavia tekstin muokkaajia.</p>	
Avainsanat	Sisällönhallinta, avoin lähdekoodi.

Author(s) Title Number of Pages Date	Ville Miettinen Content management system as a tool for customer relationship management in a small enterprise 38 pages 9 May 2012
Degree	Bachelor of Engineering
Degree Programme	Information Technology
Specialisation option	Software Engineering
Instructor(s)	Vesa Ollikainen, Senior Lecturer Pirjo Silius, Company Manager
<p>This engineering study was carried out for an independent expert company called PTSILIOUS. The aim of the study was to examine open source content management systems as help for the managing company's clients. The company's homepages were then done using the chosen content management system, so that they would help managing its clients, the chosen content management system was also tested and analysed.</p> <p>The study begins by explaining why open source programs were used and in what kind of testing environment the development was carried out. Also the programming languages that are needed to understand content management systems are presented.</p> <p>The aim of the study was to update PTSILIOUS company's homepages so that updating them would not require extensive programming experience. For this the content management systems were the best choice, and therefore six different ones were studied. To a closer look up Drupal and Joomla! systems were chosen, from these two Joomla! was found out to be the better choice for the company in question.</p> <p>Joomla! seems quite suitable for the company's use and there are many add-ons for it because of its wide distribution around the world. From company's viewpoint useful add-ons are for example suitable language packages that make the editing easier, templates that are used for modify the looks of the sites and text editors that make the editing easier.</p>	
Keywords	Content management, open-source.

## Sisälllys

### Lyhenteet

1	Johdanto	1
2	Tavoitteet ja toimintaympäristö	2
3	Sisällönhallintajärjestelmiä tukevat teknologiat	4
3.1	PHP	4
3.2	MySQL	5
3.3	HTML ja XHTML	6
3.4	CSS	10
4	Soveltuvia sisällönhallintajärjestelmiä	11
4.1	Vapaan lähdekoodin sisällönhallintajärjestelmät	16
4.2	Drupal	22
4.2.1	Historia	22
4.2.2	Ominaisuudet	23
4.2.3	Sivuston luominen Drupalin avulla	24
4.3	Joomla!	25
4.3.1	Historia	25
4.3.2	Ominaisuudet	26
4.3.3	Sivuston luominen Joomla!n avulla	27
4.4	Sisällönhallintajärjestelmän valinta	28
5	Sisällönhallintajärjestelmän käyttöönotto	28
5.1	Joomla!n asennus	29
5.2	Sivuston luominen	30
6	Toteutettu palvelu	34
6.1	Sivuston rakenne	35
6.2	Asiakaskontaktien hallintatoiminnallisuus	35
7	Yhteenveto	36
	Lähteet	38

## Lyhenteet

PHP	HyperText Preprocessor skriptikieli, joka on tarkoitettu palvelimen ohjelmointiin.
MVC	Model View Controller, kolmiosainen ohjelmointimalli.
SQL	Structured Query Language, ohjelmointikieli tietokantojen käsittelyyn.
HTML	HyperText Markup Language, ohjelmointikieli, jolla tuotetaan eri ympäristöihin siirtokelpoisia hypertekstiasiakirjoja.
XML	eXtensible Markup Language, ohjelmointikieli, jolla voidaan kuvata tiedon merkitys tiedon sekaan.
XHTML	HTML-koodia, johon sovelletaan XML-kielen mukaisen XHTML-kieliopin sääntöjä.
CSS	Cascading Style Sheets, merkintäjärjestelmä, jolla voidaan antaa ohjeita web-sivujen ja muidenkin dokumenttien ulkoasuista.
CMS	Content Management System, internetsivujen luomista ja ylläpitoa helpottava järjestelmä.
CGI	Common Gateway Interface, web-ympäristön tekniikka, jonka avulla selain voi välittää tietoa palvelimella suoritettavalle ohjelmalle.

## 1 Johdanto

Yritysten kotisivut ovat nykyisin erittäin tärkeitä. Niiden perusteella hyvin monesti tehdään päätös siitä, ostetaanko palvelu kyseiseltä yritykseltä. Tämän vuoksi niiden suunnitteluun ja toteutukseen kannattaa nykyisin panostaa.

Tämä insinööri työ syntyi tavoitteesta parantaa PTSILIUS-asiantuntijayrityksen web-sivuja asiantuntijayritysten web-sivustoilla hyvin yleisesti käytössä olevilla palveluilla: yhteydenottolomake ja useimmin kysytyjen kysymysten listaus vastauksineen. Yhteydenpitomakkeella vältetään henkilötietolain alaisten tietojen ilmoittaminen verkkosivustolla ja myös roskapostilähetykset. Tämä tarkoittaa käytännössä sitä, ettei yritys talleta sivustolleen tai tietokantoihinsa asiakkaiden tietoja, vaan ne löytyvät mahdollisesti vain yrityksen sähköpostista. Ratkaisuun päädyttiin osittain työn tekemisen aikana paljon mediassa olleiden tietomurtojen vuoksi. Useimmin kysytyjen kysymysten vastauksilla asiantuntijayrityksen tarkoituksena on lisätä asiantuntijatehtävien myyntiä ja markkinointia. Useimmin kysytyjen kysymysten vastaukset tarttuvat hyvin verkon hakumootoreihin. Näin ollen tietoisuus asiantuntijayrityksen palveista lisääntyy verkon kautta. Vastauksia yrittäjä pystyy generoimaan myös artikkeleiden kautta, joka on erittäin hyvä toiminto ohjaamaan vastauksia yrityksen tarjoaman asiantuntija-alueen piiriin. Lisäksi sivut haluttiin englannin- ja suomenkielisinä, ja ratkaisun tuli olla toteutettavissa vapaan lähdekoodin ohjelmilla.

Luvussa 2 kuvataan työn tavoitteet ja esitellään, minkälaisessa testiympäristössä sivustoa kehitettiin ja kerrotaan tiedot asiakkaana olleesta yrityksestä. Luvussa paneudutaan myös siihen, miksi asiakkaana oleva yritys alusta asti halusi sivuston pohjaavan täysin avoimen lähdekoodin ohjelmistoille. Lisäksi kerrotaan, miksi sisällönhallintajärjestelmät otettiin projektiin mukaan.

Luvussa 3 esitellään teknologiat, joiden avulla sisällönhallintajärjestelmät on luotu sekä jotka tukevat näitä järjestelmiä. Sisällönhallintajärjestelmät on yleensä luotu käyttämällä PHP- ja XHTML-kieliä. Luvussa käydään nämä läpi yleisellä tasolla sekä tutustutaan uuteen versioon HTML-kielestä, joka on nimeltään HTML5. Sisällönhallintajärjestelmiä tukevat teknologiat ovat SQL-tietokannat, joista ylivoimaisesti käytetyin on MySQL sekä XHTML-kielen muotoiluun käytettävä CSS-kieli.

Luvussa 4 esitellään sisällönhallintajärjestelmien periaate sekä se, mihin ne perustuvat. Luvussa tutustutaan lähemmin kuuteen eri järjestelmään, jotka valittiin vapaan lähdekoodin sivustolta <http://php.opensourcecms.com/>. Näistä kuudesta valittiin lisäksi läpikäytäväksi kaksi, joiden hallintajärjestelmän historiaan ja käyttösovelluksiin paneudutaan tarkemmin. Luvun lopuksi kerrotaan, miksi päädyttiin toiseen tarkassa tarkastelussa olleeseen hallintajärjestelmään.

Luvussa 5 käydään läpi, kuinka Joomla tulee asentaa palvelimelle kuvaamalla, kuinka sisällönhallintajärjestelmälle tulee luoda yhteys tietokantaan ja kuinka varmistaa asennuksen toimivuus. Lisäksi luvussa käydään läpi asiakkaan sivuston luomisen eri vaiheet ydinversion asennuksen jälkeen. Tämä sisältää muun muassa selostuksen, kuinka luoda sivustosta kaksikielinen ilman koodaamista.

Luvussa 6 esitellään lopputuotteena syntynyt sivusto ja selostetaan sivujen luontiperiaatteita. Luvussa kerrotaan myös, kuinka sivuston jatkokehitystä on tarkoitus viedä eteenpäin, jotta sivustosta saataisiin mahdollisimman yksilöllinen.

Luvussa 7 esitetään yhteenveto työstä sekä tarkastellaan työn tavoitteiden toteutumista.

## **2 Tavoitteet ja toimintaympäristö**

Työllä on kaksi tavoitetta:

1. Luoda helposti hallinnoitavissa oleva sivusto, joka on toteutettu kahdella eri kielellä. Sivustolla tulee olla usein kysytyt kysymykset -sivu, josta mahdolliset asiakkaat saavat tietoa siitä, millaiset tietotaidot yritykseltä löytyvät. Sivustolla tulee myös olla yhteydenottolomake, mikä mahdollistaa asiakkaan yhteydenoton ilman erikseen mainittavaa osoitetta.
2. Vertailla verkosta vapaan lähdekoodin ohjelmistojen alta löytyviä sisällönhallintajärjestelmiä, jotta saadaan selville, mikä näistä järjestelmistä soveltuu yrityksen käyttöön parhaiten.

Web-sivuja päädyttiin rakentamaan sisällönhallintajärjestelmän avulla niiden mahdollistaman helpon hallinnoinnin vuoksi. Koska sisällönhallintajärjestelmät eivät olleet ennestään tuttuja, päädyttiin niitä tutkimaan tarkemmin. Jo alkuvaiheessa selvisi, että verkko on täynnä erilaisia ilmaisia sisällönhallintajärjestelmiä, joista tulisi valinta tehdä. Valintaa varten käytiin läpi useita sisällönhallintajärjestelmiä, jonka jälkeen päädyttiin tutkimaan lähemmin kahta suurta suosiota nauttivaa sisällönhallintajärjestelmää.

PTSILIUS on vuonna 2004 perustettu konsulttiyritys. PTSILIUS tarjoaa hanke- ja projektiosaamista, julkishallinnon kilpailuttamista tarjouspyynnöistä sopimuksiin ja erilaisia asiantuntijataitoja tietotekniikan ja (liike)toiminnan rajamaastoista. PTSILIUS on yritysmuodoltaan yksityinen elinkeinonharjoittaja, jonka päätoimiala on muu liikkeenjohtamisen konsultointi.

PTSILIUS -yrityksen web-sivut ovat olleet yksinkertaiset, koska elinkeinonharjoittaja on keskittynyt asiantuntijatyöhön, mihin ei kuulu web-ohjelmointi. Kaikki yhteydenotot yrityksen asiakaskunnassa ovat tapahtuneet henkilökohtaisten kontaktien kautta, ja nyt olisi tarkoitus laajentaa kontaktipintaa tässä insinööriyössä rakennettavilla verkkopalveluilla.

PTSILIUS on asiantuntijayritys, jolla ei ole omia ohjelmistolisenssejä, ja web-sivustonkin paikka oli ostettu teleoperaattorin web-hotellista. Toimeksiannon aikana kuitenkin kävi ilmi, ettei teleoperaattorin web-hotelli tarjonnut sopivaan hintaan tarvittavia tietokantatukia, joten web-hotelli vaihdettiin toiseen palveluntarjoajaan. Tämän toimeksiannon lähtökohtana on avoimen lähdekoodin (open source) -ohjelmistojen käyttäminen. Näin asiantuntijayritys säilyttää edelleen periaatteensa ohjelmistolisenssin ostamisen välttämiseksi. Avoimen lähdekoodi ohjelmistojen käyttö on viime aikoina lisääntynyt yritysmaailmassa. Lisenssipohjainen ohjelmistotuote on normaalisti yhden omistajan hallinnassa, mutta julkisesti saatavilla oleva avoin lähdekoodi luo omistautuneen yhteisön, mikä vaikuttaa positiivisesti laatuun ja turvallisuuteen ohjelmiston laajan käytön ja palautteen kautta.

Avoimen lähdekoodin käytöllä on selvät edut asiantuntijayritykselle. Vaikka toteutuksesta koituukin kustannuksia, ovat kokonaiskustannukset varsinkin pitkällä tähtäimellä silti huomattavasti alhaisemmat verrattuna ohjelmistolisenssikohtaisiin ratkaisuihin. Yrittäjä voi aina valita eri toimittajien välillä tai ylläpitää järjestelmää sisäisesti riippu-



matta toimittajasta. Tukipalvelun hankkimiselle ei ole veloitetta, mihin tahansa järjestelmää käytetäänkään. Toiminnallisuuksia voi kehittää tarpeiden mukaan luomalla laajennuksia tai integroimalla verkkopalvelu muiden järjestelmien kanssa ilman rajoituksia. Jos sopivaa toiminnallisuutta ei löydy valmiista valikoimasta tai kehitysyhteisöstä, useimmat avoimen lähdekoodin järjestelmät mahdollistavat myös tarpeellisten kolmansien osapuolien järjestelmien ja moduulien saumattoman integroinnin.

Sivuston testiympäristö luotiin lataamalla XAMPP-niminen ohjelmisto omalle koneelle. Xampp on paketti, joka helpottaa käyttäjän ongelmia tämän luodessa itselleen toimivaa palvelinta. Xampp sisältää Apache web-palvelimen, PHP:n, MySQL:n sekä PERL:n. Tätä työtä tehdessä käytössä oli XAMPP-paketti johon kuuluu Apache 2.2.21, MySQL 5.5.16, PHP 5.3.8, OpenSSL 1.0.0e, phpMyAdmin 3.4.5, XAMPP Control Panel 2.5, Webalizer 2.23-04, Mercury Mail Transport System v4.72, FileZilla FTP Server 0.9.39, Tomcat 7.0.21. Asennus sujuu hyvinkin helposti joko käyttämällä valmista asennustiedostoa tai .zip-tiedostoa, jos ei halua käyttää asennustiedostoa. XAMPP-paketti löytyy asennusohjeineen osoitteesta <http://www.apachefriends.org/en/xampp.html>.

### **3 Sisällönhallintajärjestelmiä tukevat teknologiat**

Sisällönhallintajärjestelmät pohjautuvat vapaan lähdekoodin ohjelmiin. Sisällönhallintajärjestelmät on hyvin usein luotu PHP-kielellä, koska se sopii loistavasti yhteen HTML-kielen kanssa. HTML:n ja PHP:n lisäksi sisällönhallintajärjestelmissä tarvitaan yleensä SQL-tietokanta, joka hyvin suurella todennäköisyydellä on MySQL-kanta. Koska sisällönhallintajärjestelmät pohjaavat näihin oheisteknologioihin sekä vaativat niiden käyttöönottoa ennen kuin sisällönhallintajärjestelmiä voidaan käyttää, on tärkeää tuntea näitä teknologioita jonkin verran. Tutustutaan siis niistä tärkeimpiin sekä mahdollisesti tunnetuimpiin.

#### **3.1 PHP**

PHP (HyperText Preprocessor) on skriptikieli, joka on tarkoitettu palvelinpuolen ohjelmointiin. WWW-palvelimelle asennettava PHP-ohjelmisto tulkaa PHP-ohjelman. Tulkaamisesta generoitunut tulos palautetaan asiakkaalle, yleensä WWW-selaimelle, esitettäväksi. [2, s. 3.]

PHP, jonka nykyisin tunnemme, on itse asiassa seuraaja tuotteelle nimeltä PHP/FI. Sen loi vuonna 1994 Rasmus Lerdorf. Hänen kirjoittamansa ensimmäinen sovellus oli yksinkertainen joukko C-kielellä kirjoitettuja CGI (Common Gateway Interface) -skriptejä. Alun perin kirjoitetut skriptit oli tarkoitettu pitämään kirjaa vierailuista Lerdorfin internetissä sijainneeseen ansioluetteloon. Ajan kuluessa skripteihin haluttiin lisää toiminnallisuutta, jolloin Lerdorf uudelleen kirjoitti ne luoden paljon laajemman ja rikkaamman kokoonpanon. Tämä uusi malli kykeni hakemaan tietoa tietokannoista sekä loi puitteet joiden varaan käyttäjät pystyivät luomaan yksinkertaisia dynaamisia sovelluksia, kuten vieraskirjoja. Vuoden 1995 keväällä Lerdorf julkaisi järjestelmänsä lähdekoodin julkiseen käyttöön. Tämä innoitti ja salli kehittäjien tuottaa ohjelmointivirheiden korjauksia sekä yleisesti parantamaan koodia. [12.]

Vuonna 1997 israelilaiset opiskelijat Andi Gutmans ja Zeev Suraski lähestyivät Lerdorfia keskustellakseen sen hetkisestä koodista eri näkökohdista sekä suunnitelmissaan olevasta koodin uudelleenkehityksestä. Kolmikko päätti tehdä yhteistyötä luodakseen aivan uuden itsenäisen ohjelmointikielen. Tämä kieli julkaistiin uudella nimellä, joka poisti viittaukset rajoitetusta henkilökohtaisesta käytöstä, joita PHP/FI sisälsi. Tästä alkoi nykyisin tuntemamme PHP:n kehitys niin, että vuoden 1998 heinäkuussa julkaistiin PHP 3.0. Nykyisin ollaan menossa PHP 5:ssä ja arvellaan, että PHP on nykyisin asennettuna kymmeneen, ellei jopa satoihin miljooniin verkkotunnuksiin ympäri maailman.

PHP on korkean tason kieli niin kuin useimmat muutkin nykyiset ohjelmointikielät. Tämä merkitsee sitä, että ohjelmointi tapahtuu käsittein, jotka ovat tuttuja ja luonteenomaisia toisin kuin esimerkiksi symbolisessa konekielessä, jonka ohjelmointi edellyttää ohjelmoijan tuntevan tietokoneen sisäistä toimintaa. PHP muistuttaa suuresti C++-kieltä, ja se onkin hybridikieli. Ohjelmissa voi olla luokkia sekä perinteistä lineaarista koodia.

## 3.2 MySQL

MySQL on hallintajärjestelmä SQL-tietokannoille. SQL (Structured Query Language) on kieli relaatiotietokantojen käsittelyyn. SQL:n ensimmäisen standardin kehitti ANSI (American National Standards Institute) vuonna 1986, ja vuonna 1987 ISO (International Standards Organization) otti sen omakseen. Vuosina 1989 ja 1992 SQL-

standardia uusittiin ja laajennettiin. Vaikka SQL-standardia pidetään valmiina eikä niinkään innovatiivisena uutena teknologiana, mitä se oli, kun standardin rakennus 1980-luvulla alkoi, on se vieläkin laajeneva ja kehittyvä standardi.

MySQL:n kehittivät ystävykset Michael Widenius, David Axmark ja Allan Larsson vuonna 1994. He päätyivät perustamaan yrityksen nimeltä MySQL Ab, joka hallinnoi MySQL:n lisenssiä sekä tarjosi yrityksille, joiden mielestä vapaan ohjelmakoodin ohjelmistot eivät sovellu kyseiselle firmalle, maksullista lisenssiä MySQL:stä. Vuonna 2008 MySQL Ab:n osti Sun Microsystems, joka lupasi jatkaa MySQL:n niin sanottua tuplalisenssiä. [9.]

Kuitenkin jo vuonna 2010 osti Oracle Sun Microsystemsin, mikä sai yhden alkuperäisistä kehittäjistä Michael Wideniuksen aloittamaan yleisen kampanjan myyntiä vastaan, koska hän ei uskonut Oraden pitävän MySQL:n tuplalisenssiä enää voimassa [8.]. Oracle on luvannut pitää tuplalisenssin voimassa vuoteen 2015 asti, jonka jälkeen ei ole varmuutta siitä, pysyykö MySQL vielä vapaan ohjelmakoodin ohjelmana.

### 3.3 HTML ja XHTML

HTML (HyperText Markup Language) on dataformaatti, jolla tuotetaan eri ympäristöihin siirtokelpoisia hypertekstiasiakirjoja. Samalla se on menetelmä organisoida ja järjestää dataan liittyvää informaatiota. HTML-asiakirja koostuu osittain datasta, varsinaisesta tekstistä ja sivulla mahdollisesti olevista kuvista ja HTML-koodin muodostavista tunnistuksista eli metadatatista. HTML:n ehkä tärkein ominaisuus on siirrettävyys, eli HTML-asiakirja voidaan siirtää ympäristöstä toiseen menettämättä informaatiota, esimerkiksi vaikkapa Windows-ympäristöstä Linux-ympäristöön. Tämä ei kuitenkaan tarkoita, että kaikki HTML:n esittämiseen tarkoitetut järjestelmät olisivat samanarvoisia. Eriarvoisuuden tuloksena käytännössä jokaista tietokonejärjestelmää varten on kehitetty omat HTML-tulkkinsa. HTML:n kehittämisen alkaessa kyseessä oli hypertekstijärjestelmä. Sen tarkoituksena oli esittää tekstiä, joka sisälsi linkkejä muuhun tekstiin sekä joskus muihin esitystapoihin kuten kuviin ja ääneen. Ajan kuluessa HTML:ään on tullut lisää toimintoja, joiden avulla esitystapoja voidaan nykyisin käyttää suoraan tekstin esittämisen yhteydessä. [7, s.1-3.]

XML tulee sanoista eXtensible Markup Language. Siinä missä HTML-ohjelmissa käytetään ennalta kielen spesifikaatiossa määrättyjä tunnisteita, sallii XML omien tunnisteiden käytön. Silloin kun X-kirjain lisätään HTML-sanaan, tarkoittaa se, että sovelletaan XML-kielen mukaisen XHTML-kieliopin sääntöjä, joita koodin tulee noudattaa. Ne saatavat näyttää ensisilmäyksellä tiukkapipoisilta ja teknisiltä, mutta niistä on paljon hyötyä. Eliminoimalla joitakin tavallisen HTML:n sallimia koodin huolimattomuuksia, nämä lisäsäännöt tekevät koodista parempaa. [6, s.13.]

Kuten yllä mainittiin on XHTML:ssä sääntöjä, jotka poikkeavat tavallisesta HTML-koodista [6, s. 391-392]. Tässä on niistä muutama:

1. HTML:ssä ei ole väliä sillä, kirjoitetaanko tunnisteet pienin vai isoin kirjaimin vai niitä yhdistellen. XHTML:ssä kaikki tunnisteet tulee olla kirjattu pienin kirjaimin.
2. Lopputunnisteet ovat yleensä valinnaisia HTML:ssä, mutta XHTML:ssä ne ovat pakollisia kaikissa elementeissä, jotka sisältävät jotakin. Kaikkien kappaleiden täytyy esimerkiksi alkaa `<p>`-tunnisteella ja päättyä `</p>`-tunnisteeseen.
3. HTML-tunnisteiden, jotka eivät sisällä mitään sisältöä, kuten `<br>` ja `<hr>`, täytyy nyt sisältää kauttaviiva, esimerkiksi `<br />` ja `<hr />`. Tämä kertoo XHTML-tulkille, ettei sen pidä odottaa lopputunnistetta.
4. XHTML:ssä kaikki attribuuttien arvot täytyy laittaa lainausmerkkeihin. Esimerkiksi `<img src = kuva.gif border = 1>` oli korrektia HTML-kieltä, mutta XHTML:ssä se täytyy kirjoittaa ``.
5. XHTML:ssä kaikilla attribuuteilla tulee olla arvo. Esimerkiksi `<input type="checkbox" checked>` pitää kirjoittaa `<input type="checkbox" checked="checked" />`.
6. Tietyt erikoismerkit eivät ole sallittuja HTML:ssä, koska selain saattaisi tulkita ne tunnisteiden osaksi. Samat merkit ovat kiellettyjä XHTML:ssä, mutta se on paljon ankarampi siinä, ettei se salli niitä upotetuissa tyylitiedostoissa eikä skripteissä.

Edellä mainitut asiat kertovat, kuinka HTML ja XHTML toimivat yhdessä ennen HTML5:tä ja XHTML5:tä tukevien web-selaimien ilmestymistä. HTML5 on HTML-kielen uusi versio, mutta myös paljon muuta. Lähes kaikki web-sivujen ja -sovellusten uudet tekniikat luetaan HTML5:een kuuluviksi. HTML5 rakentuu vähittäisen kehityksen (evoluution) eikä kumouksen (revoluution) ajatukselle. Tällä haluttiin varmistaa, että vanhat sivut sekä HTML-rakenteet toimivat, ja sivuihin voidaan lisätä uusia piirteitä sitä mukaa kuin niitä toteutetaan selaimiin ja ne koetaan tarpeellisiksi. HTML5:ssä käytetään samanlaisia .html-loppuisia tiedostonnimiäkin kuin vanhoissa HTML-sivuissa. Kehityspeariate ei ole ollut itsestäänselvyys, sillä HTML:n vanhoissa versioissa on paljon menneisyyden painolastia. Pahinta on, että selaimissa on paljon määrittelyjen vastaista toimintaa, johon monet sivujentekijät ovat oppineet luottamaan. Toisin sanoen toiminta on ollut puutteellisesti dokumentoitua ja osittain erilaista riippuen selaimesta.

Puhtaat selainkohtaiset laajennukset eivät ole kuitenkaan yhtä ongelmallisia kuin kielen määrittelyjen piirteiden toteuttaminen määrittelyjen vastaisesti. Ajatus siitä, että HTML pitäisi suunnitella kokonaan uudelle pohjalle vanhoista versioista riippumattomiksi, on ollut houkutteleva. Siitä on kuitenkin näillä näkymin luovuttu. Yhtenä syynä tähän on, että web-sivujen kehittäminen riippuu enimmäkseen muista seikoista kuin HTML:stä suppeassa merkityksessä, merkkäuskielenä. Toiminnallisuutta, ulkoasua ja muita asioita voidaan kehittää enimmäkseen HTML:n ulkopuolella, vaikka kehitys saattaa jossain vaiheessa vaatia myös pieniä lisäyksiä HTML:ään. Vaikka selaimiin on jo toteutettu monia HTML5:n piirteitä, HTML5 on luonnos tai oikeastaan kokoelma luonnoksia. Tästä johtuen, kun asiasta lukee, kannattaa HTML5 ymmärtää merkityksessä nykyinen HTML5-luonnos. [11, s. 24-25.]

HTML5:n määrittely kuvaa mahdolliset oliot ja niiden ominaisuudet. Se, miten selain esittää dokumentin käyttäjälle, taas riippuu selaimesta ja etenkin CSS-tyyliohjeista. HTML-kielinen lähdekoodi käsitetään abstraktin dokumentin tekstimuotoiseksi esitykseksi, eli sarjallistukseksi. Jotta dokumentti voitaisiin lähettää esimerkiksi palvelimelta selaimen, se on jotenkin saatettava "sarjamuotoon". Olioiden puu on muutettava peräkkäistekstiksi, ja tämä tapahtuu käyttämällä tunnisteita, joiden perusteella vastaanottaja voi rakentaa olioiden puun omiin tietorakenteisiinsa. HTML5 määrittelee kaksi vaihtoehtoista sarjallistusta:

- HTML-sarjallistus, joka vastaa HTML:n vanhaa esitysmuotoa. Toisin kuin vanhoissa HTML:n määrittelyissä, sitä ei kuitenkaan ole määritelty SGML-merkkaukielen sovelluksena vaan täysin itsenäisesti. Pois on jätetty ne SGML:n piirteet, joita ei todellisuudessa koskaan toteutettu selaimiin.
- XHTML-sarjallistus, joka on täysin XML-merkkaukielen mukainen, eli XML-kielen sovellus. Määrittely siis edellyttää pohjaksi XML:n, mutta on muutoin itsenäinen, XHTML:n aiemmista versioista riippumaton.

Yksinkertaisimmassa tapauksessa dokumentti kirjoitetaan suoraan sarjallistettuun muotoon vaikkapa muistio-ohjelmalla tai jollakin sivunkeho-ohjelmalla, joka sisäisesti käyttää jotain puurakennetta mutta tallentaa dokumentin sarjallistettuna. Jos palvelimella on käytössä sisällönhallintajärjestelmä, dokumentit tai niiden osat ovat olemassa sen tietorakenteissa, ja se tuottaa sarjallistuksen aina uudestaan, kun siltä pyydetään sivua. [11, s. 312.]

HTML5:n XHTML-sarjallistuksesta käytetään – jopa HTML5-luonnoksissa – nimitystä XHTML5. Tämä on hämmentävää: edellinen XHTML:n määrittely on XHTML 1.1 ja XHTML 2.0:n kehittäminen jäi kesken. Syynä outoon numerointiin on se, että XHTML5 tarkoittaa vain HTML5:n XHTML-muotoa. XML asettaa useita rajoituksia merkkauksen syntaksille. Jos on todennäköistä, että HTML5-koodia tullaan käyttämään myös XHTML-sarjallistettuna, kannattaa siksi käyttää alusta alkaen sen mukaista merkkausta. Jos HTML5:lle käytetään XHTML-sarjallistusta, niin HTML5-merkkauksen seassa voidaan käyttää muitakin XML-pohjaisia kieliä. Tämä tarjoaa hyvin laajoja näköaloja, sillä XML-pohjaisia kieliä on suuri ja kasvava määrä. Tosin käyttö edellyttää, että selain pystyy käsittelemään niitä ”natiivisti” upotettuina XHTML-dokumentteihin.

Toinen syy käyttää XHTML:ää on se, että dokumentteja mahdollisesti käsitellään erilaisilla XML-työkaluilla. Niiden ei voi olettaa osaavan vanhan HTML:n kaltaista ”tunnistesalaattia”, vaan merkkauksen on täytettävä tarkasti XML:n muotovaatimukset. Web-palvelimella saattaa esimerkiksi olla toiminto, joka poimii haluttujen sivujen sisällöt ja lähettää ne toiselle palvelimelle jotenkin käsiteltäväksi, esimerkiksi kielenkääntäjälle tai vietäväksi tietokantaan. Tällaisissa toiminnoissa käytetään nykyisin yleisesti XML-muotoa, jolloin on luonnollista, että palvelimella olevat sivut ovat valmiiksi XML-muodossa (XHTML:nä). [11, s. 313-314.]

### 3.4 CSS

Web-selain pystyy esittämään verkkosivun, vaikka mitään muotoiluohjeita ei anneta. Sivuja kuitenkin yleensä halutaan muotoilla eri syistä [10. s. XII]:

- Selainten oletusarvoista tapaa esittää verkkosivu pidetään typografisesti huonona. Esimerkiksi oletusfonttina on yleensä Times New Roman, joka ei kovin hyvin sovi kuvaruudulta luettavaksi.
- Selainten oletusarvoista tapaa voidaan pitää tylsänä ja mielikuvituksettomana. Sivuille halutaan väriä sananmukaisesti ja kuvainnollisesti.
- Sivun muotoilulla voidaan edistää sivun sanoman perillemenoä muun muassa korostamalla keskeisiä asioita ja sivun osien suhteita toisiinsa. Esimerkiksi taustaväriin käytöllä voidaan tehokkaasti nostaa tärkeitä tekstejä esille.
- Muotoilemalla sivuston sivut yhtenäisen tyylin mukaisesti voidaan yritykselle tai muulle organisaatiolle luoda omintakeinen ilme. Tällöin sivujen kävijöiden on helppo mieltää sivun samaan kokonaisuuteen kuuluviksi ja tunnistaa ne myöhemminkin.

CSS eli Cascading Style Sheets on merkintäjärjestelmä, toisin sanoen kieli, jolla voidaan antaa ohjeita web-sivujen ja muidenkin dokumenttien ulkoasuista. Näitä ohjeita voidaan kutsua monella eri nimellä, kuten esimerkiksi tyyliohjeet, tyylisivut, tyyli tiedostot ja vaikkapa tyyli lomakkeet, englanniksi style sheet. Tyyliohjetta voidaan verrata ohjelappuseen, jollaisella ennen annettiin latojalle ohjeita. Ladottavan tekstin mukana lähetettiin erillinen ohje siitä, mitä kirjasinlajeja käytetään, miten teksti tulee asemoida jne. CSS poikkeaa tästä olennaisesti siinä, että ohjeet annetaan tiukasti määrämuotoisina niin, että tietokoneohjelmat voivat ”ymmärtää” ja soveltaa niitä, ja ne myös soveltavat, näytti tulos miten hirveältä tahansa. [10, s. XII-2.]

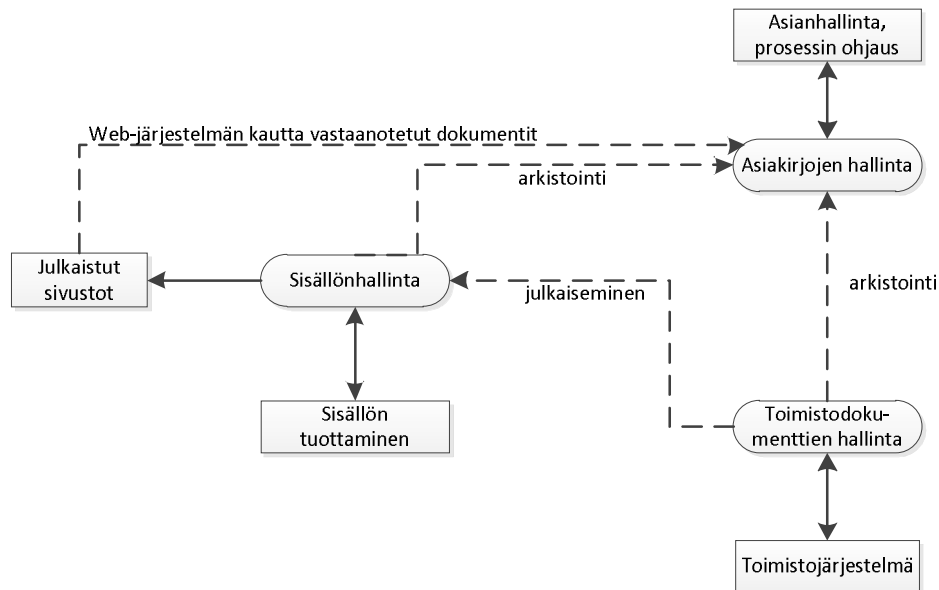
Käytännössä CSS:llä muotoillaan yleensä websivuja, tarkemmin sanoen HTML-muotoisia dokumentteja, joita web-selaimet esittävät. Lähtökohtana on, että selain joka tapauksessa esittää dokumentin jollakin tavalla käyttäen joitakin perusasetuksia, eli oletusasetuksia. CSS:llä esitetään poikkeamia näistä asetuksista sekä annetaan lisä-

yksiä niihin. Yksi keskeisistä ideoista on, että CSS-tyyliohje esittää ulkoasuehdotukset erillisenä kokonaisuutena, jonka selain voi ottaa huomioon tai jättää huomioimatta. Voidaan esimerkiksi tehdä erikseen tyyliohje kuvaruudulle esittämistä varten ja toinen tyyliohje paperitulostusta varten. Tilanne on osittain toinen silloin, kun CSS:ää käytetään yhdessä XML:n kanssa. XML-dokumentin esittämiseksi on CSS:n tai jonkin ilmaisuvoimaltaan vastaavan järjestelmän käyttö välttämätöntä, koska XML on pelkkää muotoa: millään XML-rakenteella ei ole mitään ennalta määrättyä, oletusarvoista esitysasua. Tämä merkitsee, että XML-tiedostoon liittyvän CSS-tiedoston täytyy kuvata kaikki esitysasun olennaiset piirteet. Käytännössä XML-muodosta usein tuotetaan muita esitysmuotoja, esimerkiksi HTML-muoto, sopivilla ohjelmilla sen sijaan, että sitä suoraan esitettäisiin CSS-tyyliohjeen avulla. [10, s. 2.]

#### **4 Soveltuvia sisällönhallintajärjestelmiä**

Sisällönhallintajärjestelmä (englanniksi CMS eli Content Management System) on yleisnimitys tietojärjestelmälle, joka palvelee koko organisaation sisällönhallintaa sen sijaan, että olisi keskittynyt pelkästään johonkin yksittäiseen osa-alueeseen, kuten verkkopalveluiden hallintaan. Sisällönhallinta tarkoittaa laajimmillaan kaikkea digitaalisen sisällön ylläpitoa kattaen koko elinkaaren tuottamisesta julkaisemiseen ja edelleen mahdolliseen poistamiseen tai arkistointiin. Sisällönhallinta ei ole vain web-sivujen käsittelyä, vaan koskee yhtä lailla myös erilaisia kuvan ja äänen tallennusmuotoja. Tämän lisäksi sivutaan tietotekniikan suurta sovellusaluetta, dokumenttien hallintaa (kuva 1). Sisällönhallinta onkin monessa mielessä hyvin lähellä sovellusaluetta, jota on aikaisemmin kutsuttu sähköisten dokumenttien hallinnaksi. [3, s. 7.]





Kuva 1. Dokumenttien hallinnan kuvaus.

Sisällönhallintajärjestelmälle ei täten ole olemassa mitään kovin selkeätä määritelmää vaan toimialasta riippuen sillä saatetaan viitata eri tavoin painottuneeseen tietojärjestelmään [1]. Niitä ovat

- dokumenttienhallintajärjestelmä
- www-sisällönhallintajärjestelmä
- julkaisujärjestelmä
- verkkokauppajärjestelmä
- aineistohallintajärjestelmä
- wiki-järjestelmät.

Yksinkertaiset organisaation esittelysivut Internetissä eivät yleensä kaipaa taustalleen sisällönhallintajärjestelmää. Tosin järjestelmälliset menettelytavat ovat niidenkin hallinnassa tarpeen. Yrityksen sivustojen kehittymisen myötä aineiston määrä kasvaa ja päivitystahti tihenee. Näissä tapauksissa organisaation sisäisen verkon ja ulkoisen sivuston hallintaan tulisi harkita yhtenäistä sisällönhallintajärjestelmää. Markkinoilla onkin

monia erilaisia kaupallisia sovelluksia, jotka soveltuvat joko yksityisen tai yrityksen käyttöön. [3, s. 16.]

WWW-sisällönhallinnalla pyritään hallitsemaan verkkopalvelujen sisältöä mahdollisimman tarkoituksenmukaisesti. Se on luonteeltaan julkaisupainotteista sisällönhallintaa, jossa keskiössä ovat sivupohjat. Onkin tärkeää, että kartoittaa tarpeensa ennen kuin aloittaa järjestelmän valitsemisen. Vasta omien tarpeiden selvittämisen jälkeen (tässä tapauksessa asiakkaan antamat tavoitteet, monikielisyys ja avoimen lähdekoodin ohjelmisto) kannattaa etsiä tietoa vaikkapa internetistä. Esimerkiksi osoitteesta <http://php.opensourcecms.com/> löytyy yli 100 ohjelmistoa, joista voi valita tarvitsemansa. Valintaa helpottaa sivustolla sijaitseva arviointi, jossa käyttäjät ovat voineet ilmaista tyytyväisyytensä tai tyytymättömyytensä kyseessä olevaan ohjelmistoon.

WWW-sisällönhallintajärjestelmien suosio monesti perustuu niiden käytön helppouteen, mutta myös sivupohjien laatimisen mahdollisuuteen. Sivupohjalla tarkoitetaan halutunlaisen sivun luomista niin, että ohjelmoimalla luodaan puitteet, jotka saavat sivuston sisällön näyttämään uniikilta. Tämä vaatii kuitenkin monesti ohjelmointitaitoa, sillä sivupohjat luodaan käyttämällä CSS-, PHP- ja HTML-kieliä. Moneen www-sisällönhallintajärjestelmään on kehitetty sivupohjan luontiohjelmia, joiden avulla luonti onnistuu ilman koodaamista. Koodin lisäksi onnistunut sivupohja vaatii yleensä Photoshop-ohjelman tai muun kuvan käsittelyohjelman apua. Näiden avulla saadaan sivuston taustakuvat ja mahdolliset mainospalkit näyttämään hyviltä. Jos kuitenkin valitsee aloittavansa sivupohjan luomisen itse koodaamalla, ei se ole loppujen lopuksi kovin monimutkaista, kunhan sivusto pysyy yksinkertaisena.

Sivupohjaa luotaessa esimerkiksi Joomlaan tarvitaan siihen vähintään kolme tiedostoa, joista tärkein on templateDetails.xml-tiedosto (koodiesimerkki 1). Tiedostossa kerrotaan ensin sivupohjan luontitietoja kuten tekijä, luontipäivä, onko kyseessä vapaan lähdekoodin ohjelmisto vai maksullinen, pohjan mahdollinen versio ja kuvaus pohjasta. Yleistä selostusta seuraa files-osio, jossa käydään läpi, mistä kaikista tiedostoista pohja rakentuu. Tässä tapauksessa pohja rakentuu kahdesta tiedostosta mukaan lukien templateDetails.xml-tiedosto ja kaksi kansiota. Files-osiossa voidaan jokainen tiedosto luetella erikseen kuitenkin, koska CSS-tiedostoja ja kuvia voi olla kuinka monia. On hyvä esitellä nämä yleensä omina kansioinaan. Viimeisimpänä käydään läpi, minkälaisia

sijoituspaikkoja Joomlaan eri elementeille voi antaa. Esimerkiksi määreellä left kuvataan, että elementtejä voidaan sijoittaa vasemmalle sivustossa.

```
<?xml version="1.0" encoding="utf-8"?>
<!DOCTYPE install PUBLIC "-//Joomla! 1.6//DTD template 1.0//EN"
"http://www.joomla.org/xml/dtd/1.6/template-install.dtd">
<extension version="1.7" type="template" client="site">
  <name>pirjo</name>
  <creationDate>2011-13-11</creationDate>
  <author>Ville Miettinen</author>
  <copyright>Ville Miettinen 2011</copyright>
  <license>GNU/GPL</license>
  <version>1.0.0</version>
  <description>Testi Template</description>
  <files>
    <filename>index.php</filename>
    <filename>templateDetails.xml</filename>
    <folder>images</folder>
    <folder>css</folder>
  </files>
  <positions>
    <position>breadcrumb</position>
    <position>left</position>
    <position>right</position>
    <position>top</position>
    <position>user1</position>
    <position>user2</position>
    <position>user3</position>
    <position>user4</position>
    <position>footer</position>
  </positions>
</extension>
```

Koodiesimerkki 1. HTML-tiedosto tarvittavine tietoineen.

Toinen vaadituista tiedostoista on index.php (koodiesimerkki 2). Tämä tiedosto on ainoa, jossa tarvitaan HTML- ja PHP-koodia, ja se on ainoa tiedostoista, joka välittää sisällön tiedot käyttäjälle. Koodit tulee kirjoittaa niin, että HTML toimii koodin pohjana ja siihen sitten lisätään PHP-tunnisteiden sisään kaikki kohdat, jotka tarvitsevat PHP:ta toimiakseen. Kaikissa index.php-tiedostoissa tulee olla määritelty tiedot, jotka näkyvät esimerkin alusta head-osan loppuun. Huomionarvoista tässä osassa on koodin ensimmäinen rivi, joka varmistaa sen, ettei sivupohjaan pääse suoraan käsiksi. Se on tietoturvallisuuden kannalta aina lisättävä. Tämän lisäksi huomionarvoisia ovat kohdat `$this->language`, jossa määritetään se, että sivupohja voi olla monikielinen, koska siinä ei aseteta vain tiettyä kieltä sivupohjan käyttöön, ja `$this->template`, jossa

määritettään, että kyseessä on juurikin hallittu sivupohja. Body-osassa käydään sitten läpi sisällön esityksen ja siinä tulee käyttää CSS-tiedostossa esiintyviä nimityksiä, jotta CSS-tiedoston luomat tyylit tulevat käyttöön. Tästä esimerkkinä div id=content, jolloin CSS-tiedoston content-osion tyyli sovelletaan tähän kohtaan.

```
<?php defined( '_JEXEC' ) or die( 'Restricted access' );?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="<?php echo
$this->language; ?>" lang="<?php echo $this->language; ?>" >
<head>
<jdoc:include type="head" />
<link rel="stylesheet" href="<?php echo $this->baseurl
?>/templates/<?php echo $this->template;?>/css/template.css"
type="text/css" />
</head>

<body>
<div id="container">
    <div id="header">PTSILIUS ASIANTUNTIJAYRITYS</div>
    <div id="sidebar_left" class="float"><jdoc:include
type="modules" name="left" /></div>
    <div id="content" class="float"><jdoc:include
type="component" /></div>
</body>
</html>
```

Koodiesimerkki 2. PHP-tiedoston esimerkki.

Kolmas vaadittu tiedosto on template.css (koodiesimerkki 3). Tähän tyylitiedostoon sitten luodaan tarvittavat tiedot haluttua tyyliä varten. Esimerkkinä on jo aikaisemmin mainittu content, joka tässä tapauksessa esitellään ensin sanalla #content. Sen jälkeen kerrotaan, että sen leveys on 660 pikseliä, ja mahdollisen sisällön tekstin tulee olla keskitetty. Edellä mainittujen tyyliasetusten lisäksi voidaan esimerkiksi määrittää haluttu taustaväri, korkeus tai vaikka marginaali.

```
* {
    padding: 0;
    margin:0;
}
ul {
    list-style:none;
}
.float {
    float: left;
}
.clear {
```

```

        clear: both;
    }
#container {
    width:960px;
    margin: auto;
}
#header {
    background-color:#999999;
    height: 150px;
}
#content {
    width: 660px;
    text-align: center;

}
#sidebar_left {
    text-align: center;
    background-color:#CCCCCC;
    width: 300px;
}
#sidebar_right {
    background-color:#CCCCCC;
    width: 90px;
}
#footer {
    background-color:#999999;
    text-align:center;
}

```

Koodiesimerkki 3. CSS-tiedoston määrittäjiä.

Tämän työn toteutukseen tarvittiin yksi sisällönhallintajärjestelmä. Valintaa varten tutkittiin muutamia ohjelmistoja. Jotta sisällönhallintajärjestelmät saadaan toimimaan, tulee käyttäjän luoda itselleen palvelinympäristö, johon hyvin usein kuuluu PHP:n ja jonkin tietokannan asentaminen.

#### 4.1 Vapaan lähdekoodin sisällönhallintajärjestelmät

Kuten aikaisemmissa luvuissa tuli todetuksi, on vapaan lähdekoodin ohjelmistoja hyvin monia. Tutkimukseen valittiin mukaan kuusi sisällönhallintajärjestelmää, joita tutkimalla oli tarkoitus vähentää ehdokkaiden määrä kahteen, jotka käytäisiin läpi tarkemmin. Valinta tapahtui menemällä sivustolle <http://php.opensourcecms.com/> ja valitsemalla sieltä top 25 -joukosta ne ohjelmat, jotka vaikuttivat asiakkaan kannalta helpoimmilta käyttää. Nämä ohjelmat ovat Joomla!, Drupal, WordPress, Concrete5, CMS made Simp-

le ja Etomite. Tosin valinnan jälkeen Drupal on tippunut alle top 25:n sen julkaistua uuden 7.10-version.

Joomlaa käytetään kaikkialla maailmassa, ja se on yksi suosituimmista ellei jopa suosituin sisällönhallintajärjestelmä. Joomla sopii erinomaisesti sivustoille, joilla ylläpitoa ja sisällöntuottamista halutaan jakaa useamman henkilön kesken. Sisällöntuottaminen vaatii käytännössä vain tekstieditorin käyttötaidon, mikä helpottaa tuotantoa suuresti. Joomla'n vahvuudet ovat sen laaja levinneisyys, jolloin esimerkiksi sivupohjia ja erilaisia työskentelyä helpottavia lisäosia on helppo löytää, sitä kehitetään jatkuvasti eikä ole pelkoa järjestelmän kaupallistamisesta. Pelkoa ei ole, koska Joomla-projekti alkoi aikoinaan Mambo-nimisen sisällönhallintajärjestelmän sivuprojektina, joka siirtyi omaksi projektikseen kehittäjien pelätessä Mambon kaupallistamista.

Drupal on sisällönhallintajärjestelmä, joka alun perin kehitettiin opiskelijaporukan viestintävälineeksi. Tätä ideaa sitten lähdettiin kasvattamaan, ja Drupal syntyi. Tätä nykyä Drupal on GPL-lisenssin alainen, ja se tukee monia verkkosivuja, blogeista yritysesittelyihin ja edelleen suuriin yhteisövetoiisiin sivuihin. Drupal on suhteellisen monimutkainen järjestelmä, johon tarvitsee paneutua jonkin aikaa, jotta järjestelmän oppii. Drupalin vahvuudet ovat sen muokattavuudessa, joka on huippuluokkaa, tosin viimeisimpänä julkaistu versio 7 on monien mielestä paisunut liikaa. Paisumisella tässä tapauksessa tarkoitetaan sitä, että ydinversion asennuksen mukana tulee niin paljon kaikkea, että sivuston luominen on muuttunut jo liian monimutkaiseksi pelkän ydinversion asentajille. Yleensä ydinversio sisältää sivuston luomisen kannalta tarvittavat osat, Drupal versio 7:n mukana siis tulee monien mielestä liikaa sivuston luomisen kannalta turhia moduuleita, jotka vaikeuttavat ohjelmiston oppimiskäyrää suuresti.

WordPress on vuonna 2003 aloitettu projekti, jonka tarkoituksena oli helpottaa kirjoittamista ja sen suunnittelua. Projekti alkoi pelkkänä blogityökaluna mutta on sittemmin kasvanut täysiveriseksi sisällönhallintajärjestelmäksi. WordPress syntyi halusta luoda elegantti ja hyvin suunniteltu henkilökohtainen julkaisujärjestelmä. Huolimatta Wordpressin sisällönhallintajärjestelmästatuksesta, on sen suurin vahvuus siinä, että blogikirjoitukset voidaan pitää täysin omana versiona. Tällöin niitä voidaan kommentoida ilman mitään kirjautumisia, mikä yleensä saattaa sivuston alttiiksi spam-viesteille. Tähän on kuitenkin varauduttu, ja Wordpressin ydinversion mukana tulee mahdollisuudet

listata epäsuosittuja käyttäjiä sekä avoin tarkastusohjelma spam-viestien hallintaan ja torjumiseen. Verkosta löytyy myös monia ohjelmia, jotka vievät toiminnon pidemmälle.

Concrete5 on viimeisin versio Concrete-sisällönhallintajärjestelmistä. Vuonna 2003 kaksi yhdysvaltalaista kehittäjää tekivät paikalliselle yrityksellään nettisivuja, jotka osoittautuivat monimutkaisiksi päätösten tekijöiden ja yleisön useasti toisiaan vastustavien näkökulmien vuoksi. Tämä aiheutti sen, että sisältöä siirrettiin usein paikasta toiseen, eikä määritä-tuota-käy läpi -menetelmä tuottanut haluttua tulosta. Tästä johtuen he tarvitsivat jotakin joustavampaa, ja millä pystyisivät muuttamaan sivustoa helposti vaikuttamatta negatiivisesti projektin aikatauluun. Tämän ideariihen tuloksena syntyi Concrete. Concrete5 sisältää mahdollisuuden muokata sivustoa niin sanotusti lennosta eli samaan aikaan sivustoa selatessa sekä ajastetun sisällön julkaisemisen. Tämä tarkoittaa sitä, että voi luoda julkaisun valmiiksi hyvissä ajoin ja laittaa sen julkaistavaksi haluttuna päivänä.

CMS Made Simple on vuonna 2004 julkaistu projekti, joka tarjoaa websivujen kehittäjille yksinkertaisen ja helppokäyttöisen apuohjelman. Ohjelmalla voidaan luoda pienehköjä, kymmenistä sivuista satoihin sivuihin ulottuvia semistaattisia websivuja. CMS Made Simple soveltuu parhaiten joukkueiden ja yritysten sivustojen luomiseen, eikä niinkään artikkelipohjaisen sivuston tai blogisivuston luomiseen. CMS Made Simple on loppukäyttäjän kannalta yksinkertainen ja sillä pystyykin helposti ylläpitämään vanhaa ja luomaan uutta sisältöä. Kuitenkin sivuston kehittäjällä saattaa olla ongelmia, koska sivupohjien kehitys vaatii kokemusta Smarty Templatesista (järjestelmä toiminto- ja esityslogiikkojen erottamiseksi). Tämä karsii kehittäjiä pois harvinaisuutensa vuoksi, ja käyttöönotto saattaa vaatia jollekulle ulkopuoliselle maksamista.

Etomite syntyi sisältöön keskittyvän sisällönhallintajärjestelmän tarpeesta. Etomiten luoja ei tarvinnut suuria, paljon laajennuksia sisältäviä järjestelmiä vaan yksinkertaisen sivuston hallintaan soveltuvan järjestelmän. Etomiten ensimmäinen versio oli nimeltään Phase 0.2.1, ja se oli vain harrasteprojekti. Kuitenkin versioon 0.5 mennessä kehittäjä oli saanut rohkaisevia näkökulmia ystäviltaan ja siirsi ohjelmiston julkiseksi. Laitettuaan ohjelmistonsa opensourcecms:n foorumeille tiedoksi alkoivat ihmiset raportoida ohjelmointivirheitä ja ilmaista, kuinka he olivat tyytyväisiä Phaseen. Tästä ilahtuneena päädyttiin se listamaan opensourcecms:n sivuille. Julkiseksi tulon jälkeen ottivat erään

yrittäjien asianajajat yhteyttä tekijään ja ilmoittivat, ettei ohjelmisto voisi olla nimeltään Phase. Tästä johtuen uudesta nimestä pidettiin äänestys, ja voittajaksi selvisi Etomite.

Valinnan helpottamiseksi otettiin sisällönhallintajärjestelmistä tutkittavaksi tehtävänannon vastaavuuksia, kuten löytyykö kyseiselle järjestelmälle suomennos tai mikä on järjestelmän tietokantatyypit (taulukko 1). Drupalin kohdalla tehtiin arvostelu kahteen kertaan, koska kesken vertailun julkaistiin siitä versio 7. Näiden asioiden lisäksi arvioitiin hyvin vähän sisällönhallintaohjelmia käyttäneen ihmisen näkökulmasta käytettävyyttä. Yllätyksenä ei tullut, että jokaiseen sisällönhallintajärjestelmään on käytetty kehityskielenä PHP:tä. Sen sijaan yllätyksenä tuli, että sattumanvaraisesti valituista sisällönhallintajärjestelmistä yli puolissa oli mahdollisuus muuttaa järjestelmä tukemaan suomen kieltä. Suomen kielen saaminen järjestelmän käyttökieleksi helpottaa sisällönhallintajärjestelmän käyttöä ja oli vaatimus asiakkaalta. Toisena yllätyksenä oli se, että moni sisällönhallintajärjestelmä tuki pelkästään MySQL-tietokantaa. Tämä oli yllätys, koska kuten jo edellisessä luvussa mainitsin, ei ole takuita siitä, ettei Oracle kaupallistaisi MySQL-kieltä lopullisesti.

Nimi	Kehitys kielet	Suomennos	Tietokanta tyypit	Käytettävyys
Joomla	PHP, (x)HTML	Kyllä	MySQL	Hyvä
Drupal	PHP, (x)HTML	Kyllä	MySQL, SQLite (PostgreSQL)	Välttävä (Kohtalainen)
Wordpress	PHP, (x)HTML	Kyllä	MySQL	Hyvä
Concrete5	PHP, Python, (x)HTML	Kyllä	MySQL	Kohtalainen
CMS made simple	PHP, (x)HTML	Ei	MySQL, PostgreSQL	Hyvä
Etomite	PHP	Ei	MySQL	Välttävä

Taulukko 1. Yleistä tietoa sisällönhallintajärjestelmistä sekä käytettävyyden vertailu.

Käytettävyttä arvioitaessa vertailtiin sitä, kuinka helppoa sisällön tuottaminen on pelkällä ydinversiolla ilman mitään lisäosia vertailuasteikon toimiessa heikosta erinomaiseen. Joomlaissa sisällön tuottaminen on erittäin helppoa. Siinä tulee vain luoda sivulle



erillinen artikkelin julkaisemiseen soveltuva sivupohja ja sitten kirjoittaa tekstieditorilla sivulle tarvittavat tekstit. Miinuksena on se, ettei ydinversion mukana tulevassa tekstieditorissa ole hirveän laajoja muokkausmahdollisuuksia, joten teksti tuotetaan yleensä erillisellä ohjelmalla tai sitten ladataan erillinen lisäosa puutteita korjaamaan. Drupalin kohdalla versiossa 7 sivujen sisällön luominen on samanlaista kuin Joomlaissa. Drupalin heikkoudeksi tulee sen ydinversion mukana tulevat liialliset osat. Olisi hyvä, jos käyttäjä itse voisi määrätä, mitä osia itselleen haluaa. Voidaankin siis sanoa, että edellinen versio 6, jota ehdittiin työn puitteissa jonkin verran käyttää, oli helpompi vasta-alkajan oppia käyttämään. Tämä selittää suluissa olevan arvosanan.

Wordpressissä kirjoitusten luonti on parhainta ja siitä näkeekin, että järjestelmä luotiin alun perin vain blogien ylläpitoa helpottamaan. Heikkoutena voidaan pitää järjestelmän juuria blogeissa eikä testattaessa heti löytynyt mahdollisuutta luoda tekstiä muualle kuin ennalta määrättyyn blogiosioon. Concrete5 on pääasiallisesti yrityspohjalta rakennettu järjestelmä, joka on ensikertalaiselle hieman vaikeampiselkoinen. Concrete5 tarvitsee suhteellisen pitkän perehtymisen, jotta löytää kaiken tarvittavan ilman vaikeuksia. Tämä aloittamisen vaikeusaste laskee muuten hyvän järjestelmän käytettävyyttä suhteellisen paljon.

CMS Made Simple on nimensä mukaisesti erittäin helppokäyttöinen. Siinä oli Joomlaan ohella kaikkein aloittelijaystävällisin käyttöliittymä, jonka avulla pääsi helposti sivuston hallitsemiseen käsiksi. Sen heikkoutena voidaan pitää sitä, että sisällön tuottamiseen käytännössä tarvitsee olla oma erillinen tekstinkäsittelyohjelma. Tämä johtuu siitä, että CMS Made Simplen oma tekstikäsittelijä on todella suppea verrattuna vaikkapa Concrete5:n vastaavaan. Etomite oli pettymys verrattuna mihin tahansa muuhun tutkimuksen alla olleeseen sisällönhallintajärjestelmään. Etomite ei ole aloittelijaystävällinen eikä oikein kunnolla sovellu sisällönluomiseen. Etomite on tarkoitettu täysin ja vain sisällönhallintaan. Kaikki muu tulee tuottaa ulkoisesti, joten ohjelmointikokemusta tarvitaan järjestelmän hyödylliseen käyttöön.

Tämän tutkimuksen perusteella pystyttiin sulkemaan pois kaksi järjestelmää, jotka olivat CMS Made Simple ja Etomite. Tämä tehtiin siksi, ettei kumpikaan täyttänyt sivustolle annettua vaatimusta suomennoksen löytymisestä. Tutkimusta jatkettiin vertailemalla järjestelmien asennustapoja (taulukko 2). Tämä tehtiin, koska haluttiin tarkistaa, ettei järjestelmän asennus olisi liian vaativaa niin sanotun tavallisen käyttäjän näkökulmasta.

Jäljelle jääneistä järjestelmistä jokaisella on asennuksen helpottamiseksi luotu oma asennusohjelmansa, joka käynnistyy, kunhan ensin on tehty taulukossa mainitut asiat valmiiksi. Ensimmäiseksi kannattaa luoda tietokanta valmiiksi, kuten taulukosta 1 näkee. Suurimmalla osalla jäljellä olevista järjestelmistä se on MySQL-kanta. Kun tietokanta on luotu, siirretään jokaisen järjestelmän kohdalla asennuksessa tarvittava paketti sellaiseen kansioon palveluntarjoajan levyllä, johon pääsee käsiksi verkkosivujen kautta. Tämän jälkeen tulee järjestelmissä, pois lukien Joomla, muokata PHP-oikeuksia, jotta asennus voidaan suorittaa läpi ilman ongelmia. Kun nämä etukäteen suoritettavat valmistelut on tehty, mennään internetselaimella osoitteeseen, joka vastaa paketin sijaintia. Tämän jälkeen on asennus jokaisen kohdalla ruudulta seurattavien ohjeiden toteuttamista.

<b>Nimi</b>	<b>Asennus</b>
Joomla	Kanta valmiiksi ja paketti oikeaan kansioon
Drupal	Kanta valmiiksi, paketti oikeaan kansioon ja PHP-oikeuksien muokkausta
Wordpress	Kanta valmiiksi, paketti oikeaan kansioon ja PHP-oikeuksien muokkausta
Concrete5	Kanta valmiiksi, paketti oikeaan kansioon ja varmistus, että mod_rewrite-toiminto on päällä

Taulukko 2. Sisällönhallintajärjestelmien asennustyyli.

Asennusten vertailujen jälkeen huomattiin, että asennustavat eivät poikkea toisistaan juuri ollenkaan. Pieniä eroja tosin löytyi, ja asennuksen vaikeuden perusteella Joomla vaikutti kaikkein helpoimmalta asentaa, koska kaikki muut järjestelmät vaativat suurempaa muokkausta asennusympäristön PHP-oikeuksiin. Joomla ei tästä kuitenkaan valintaa tehdessä hirveästi hyötynyt, koska asennustavalla ei ollut lopulta hirveän suurta merkitystä valittaessa niitä kahta sisällönhallintajärjestelmää, joita tutkittiin laajemmin asentamalla ne testiympäristöön. Kun tarkastettiin, millä kielillä jäljellä olleet sisällönhallintajärjestelmät oli tehty, päätettiin sekin jättää vähemmälle huomiolle kaikkien käyttäessä PHP- ja (X)HTML-kieliä. Valintoja tehtäessä keskitettiin siis huomio tietokannan tyyppiin ja käytettävyyteen.

Ensimmäinen valinta osui Wordpressin ja Joomlaan välille, koska molemmilla oli MySQL-kanta sekä molempien käytettävyys aloittelijan näkökulmasta oli hyvä. Tämä valinta kohdistui Joomlaan, koska Joomla oli enemmän hallinta- ja sisällöntuottojärjestelmä, kun taas Wordpress blogitaustansa mukaan oli keskittynyt enemmän sisällöntuottamiseen. Voidaankin sanoa, että Joomla oli kokonaisvaltaisampi. Joomlaan valinnan jälkeen siirryttiin miettimään toista sisällönhallintajärjestelmää pidempiaikaiseen testaukseen. Tämä valinta osui jäljellä olevista järjestelmistä Drupaliin. Tämä johtui Drupalin mahdollisuuksista käyttää eri tietokantoja eikä vain MySQL-kantaa. Tämä moninaisuus kaatoi alleen Concrete5:n ja Wordpressin hieman aloittelijaystävällisemmät vaihtoehdot, koska MySQL-kannalle haluttiin etsiä vaihtoehtoa.

## 4.2 Drupal

Drupal on PHP-kielellä ohjelmoitu sisällönhallintaohjelmisto, joka toimii erilaisissa tietokannoissa kuten MySQL ja SQLite. Sen nimi Drupal tulee hollannin kielen sanasta druppel eli pisara, joka on kuvattuna Drupalin logossa.

Tätä työtä tehdessä Drupalista julkaistiin versio 7.10, jonka minimivaatimukset toiminnalle ovat PHP 5.2.5 tai korkeampi. Tietokannaksi valittavana on monia eri vaihtoehtoja: MySQL 5.0.15 tai korkeampi, jossa tulee olla päälle kytkettynä PDO eli PHP Data Objects, SQLite 3.3.7 tai korkeampi. Näiden tietokantojen lisäksi Drupal 6.10 tukee PostgreSQL-versio 8.3 tai korkeampaa. Lisäksi on mahdollista asentaa erillisellä moduulilla tuki Microsoft SQL Serverille ja Oracelle. Näiden lisäksi kotikoneelle Drupalin asentaessa tulee asentaa web-palvelin. Drupalin tapauksessa suositeltu palvelin on Apache 1.3 tai 2.x. Toisena vaihtoehtona ovat Microsoftin IIS, josta versiot 5, 6 ja 7 toimivat Drupalin kanssa.

### 4.2.1 Historia

Drupalin kehitys lähti vuonna 2000 liikkeelle, kun Antwerpenin yliopistossa opiskellut Dries Buytaert yhdessä ystäviensä kanssa jakoi yhden adsl-modeemin. Tuohon aikaa tämä oli hyvin suurta luksusta, mutta Driesiä rupesi harmittamaan se, ettei heillä ollut mahdollisuutta viestitellä tai jakaa asioita verkon yli.

Tästä seurasi kaverien välisen viestintäjärjestelmän synty. Kaveriporukka halusi säilyttää vielä koulusta poistumisen jälkeenkin. Tästä johtuen Dries siirsi palvelun verkkoon virheellisesti kirjoittaen hollanniksi kylää tarkoittavan sanan *dorp* muotoon *drop*. Internetiin siirron jälkeen yhteisö rupesi muuttumaan, ja sen jäsenet rupesivat puhumaan uusista teknologioista. Pikkuhiljaa sivustosta tuli henkilökohtainen testiympäristö, jota ajoivat eteenpäin keskustelut ja ideoiden esille tuominen. Näitä teknologioita sitten testattiin *drop.org*:ssa uusina lisäyksinä ohjelmistoihin, jotka ylläpitivät sivustoa. [4.]

Vuonna 2001 Dries päätti julkaista *drop.org*:n takana olleen ohjelmiston ja nimesi sen *drop*:n hollanninkielisellä vastineella *druppel*. Nykyään Drupalin kehitys on keskittynyt *drupal.org*-sivustolle, joka on itsekin Drupalilla luotu sivusto. Siellä työkseen Drupalia käyttävät ja ohjelmaan omaksi harrastukseksi itseksensä tutustuneet kehittäjät osallistuvat ohjelmiston kehitykseen, sekä pyytävät apua toisiltaan ja neuvovat toisiaan.

#### 4.2.2 Ominaisuudet

Perusasennuksen jälkeen Drupal tarjoaa muutamia ominaisuuksia kuten artikkelien julkaisun ja käyttäjien hallinnan. Ohjelmiston luonne on kuitenkin sellainen, että jokainen käyttäjä voi muokata Drupalista omanlaisensa lisäosilla eli moduuleilla, joista muutamia tulee perusasennuksen mukana. Moduulit ovat niin iso osa Drupalia, että ne tulee ottaa huomioon jo suunnitteluvaiheessa. Ne löytyvät julkaistuina *Drupal.org*-sivustolta, jossa niihin voi helposti tutustua. Moduulien asennus tapahtuu lataamalla moduuli koneelle ja siirtämällä tiedostot johonkin kansioon. Tämän jälkeen moduuli otetaan käyttöön Drupalin käyttöliittymän kautta.

Perusasennuksen mukana tarjotaan oman sivuston etusivulle kirjautumis- ja rekisteröintimahdollisuus kaikille sivulle saapuville henkilöille. Mukana tulevat työkalut mahdollistavat rekisterikäytäntöjen muokkauksen, eli onko kaikkien mahdollista vain rekisteröityä käyttäjiksi vai pitääkö jonkun ensin hyväksyä rekisteröinti tai mahdolliset rekisteröityneen käyttäjän oikeudet.

### 4.2.3 Sivuston luominen Drupalin avulla

Sivuston luominen alkaa lataamalla Drupalin viimeisin versio osoitteesta <http://drupal.org/project/drupal>. Paketin latauksen jälkeen siirretään paketti palveluntarjoajan verkkolevyn HTML-kansioon, jonne se tulee purkaa. Purkamisen luo kansioon uuden kansion nimeltä Drupal-X.X, siis esimerkiksi versio 7:ää asennettaessa kansio on Drupal-7.0, tämä kansio tulee nimetä niin, että se viittaa sivuston nimeen, jota ollaan luomassa. Seuraavaksi tulee luoda sivusto sillä nimellä, mihin kansion nimessä viitattiin ja sivuston tulee viitata siihen samaiseen kansioon, joka luotiin purkamalla asennuspaketti. Tämän jälkeen kannattaa luoda Drupalia varten oma tietokanta, jolle on luotu käyttäjä, jolla on oikeudet kaikkeen. Tietokannassa käytettävänä tiedonhallintajärjestelmänä voi olla MySQL, SQLite tai aikaisemmassa versiossa PostgreSQL.

Seuraavaksi tulee muokkaila tiedostoa nimeltä default.settings.php. Se tulee nimetä default.php:ksi ja sille tulee antaa mahdollisuus kirjoittaa palvelimelle. Tämä täytyy tehdä, jotta asennuskripti saa tarvittavat oikeudet asentaa Drupalin. Asennusohjelman tulisi muokata tiedosto asennuksen valmistuttua kirjoitussuojatuksi, mutta tiedosto kannattaa silti tarkistaa asennuksen jälkeen. Kun tiedostolle on annettu tarvittavat oikeudet, tulee sivustolle mennä internetselaimella antamalla sivuston osoite. Osoitteessa käyttäjän tulee antaa tiedot siitä, minkälaisen asennuksen hän haluaa sekä määrittää sivuston kieli ja tietokannan tiedot, jonka jälkeen asennus käynnistyy. Asennuksen jälkeen sivusto on periaatteessa valmis käytettäväksi ja sitä voidaan lähteä muokkaamaan halutulla tavalla.

On kuitenkin suositeltavaa suorittaa muutama tehtävä ennen sivuston varsinaista käyttöä. Jos ei pidä osoitteista, joissa on esillä koodia, täytyy mennä Drupalin pääkäyttäjän tunnuksilla Search and metadata -tietoihin ja valita sieltä puhtaat osoitetiedot. Tämä kannattaa tehdä, jotta sivuston osoite on selvempi ja helpottaa osoitteen käsinkirjoittamista. Kannattaa myös varmistaa, että Drupal on luonut asennettaessa files-kansion. Tämä kansio on tärkeä, sillä sinne tallentuvat esimerkiksi kaikki käyttäjän kuvat, joten on tärkeää luoda kansio, jos asennus sitä ei tehnyt. Tämä on ollut ongelma versioissa ennen 7:ää mutta se kannattaa silti varmistaa. Näiden tehtävien jälkeen voidaan sivustoa lähteä kehittämään aktiivisesti.

### 4.3 Joomla!

Virallisesti Joomla! kirjoitetaan huutomerkkin kanssa, mutta kuten niin useasti kirjallisuudessa myös tässä työssä huutomerkki jätetään pois. Tätä työtä tehdessä Joomla on edennyt versioinnissa versioon 1.7.3. Tämän version rinnalla kulkee ns. LTS (Long Term Support) -versio 1.5. Uusin versio tulee olemaan seuraava LTS-versio ja sen piti alun perin olla versio 1.8, mutta viime elokuussa ilmoitettiin, että kyseinen versio tulee olemaan 2.5. Tämä muutos tehtiin, koska oli päätetty versioiden numeroinnin muutoksesta sellaisiksi, joissa jokainen LTS-versio tulee olemaan x.5-versio.

Työssä käytetyn version 1.7.x vaatimukset ovat samantapaiset kuin Drupalissa: molemmat tarvitsevat käyttöönottoa varten PHP:n, tietokannan ja web-palvelimen. PHP:n tulee olla 5.2.4 tai korkeampi, tietokannan tulee olla MySQL 5.0.4 tai korkeampi sekä web-palvelimeksi sopivat joko Apache 2.x tai Microsoft IIS 7.

Metropolia Ammattikorkeakoulussa on aikaisemmin tehty insinööriä, joka tutustuu sisällönhallintajärjestelmistä ainoastaan Joomlaan. Työ on nimeltään Joomla-sisällönhallintajärjestelmä ja sen on kirjoittanut Juho Jaakonsaari [13]. Kyseisessä työssä selvitettiin Joomla-sisällönhallintajärjestelmän toimivuutta yhdistyksen web-sivuina. Työ tehtiin Metropolia Ammattikorkeakoulun opettajien paikallisyhdistykselle, ja se on tehty keväällä 2009.

#### 4.3.1 Historia

Joomla oli aluksi yksi Mambo-nimisen sisällönhallintajärjestelmän kehityshaaroista. Kuitenkin vuonna 2005 Mambon oikeudet omistanut yritys Miro International julkaisi avoimen kirjeen aikeistaan perustaa erillinen Mambo-säätiö, jonka tavoitteena olisi varmistaa Mambon menestys ja tulevaisuus. Hyvin nopeasti alkoi ilmetä ihmettelyä säätiön vaikuttimista, kun kuudesta säätiön hallituksen jäsenestä neljä oli Miro Internationalin työntekijöitä. Tästä johtuen moni kehittäjätiimin jäsen pelkäsi, että säätiö saattaa tehdä Mambosta kaupallisen version ja lopettaa avoimen lähdekoodin Mambon.

Seurannut viikko käytiin kiivasta keskustelua siitä, onko tarkoituksena kenties kaupallistaa avoimen lähdekoodin Mambo, sillä Miro Internationalilla oli olemassa jo kaupallinen version nimeltä Mambo cms. Tätä peläten Mambon kehitystiimin johtaja Andrew Eddie

julkaisi vastineen Mambo-säätiölle, jossa kehitystiimi yksimielisesti ilmoitti olevansa huolestunut siitä, miten Miro ja Mambo-säätiö vaikuttavat kehitykseen. Kehitystiimin mielestä Mambon kehitystä tulisi ohjata käyttäjien tarpeet ja kehittäjien taidot eivätkä niinkään johtajat. Koska Mambo-säätiö oli viimekädessä suunniteltu antamaan päätäntävalta asioista Mirolle, päättivät kehittäjät jatkaa oman sisällönhallintajärjestelmänsä pariin, joka myöhemmin nimettiin Joomlaiksi. [5.]

#### 4.3.2 Ominaisuudet

Versiosta 1.5 lähtien Joomla on toteutettu MVC-mallin mukaisesti, toisin sanoen järjestelmän looginen koodi ja sisällönosuus on erotettu näkyvästä osuudesta. MVC-malli jakaa ohjelman kolmeen itsenäiseen osuuteen.

- malli (Model) huolehtii tiedon tallentamisesta, ylläpidosta ja käsittelystä
- näkymä (View) määrittää käyttöliittymälle ulkoasun ja millä tavalla mallin tiedot esitetään käyttöliittymässä
- ohjain (Controller) vastaanottaa käyttäjän käskyt ja muuttaa mallia sekä näkymää vastauksena niihin.

Tämä jako helpottaa kehittäjien työtä merkittävästi, esimerkiksi sivupohjan tekijän ei tarvitse miettiä järjestelmän loogisuutta.

Perusasennuksen mukana tulevat sivustopohjat tarjoavat luodulle sivustolle valmiiksi ns. front-end-käyttäjän rekisteröitymisen ja sisäänkirjautumisen samaan tapaan kuin Drupal. Eroa on kuitenkin ns. back-endin toteutuksessa. Sen kautta sivuston administrator eli tekninen pääkäyttäjä pääsee muokkaamaan ja asentamaan uusia komponentteja sekä mahdollisia laajennuksia, kuten vaaditun suomen kielipaketin.

Joomlan sisältö koostuu pääryhmistä (section) ja sen alaryhmistä (categories). Sisältö on tallennettuna MySQL-tietokannassa, joka käsittelee tietojen tallennuksen ja haut. Tietokanta keskustelee sisällönhallintajärjestelmän kanssa, joka sitten sivupohjan kautta luo dynaamisen web-sivuston.

Joomlan muuntokyky perustuu moniin lisäosiin ja komponentteihin. Komponenttien ansiosta järjestelmää voidaan laajentaa, tarpeiden niin vaatiessa, kustannustehokkaas-

ti. Koska Joomla on avoimen lähdekoodin ohjelma, voidaan tarvittavat lisäosat luoda myös itse. Tosin Internet on täynnä lisäosia Joomlaan, joista osa on kaupallisia, joten todennäköisesti ei oman luomiselle tule varsinaista tarvetta.

Joomlan vahvuutena on sen suosio monessa maassa, jolloin monikielisten sivujen tekeminen helpottuu tarvittavien kielipakettien löytyessä suhteellisen helposti.

#### 4.3.3 Sivuston luominen Joomlaan avulla

Joomlalla sivuston tekeminen on samanaikaisesti helppoa ja vaativaa. Sivuston voi pysyttää nopeasti ja asennuspaketin mukana tulevat sivupohjat mahdollistavat sivuston nopean käyttöönoton. Sivuston käyttöönotto käydään tarkasti läpi luvussa 5.

Käyttöönoton jälkeen alkaa varsinainen sivuston luomistyö. Joomlaan löytyy useita piilotettuja ominaisuuksia. Nämä löytyvät moduulien hallinnasta. Tällaisesta hyvänä esimerkkinä toimii työssä tarvittu kielen vaihtaja. Luomistyöstä saa helposti hyvin vaativan, jos rupeaa käymään läpi jokaisen nappulan paikan erikseen. Tämä on toisaalta hyvin suositeltavaa, jotta sivusto eroaisi muista tarpeeksi.

Jos pelkkä elementtien muokkailu ei riitä tekijälle, tuleeekin siirtyä suoraan luomaan omat tyylisivut. Tyyliohjan saa ladattua omaan sivustoonsa Joomlaan kautta helposti. Tarvittavat tiedostot luodaan omaan kansioon, ja tämä kansio sijoitetaan Joomlaan Template-kansioon. Tämän jälkeen täytyy mennä lisäosien hallintaan ja valita etsi ja asenna -vaihtoehto. Siellä tulee suorittaa etsintä, jolla sivupohja oikein asennettuna löytyy. Löydyttyä sivupohja tulee vielä asentaa, jonka jälkeen se on valmis käytettäväksi.

Jotta yllämainittu sivupohjan löytäminen onnistuu, tulee sivupohjan täyttää Joomlaan vaatimukset sivupohjalle. Näitä vaatimuksia ei ole montaa, mutta ne ovat sitäkin tärkeämmät. Ensimmäinen vaatimus on, että sivupohja rakentuu vähintään yhdestä CSS tiedostosta, index.php-tiedostosta ja templateDetails.xml-tiedostosta. Xml-tiedostossa kerrotaan sivupohjan tekijän tiedot, se mistä muista tiedostoista pohja koostuu, minkälaisia elementtejä sivupohjan käytettävissä on sekä sivupohjan nimi. Toinen vaatimus onkin, että sivupohjan nimi tulee olla täysin sama kansion nimen kanssa, jonne tiedostot on tallennettu. Index.php-tiedosto toimii sivuston kokoavana pääsivuna joten sinne



luodaan HTML- ja PHP-koodilla kaikki tarvittavat tiedot siitä, miten ja millä ohjelmointipalikoilla sivusto koostetaan. Index-tiedostoon voidaan kirjoittaa myös kaikki tyyllitavat. Jotta siitä tulee paljon selvempi sijoitetaan tyylikelementit omaan template.css-tiedostoonsa. CSS-tiedostossa sitten selvitetään, minkälaiset väripohjat sivulle halutaan ja minkälaisia elementit ovat.

#### 4.4 Sisällönhallintajärjestelmän valinta

Joomlaan ja Drupaliin tutustumisen jälkeen vertailtiin näillä sisällönhallintajärjestelmillä luotuja sivuja ja päätettiin, kumpaa sivustoa olisi helpompi käyttää. Drupalin ongelmana toimi alusta lähtien ydinversion liiallinen paisuminen. Tästä johtuen sivuston päivittäminen sillä oli aina useamman mutkan takana. Joomlaan ongelmana olivat asennuksen mukana vahingossa tulleet valmiit sivustot, joiden tutkimisesta oli toki hyötyäkin, mutta pidemmällä tähtäimellä niiden mukana tuomat kuvat ja tiedot sekoittivat sivuston luomista.

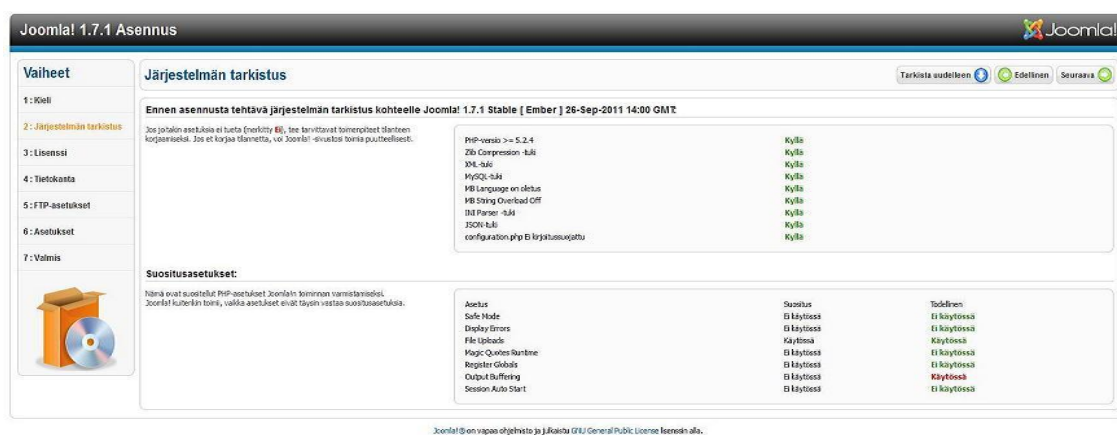
Joomla päätettiin ottaa lopulliseksi sisällönhallintajärjestelmäksi sen tarjotessa aloittelija ja ystävällisemmän sivuston hallintapuolen sekä kaikkein yksinkertaisimman tavan luoda sivustosta kaksikielinen ilman koodin muokkaamista. Koodin muokkausta haluttiin välttää, jotta asiakas pystyy itse luomaan sivustolle uutta ilman koodiasiantuntemusta. Lisäksi Joomlaan ulkoasun muokkaaminen on paljon yksinkertaisempaa kuin Drupalissa. Ulkoasu oli asiakkaalle tärkeä, jotta sivustosta tulisi tarpeeksi näyttävä ja houkutteleva.

## 5 Sisällönhallintajärjestelmän käyttöönotto

Kuten edellisessä luvussa mainittiin, päädyttiin työssä valitsemaan Joomla toteutettavaksi järjestelmäksi. Testiympäristönä toimi luvussa 2 mainittu ympäristö. Testiympäristön lisäksi Joomla asennettiin palveluntarjoajan palvelimelle. Kuitenkin kaikki testaaminen ja rakentaminen tapahtuivat testiympäristössä.

## 5.1 Joomla'n asennus

Joomla'n asennus tapahtuu lataamalla asennuspaketti vaikkapa osoitteesta [www.joomla.fi](http://www.joomla.fi), joka sitten puretaan haluttuun kohteeseen. Tästä kansioista täytyy olla yhteys internetiin, sillä Joomla'n varsinainen asennus tapahtuu purkamisen jälkeen selaimen kautta; kansio on yleensä internetpalvelimen juurihakemisto. Asennuspaketti käynnistää Joomla'n asennuksen, josta aluksi valitaan haluttu asennuskieli, jonka jälkeen ohjelma tarkistaa, löytyykö palvelimelta tarvittavat tuet Joomla'n käyttöön. Samalla annetaan suositukset PHP-asetuksiksi. Koska nämä ovat vain suosituksia, toimii Joomla, vaikkei kaikki täsmäisikään (kuva 2).



Kuva 2. Järjestelmän sopivuus palvelimelle.

Tarkistuksen jälkeen seuraavat Joomla'n lisenssitiedot. Neljäntenä vaiheena päästään määrittämään tietokanta-asetukset (kuva 3). Tässä vaiheessa MySQL-palvelimen asetukset on oltava jo luotuna, sillä tässä vaiheessa pitää valita tietokannan tyyppi, palvelimen nimi, käyttäjätunnus ja salasana palvelimelle sekä tietokannan nimi. Viimeisimpänä on taulun etuliitteen valinta, joka on viimeisimpänä ilmestynyt muutos tietokannan luomisessa. Tähän päädyttiin sen vuoksi, ettei usealta palveluntarjoajalta saa yleensä kuin yhden tietokannan käyttöönsä, ellei halua maksaa lisää toisesta tietokannasta. Tämä mahdollistaa useamman Joomla-sivuston asentamisen yhdelle tietokannalle, jolloin taulujen etuliitteen erottavat eri sivustot toisistaan ja niiden hallinnointi onnistuu eriaikaisesti.

**Perusasetukset**

Tietokannan tyyppi \*  
 MySQLi ▼ *Tämä on luultavasti "MySQLi"*

Palvelimen nimi \*  
 localhost *Tämä on yleensä "localhost"*

Käyttäjätunnus \*  
 testi *Esimerkiksi "root" tai palveluntarjoajan antama käyttäjätunnus*

Salasana  
 ●●●●●● *Sivuston tietoturvan takia mysql-tilin salasana on pakollinen*

Tietokannan nimi \*  
 testi *Osa palveluntarjoajista sallii vain tietyn tietokannan nimen sivustolle. Käytä tällöin taulun etuliitettä erottaaksesi eri Joomla! sivustojen käyttämät taulut toisistaan.*

Taulun etuliite \*  
 testi\_ *Valitse tietokannan etuliite tai käytä **satunnaisesti luotua**. Yleensä kolme tai neljä merkkiä pitkä joka sisältää vain alfanumeerisia merkkejä ja jonka TÄYTYY päättyä alaviivaan.*

Vanhin tietokannan käsittely \*  
 Tee varmuuskopio *Aikaisemman Joomla! -asennuksen luomat varmuuskopiot tauluista poistetaan*  
 Poista

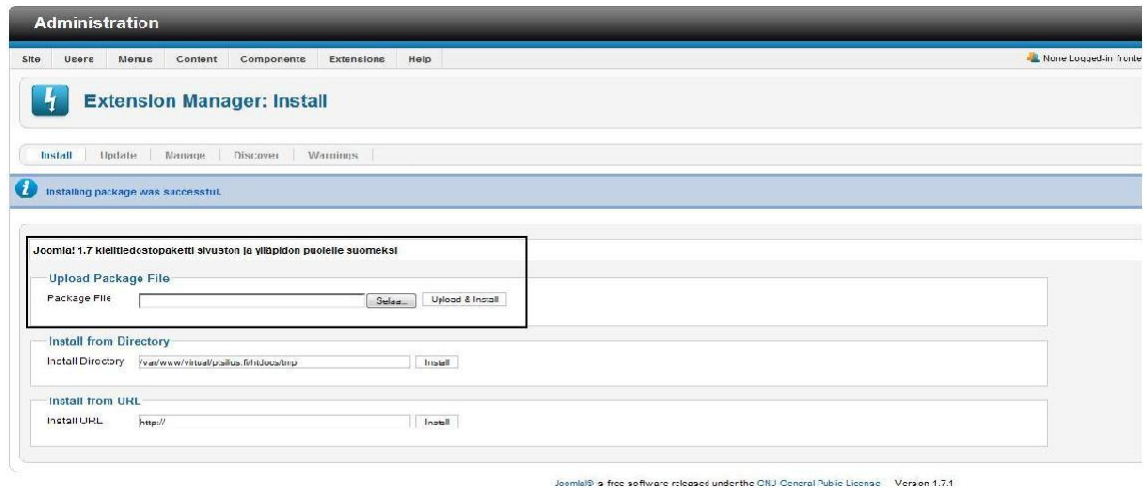
Kuva 3. Tietokantaselitykset tarkemmin.

Viidentenä vaiheena annetaan mahdollisuus luoda ftp-tunnukset sivustolle. Tämä voidaan kuitenkin suorittaa myöhemminkin Joomla:n ylläpitoliittymän (backend) kautta. Kuudennessa vaiheessa annetaan sivustolle nimi sekä määritetään pääkäyttäjän tunnus, sähköposti ja salasana. Lisäksi kysytään, haluaako käyttäjä mahdollisesti asentaa paketin mukana tulleet esimerkkiartikkelit. Artikkelit on hyvä asentaa testiympäristöön, jos käyttäjällä ei ole aikaisempaa kokemusta Joomla:sta.

Viimeisenä vaiheena tulee asennuksen loppuun poistaa palvelimelta Joomla:n asennuskansio tietoturvasyistä. Kansio löytyy juurihakemistosta, jonne asennuspaketti purettiin.

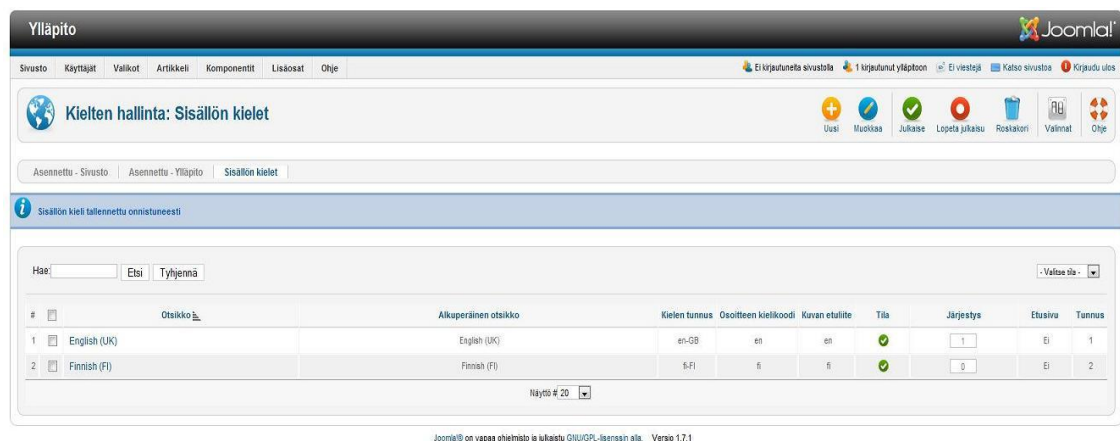
## 5.2 Sivuston luominen

Asennuksen jälkeen sivusto on käyttövalmis, mutta sivusto on vain ontto pohja, jota tulee lähteä muokkaamaan tarvittavaan suuntaan. Koska Joomlaan on tarjolla suomenkielinen paketti, kannattaa sivuston luominen aloittaa asentamalla sivustolle tämä paketti helpottamaan sen luontia. Asennus tapahtuu kirjautumalla ylläpitoliittymään ja menemällä laajennushallinta-välilehdelle. Täältä tulee ladata ja asentaa valittu paketti (kuva 4).



Kuva 4. Paketin sivustolle lataus ja asennus.

Asennuksen jälkeen suomen kieli on valmiina ylläpitojärjestelmän käytettäväksi. Kuitenkaan sitä ei vielä voida käyttää sivuston kielenä, joten suomi tulee asettaa sisällön kieleksi. Tämä tapahtuu menemällä lisäosista kielten hallintaan ja siellä tulee luoda kielen tiedot. Tehtävää helpottaa, jos katsoo esimerkin aina mukana tulevasta englannin versiosta. Luonnin jälkeen kieli ilmestyy käytettävissä olevien kielten listaan (kuva 5).



Kuva 5. Käytettävissä olevat kielet.

Koska sivustosta haluttiin tehdä kaksikielinen, tarvittiin kielen vaihtamiseen Joomla mukana tuleva kielen suodatus. Suodatus löytyy Joomla liitännäisten hallinnasta, josta se tulee aktivoida.

Aktivoimisen jälkeen siirryttiin luomaan varsinaista sivustoa kaksikieliseksi. Tämä tapahtuu aluksi menemällä valikoiden hallintaan ja sieltä vaihtamalla asennuksen mukana tullut etusivu halutulle kielelle. Tämä ei kuitenkaan suoraan onnistu vaan valikoiden hallinnassa on tehtävä uusi valikko ja sille oma valikon nimike. Tämä nimettiin työssä default-valikoksi. Valikko täytyy tehdä siksi, että Joomla:ssa täytyy olla yksi valikko, joka on tarkoitettu kaikille kielille (kuva 6) ja sen täytyy olla asetettuna aluksi etusivuksi. Tämän valikon tulee olla sitten täysin tyhjä, ja se toimii välityksenä tuleville kielille.

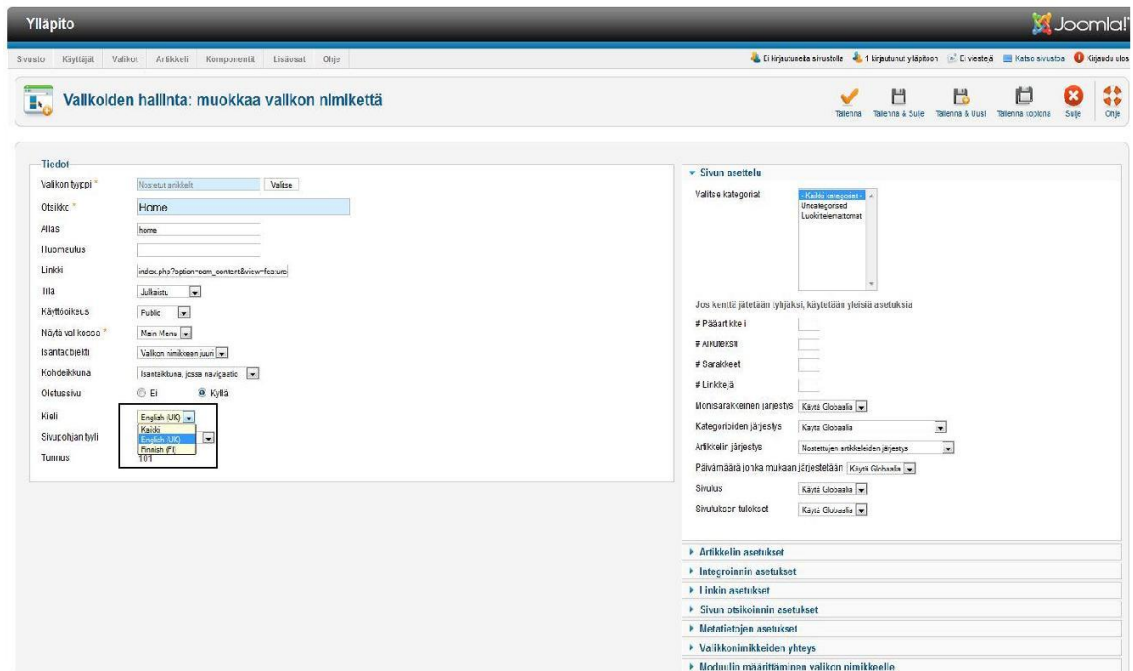
Otsikko	Valikon nimikkeiden määrä			Moduulit linkitetty valikkoon	Tunnes
	# julkaistu	# julkaisematon	# roskakori		
Main Menu (Valikon tyyppi: mainmenu)	2	0	0	• Main Menu (Public asemassa position-7)	1
default (Valikon tyyppi: default)	1	0	0		2
Päävalikko (Valikon tyyppi: päävalikko)	2	0	0	• Päävalikko (Public asemassa position-7)	3

Näytetty 4/20

Joomla!® on vapaa ohjelmisto ja julkaistu GNU/GPL-lisenssin alla. Versio 1.7.1

Kuva 6. Päävalikot luotuna.

Tyhjän valikon jälkeen voidaan joko palata aikaisemmin olemassa olevan valikon luomiseen tai tehdä uusi valikko, johon lähdetään luomaan kielen mukaisia valikon nimikkeitä. Nimike luodaan valitsemalla valikkonäkymässä uusi valikon nimike. Nimike sitten kannattaa nimetä aloitussivua kuvaavaksi. Tässä työssä nimiksi tulivat home ja etusivu. Kun nimeäminen on tapahtunut mieleiseksi, tulee valita kielivalikosta haluttu kieli, johon kyseinen sivu viittaa (kuva 7).



Kuva 7. Valikon nimikkeen asettaminen tietylle kielelle.

Kielten asetusten jälkeen kannattaa siirtyä etsimään kohteeseen sopivaa sivupohjaa, tai jos sellaista ei löydy, luomaan oma pohjansa. Pohjan luominen aivan alusta asti on työlästä ja vaatii ohjelmointiosaamista, sekä mielellään kuvankäsittelyohjelman. Tässä työssä päädyttiin käyttämään valmista vapaan ohjelmistokoodin sivupohjaa, jota hienon muokattiin tarpeen mukaiseksi. Tähän ratkaisuun päädyttiin sivupohjan luomisen vaatiman kuvankäsittelyohjelman puutteen vuoksi. Ratkaisuun vaikutti myös projektin aikataulu, koska sivusto haluttiin käyttöön mahdollisimman nopeasti ja sivupohjan luominen tyhjästä olisi kestänyt kauemmin. Asiakkaalle näytettiin muutamia vaihtoehtoja pohjista, joista sitten päädyttiin [www.siteground.com](http://www.siteground.com)-sivustolta löytyvään world wide -vaihtoehtoon sen selvyuden vuoksi.

Sivupohjan valinnan jälkeen world wide -pohjaa muokattiin asiakkaan pyytämällä tavalla: fonttikokoa nostettiin, jotta tekstistä tulisi näkyvämpää sekä poistettiin pohjan mukana tullut hakupalkki, koska sivustolla ei vielä sille käyttöä ollut. Edellä luvussa 3 käytiin läpi, mitä kaikkea Joomla:n sivupohja vaatii toimiakseen. Kun pohja on saatu valmiiksi, tarvitsee sivupohja vielä ladata sivustolle. Tämä tapahtuu menemällä laajennuksen hallintasivulle ja asentamalla valmiin paketin, kuten kielipakettiakin asentaessa (kuva 4). Toinen vaihtoehto on asentaa haluttu sivupohja suoraan halutusta kansiosta. Tämä tulee yleensä kyseeseen, kun kyseessä on itse tehdyn sivupohjan asentaminen.

Sivupohjan lisäksi sivustolle asennettiin sivuston ylläpitoa helpottamaan JCE (Joomla Content Editor). JCE on ilmainen tekstieditori, joka helpottaa sivuston ylläpitoa, koska se tarjoaa monipuolisemmat mahdollisuudet muokata käsittelyssä olevaa tekstiä ilman erillistä ohjelmaa. Tämä on etu varsinkin siinä tapauksessa, että sattuisi tulemaan tarve muokata tekstiä nopeasti eikä aikaa ole muokata sitä erillisessä editorissa. JCE:hen löytyy myös monia lisäosia, jotka edelleen helpottavat tekstieditorilla työskentelyä mutta ne ovat maksullisia, eli vain keskeinen osa on ilmainen. Nämä eivät todellakaan ole tarpeellisia, vaan jo pelkkä JCE:n lisäys helpottaa työtä paljon.

Asentaminen tapahtuu taas kerran laajennuksen hallinnasta ja sieltä valitsemalla oikean paketin. Parhaiten paketin oikein asennuksen voi tarkistaa menemällä komponentti valikkoon ja sieltä tarkistaa näkykö valikossa JCE administration-valintamahdollisuus. Jos tämä löytyy, on paketti toimintakunnossa. Kun paketti on asennettu, täytyy mennä sivusto-valikosta sivuston asetuksiin ja sieltä valita oletustekstieditoriksi JCE.

## 6 Toteutettu palvelu

Valmis sivusto löytyy osoitteesta [www.ptsilius.fi](http://www.ptsilius.fi). Sivusto aukeaa selaimen oletuskielen mukaan joko suomeksi tai englanniksi (kuva 8).



Kuva 8. PTSILIUS-asiantuntijayrityksen suomalainen tervetuloasivu.

## 6.1 Sivuston rakenne

Sivustolla pääsee vaihtamaan kieltä painamalla halutun kielen lippua, jolloin sivusto vaihtuu toisen kielen tervetuloasivulle. Tähän tervetuloasivulle ohjaukseen päädyttiin siksi, että sivulla vierailijat näkevät aina, mistä sivustolla on kyse vaihtaessaan kieltä. Vasemmalla sijaitsevasta päävalikosta voidaan sitten valita usein kysyttyä, johon PTSILIUS-sivuston ylläpitäjä päivittää asiakkailta mahdollisesti tulevia kysymyksiä liittyen yritykseen. Näitä kysymyksiä voidaan osoittaa yritykselle yhteydenottolomakkeen kautta. Tähän lomakkeeseen tulee antaa seuraavat tiedot itsestään: nimi, mielellään toimiva sähköposti, jotta kysymyksen esittäjään saadaan suoraan yhteys, aihealue sekä viimeisenä kuvailla ongelma tai osoittaa yritykselle kysymys. Usein kysyttyjä kysymyksiä voidaan pitää myös mainoksena sille, miten hyvin asiantuntijayritys osaa asiansa kysytyltä osa-alueelta.

## 6.2 Asiakaskontaktien hallintatoiminnallisuus

Sivustosta tuli hieman yksinkertainen, mutta se ajaa tarkoituksensa. Tämä tarkoituksellisuus on helpottaa yrityksen ja asiakkaan yhteydenpitoa, joka aikaisemmin oli vain sähköpostien varassa. Sähköposti on toki vielä suuressa osassa yhteydenpitoa ja niin sen tuleekin olla, mutta nyt mahdollinen asiakas saa tietoa yrityksen osaamisesta kysymyssivulta. Lisäksi yhteydenottokynnys madaltuu suuresti, kun kysymystä miettivän asiakkaan ei tarvitse avata erillistä sähköpostia vaan vain kirjoittaa mietteensä lomakkeeseen. Yhteydenottolomake myös vähentää paljon niin sanottuja spam-hyökkäyksiä, joissa sivustoilta etsitään sähköpostiosoite, johon sitten suunnataan suuret määrät viiruksia, haittaohjelmia ja turhia mainoksia sisältäviä sähköposteja.

Sivusto ei ole kuitenkaan aivan viimeisessä muotissaan vaan sivuston jatkokehitys on meneillään ja sitä jatketaan yhdessä PTSILIUS-yrityksen kanssa. Suurimpana kehityskohteena toimii haun luominen sivustolle. Haun luominen helpottaa sivustolle tulevan asiakkaan asian etsimistä, jos sivustoa kasvatetaan tarpeen tullen. Kehityksessä on myös usein kysyttyjen kysymysten muokkaaminen niin, että kysymykset jaetaan aihealueisiin, mikä myös helpottaa mahdollisen asiakkaan tiedon etsintää. Aikaa vievin kehitys on todennäköisesti sivustolle oman sivupohjan teko, jossa käytettäisiin hyväksi PTSILIUS-yrityksen omia kuvia, joita ei tähän mennessä ole käytetty. Kuvat eivät vain sopineet yhteen väliaikaisena ratkaisuna käyttöön otetun valmiin sivupohjan kanssa.



## 7 Yhteenveto

Verkkosivustojen yleistyessä pienyritysten täytyy pystyä erottumaan massasta edukseen, eivätkä pienyrittäjät usko saavansa verkkosivuistaan tarpeeksi hienoja. Tässä on otettu avuksi monesti maksulliset palvelut, jotka käyttävät hyväkseen markkinoille vapaan ohjelmakoodin alla julkaistuja sisällönhallintajärjestelmiä. Onkin tärkeää, että pienyritykset tutustuisivat markkinoilla oleviin ilmaisiin sisällönhallintajärjestelmiin. Tämä säästäisi paljon rahaa muuhun, sillä sivujen teko ei loppujen lopuksi ole hirveän vaikeaa.

Työn tavoitteena oli luoda PTSILIUS-yritykselle uudet verkkosivut, joiden tarkoituksena oli olla vähintään kaksikieliset ja niistä täytyi löytyä yhteydenottolomake, jottei yrityksen sähköpostiosoitetta saisi niin vain haltuunsa. Tämän lisäksi työn tavoitteena oli tutkia sisällönhallintajärjestelmiä, jotta saataisiin selville, miten ne voisivat helpottaa yrityksen tekemää sivuston päivitystä.

Työssä tutkittiin kuutta eri järjestelmää osoitteesta <http://php.opensourcecms.com/> löytyvinä valmisasennuksina eli järjestelmien demoina. Näistä kaksi asennettiin testiympäristöön ja tutkittiin niiden soveltuvuutta yrityksen uusiksi sivuiksi. Yrityksen sisällönhallintajärjestelmäksi valikoitunut Joomla osoittautui hyväksi muokattavuutensa takia. Laaja levinneisyys takaa useita vapaan ohjelmakoodin lisäosia, joita voi sitten hyödyntää sivuston jatkokehityksessä.

Tavoitteena ollut kaksikielinen helposti hallinnoitavissa oleva sivusto saatiin luotua. Helppo hallinnointi toteutui tutkimalla eri sisällönhallintajärjestelmiä, jotka mahdollistavat sivuston ylläpidon ilman koodin tuntemusta. Näistä tutkituista sisällönhallintajärjestelmistä parhaiten työhön soveltuvaksi paljastui Joomla-sisällönhallintajärjestelmä. Kaksikielisyyden mahdollisti Joomla-sisällönhallintajärjestelmän suuri suosio ja levinneisyys, mistä johtuen Joomla on käännetty erittäin monelle kielelle, joihin suomi kuuluu. Joomlaa käytettiin hyväksi myös yhteydenottomahdollisuuden luomisessa niin, ettei sivustolta löydy mistään erikseen kirjoitettuna yhteydenotto-osoitetta. Tämä tapahtui luomalla Joomlaan oma yhteydenottolomake, joka lähettää järjestelmän hallintapuolella mainittuun sähköpostiin kyseiset yhteydenotot. Usein kysytyt kysymykset -sivu toteutettiin niin, että yrittäjä päivittää sivuja sen mukaan, miten ja mitä kysymyksiä yritykselle

mahdollisesti alkaa ilmestyä. Tavoitteet saavutettiin kohtuullisesti ja kuten edellä luvussa 6 mainittiin, sivuston kehitystyö jatkuu tämän työn valmiiksi saamisen jälkeenkin.

Toisena tavoitteena ollut sisällönhallintajärjestelmien vertailu onnistui hyvin. Yritykselle löydettiin sivuston ylläpitämiseen hyvin soveltuva järjestelmä. Vertailun pääkohtina toimineet tiedot siitä, onko järjestelmällä suomenkielinen käännös ja minkälaiset tietokantavaihtoehdot järjestelmältä löytyvät, auttoivat päättämään, mitä kahta järjestelmää tutkitaan tarkemmin. Tarkemman tarkastelun alle päätyivät Joomla ja Drupal. Näistä Joomla paljastui paljon Drupalia käyttäjäystävällisemmäksi. Käyttäjäystävällisyyden huomaa parhaiten siitä, ettei Joomlaan ydinversion mukana tullut mitään tarpeetonta. Kuitenkin tarvittaessa siihen on tarjolla kaikenlaisia lisäosia. Joomla täyttääkin erittäin hyvin tavoitteen löytää yritykselle sisällönhallintajärjestelmä, joka soveltuu tarpeeseen hyvin. Työn tapauksessa tarpeena oli sisällönhallintajärjestelmä, joka tarjoaa helpon tavan hallinnoida sivustoa ilman ohjelmointiosaamista.

## Lähteet

1. Wikipedia. Sisällönhallintajärjestelmä. Verkkodokumentti. <http://fi.wikipedia.org/wiki/Sis%C3%A4ll%C3%B6nhallintaj%C3%A4rjestelm%C3%A4>. Luettu 10.10.2011.
2. Kolehmainen, Kauko. 2006. PHP & MySQL Teoriasta käytäntöön. Readme.fi.
3. Samela, Juha. 2002. Verkkosisällön hallinta. IT Press.
4. Drupal. About Drupal History. Verkkodokumentti. <http://drupal.org/about/history>. Luettu 13.11.2011
5. Joomlaportal. Mambon-kehittäjätiimin ja Miron tiet eroavat. Verkkodokumentti. <http://www.joomlaportal.fi/content/view/45/2/>. Luettu 14.11.2011.
6. Oliver, Dick. 2001. HTML & XHTML Trainer Kit. IT Press.
7. Ek, Jesper. 1999. Dynaaminen HTML käytännössä. Schildts Kustannus Oy – Pagina.
8. Monty Says. Help saving MySQL. Verkkodokumentti. <http://monty-says.blogspot.com/2009/12/help-saving-mysql.html>. Luettu 18.3.2012.
9. Wikipedia. MySQL AB. Verkkodokumentti. [http://en.wikipedia.org/wiki/MySQL\\_AB](http://en.wikipedia.org/wiki/MySQL_AB). Luettu 18.3.2012.
10. Korpela, Jukka K. 2008. CSS verkkosivujen muotoilussa. WSOYpro OY Docento-tuotteet.
11. Korpela, Jukka K. 2011. HTML5 uudet ominaisuudet. WSOYpro OY Docento-tuotteet.
12. PHP.net. History of PHP. Verkkodokumentti. <http://fi.php.net/manual/en/history.php>. Luettu 18.3.2012.
13. Jaakonsaari, Juho. Joomla-Sisällönhallintajärjestelmä. Verkkodokumentti. [https://publications.theseus.fi/bitstream/handle/10024/6958/jaakonsaari\\_juho.pdf?sequence=1](https://publications.theseus.fi/bitstream/handle/10024/6958/jaakonsaari_juho.pdf?sequence=1). Luettu 30.4.2012.