

Antti Hämäläinen

# MAPSERVER FOR WINDOWS

Opinnäytetyö  
Tietotekniikan koulutusohjelma


Toukokuu 2011




**MIKKELIN AMMATTIKORKEAKOULU**

Mikkeli University of Applied Sciences

## KUVAILULEHTI

 <b>MIKKELIN AMMATTIKORKEAKOULU</b> <small>Mikkeli University of Applied Sciences</small>	<b>Opinnäytetyön päivämäärä</b>  16.5.2011		
<b>Tekijä(t)</b> Antti Hämäläinen	<b>Koulutusohjelma ja suuntautuminen</b> Tietotekniikan koulutusohjelma		
<b>Nimeke</b> MapServer for Windows			
<b>Tiivistelmä</b>  Tämän opinnäytetyön päämääränä oli saattaa toimintakuntoon MapServer-karttapalvelinohjelmiston Windows-versio ja havainnollistaa sen toimintaa ja käyttöä spatiaalisen datan esittämiseen Internetissä.  Työssä käydään läpi paikkatiedon ja paikkatietojärjestelmien määritelmiä ja perusteita. Näihin perustietoihin kuuluvat myös oleellimmat Open Geospatial Consortiumin alalle luomat standardit, karttadatan eri formaatit rasterikuvista tietokantoihin sekä mahdolliset paikkatiedon lähteet paikkatietojärjestelmän tai -sovelluksen käytettäväksi.  MapServer for Windows asennettiin kohdekoneeseen ja karttapalvelinohjelmiston komponentit sekä hakemisto- ja tiedostorakenne kuvattiin. MapServerin ohjelmointirajapinnan testaamiseksi laadittiin esimerkksiovellus käyttäen MapServerin CGI-rajapintaa. Sovellus käännettiin myös PHP-kielelle MapScriptin kokeilua varten. Lisäksi tutkittiin, mitä valmiita sovelluksia on jo olemassa, ja yksi niistä asennettiin lyhyttä kokeilua varten.  Lopuksi arvioidaan asetettujen tavoitteiden saavuttamista ja pohditaan sekä MapServerin potentiaalia paikkatiedon esittämiseen, että paikkatiedon tulevaisuutta.			
<b>Asiasanat (avainsanat)</b> Paikkatiedot, kartat, MapServer, GIS, mapscript			
<b>Sivumäärä</b> 46	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 50%;"><b>Kieli</b> Suomi</td> <td style="width: 50%;"><b>URN</b></td> </tr> </table>	<b>Kieli</b> Suomi	<b>URN</b>
<b>Kieli</b> Suomi	<b>URN</b>		
<b>Huomautus (huomautukset liitteistä)</b>			
<b>Ohjaavan opettajan nimi</b> Jukka Selin	<b>Opinnäytetyön toimeksiantaja</b> Mikkelin ammattikorkeakoulu		

## DESCRIPTION

 <b>MIKKELIN AMMATTIKORKEAKOULU</b> Mikkeli University of Applied Sciences		<b>Date of the bachelor's thesis</b>  16.5.2011
<b>Author(s)</b> Antti Hämäläinen	<b>Degree programme and option</b> Information technology	
<b>Name of the bachelor's thesis</b> MapServer for Windows		
<b>Abstract</b>  <p>The objective of this bachelor's thesis was to put the Windows version of MapServer to a working condition and demonstrate how to use it. MapServer is an open source platform for publishing spatial data and mapping applications to the web.</p> <p>Definitions and basics of geospatial data and geographic(al) information systems were introduced in this thesis. These basics also covered the most essential standards by Open Geospatial Consortium, different formats of geospatial data, ranging from raster images to databases, and potential sources of geospatial data to be used by geographic(al) information systems or mapping applications.</p> <p>MapServer for Windows was installed on a workstation and the components and directory structure of the software were described. A sample application was created to test MapServer's Common Gateway Interface (CGI). The sample application was then converted to use PHP language to try out MapScript. The already available applications for MapServer were listed, and one of them was installed for a brief test.</p> <p>As a conclusion, achieving the goals set was evaluated. In addition, MapServer's potential and the future of geospatial information were contemplated.</p>		
<b>Subject headings, (keywords)</b> Geospatial data, map, MapServer, GIS, mapscript		
<b>Pages</b> 46	<b>Language</b> Finnish	<b>URN</b>
<b>Remarks, notes on appendices</b>		
<b>Tutor</b> Jukka Selin	<b>Bachelor's thesis assigned by</b> Mikkeli University of applied sciences	

## SISÄLTÖ

1	JOHDANTO.....	1
2	PAIKKATIETO.....	3
3	PAIKKATIETOJÄRJESTELMÄT.....	6
3.1	Kaupalliset ohjelmistot.....	9
3.2	Ilmaisia ohjelmistoja.....	10
3.3	Standardeja.....	10
3.4	Karttadata.....	12
3.5	Tiedostoformaattit.....	13
4	MAPSERVER FOR WINDOWS.....	15
4.1	MapServer for Windowsin asentaminen.....	17
4.2	Palvelinohjelmiston komponentit.....	19
4.3	Tiedosto- ja hakemistorakenne.....	21
4.4	Ohjelmointiesimerkkejä.....	23
4.5	Valmiita sovelluksia.....	37
5	POHDINTA.....	41
	LÄHTEET.....	43
LIITTEET		
LIITE 1	CGI-esimerkin käynnistävä HTML-lomake	
LIITE 2	CGI-esimerkin template-tiedosto	
LIITE 3	PHP MapScript -esimerkin koodi	

## 1 JOHDANTO

Perinteinen kartta on monin tavoin todella hyödyllinen väline, mutta sitä pidetään vaikeasti laadittavana ja päivitettävänä, staattisena ja rahallisesti kalliina. Kartta on silti korvaamaton havainnollistava työkalu, ja nykypäivänä sen esitystapa onkin käytännön syistä siirtynyt paperilta sähköiseen online-maailmaan mahdollistaen jatkuvan päivittyvyyden ja uudenlaiset toiminnot. Tämä muutos on kyennyt varsin tehokkaasti korjaamaan suurimmat perinteisten karttojen ongelmat.

Tänä päivänä harva voi enää sanoa, ettei ole käyttänyt tietotekniikan kehityksen mahdollistamia paikkatietosovelluksia. Erilaiset elektroniset paikkatietojärjestelmät ja -sovellukset ovat yleistymässä kovaa vauhtia ihmisten elämässä. Yhä useampien asioiden yhteydessä hyödynnetään paikkatietoa, joka havainnollistetaan graafisesti karttaformaatin avulla. Yleisimmin tämä jokapäiväistynyt paikkatiedon hyödyntäminen näkyy erilaisten internetin karttapalveluiden käyttönä, joilla ihmiset hakevat sekä yritysten sijaintitietoja että ajo- ja kulkureittejä ohjeistuksineen muutamalla klikkauksella. Yhä useammin tämä tapahtuu jonkin kannettavan informaatiovälineen avulla, oli se sitten älypuhelin tai tablet-tietokone. Laajemmassa yhteydessä elektronisen paikkatiedon käyttämiseen voidaan myös laskea kuuluvaksi GPS- ja navigaattorisovellukset laitteistoineen.

Oma lukunsa elektronisen paikkatiedon käytössä on ammattimainen paikkatiedon hyödyntäminen. Tällöin on usein käytössä dataa, jota ei yleensä ole saatavilla julkisissa palveluissa. Tähän voi kuulua esimerkiksi yksityisen yrityksen tilaamien markkinointitutkimusten selville saama informaatio, jossa kerätty data on mahdollisuuksien mukaan linkitetty sijaintitietoon. Näissä tapauksissa järjestelmiä käytetään vaikkapa kaupallisten palvelujen ja infrastruktuurin sijoittamiseen menestyksen kannalta edullisimmalle paikalle. Paikkatietojärjestelmää tai -sovellusta käytetään siis päätöksenteon ja analyysin apuvälineenä. Vaikka käyttötarkoitus onkin tässä tapauksessa varsin erilainen verrattuna yksinkertaisten sijaintitietojen tai reittien hakuun, on pohjalla kuitenkin varsin samantapainen tietotekninen järjestelmä ja idea.

Tämän opinnäytetyön tarkoituksena on perehtyä päällisin puolin paikkatietoon, paikkatietojärjestelmiin ja niiden käyttötarkoituksiin ensin yleisellä tasolla, ja sitten pureutua hieman syvemmälle spatiaalisen datan esittämiseen nykyaikaisilla tietoteknisillä menetelmillä. Erityisesti painopiste on ilmaisissa palveluissa ja ohjelmistoissa työn toimeksiantajan, Mikkelin ammattikorkeakoulun, vaatimuksesta. Tärkein päämääräni on saattaa toimintakuntoon ilmainen, avoimeen lähdekoodiin perustuva MapServer for Windows -karttapalvelinohjelmisto ja havainnollistaa sen toimintaa ja käyttöä.

Aluksi esittelen, mitä paikkatiedolla tarkoitetaan käytännössä ja millaisiin kysymyksiin paikkatiedon avulla on mahdollista saada vastauksia, kun sopivat analysointityökalut ovat käytettävissä. Tämän jälkeen esittelen lyhyesti joitakin saatavilla olevia merkittäviä paikkatietojärjestelmiä ja -ohjelmistoja niin kaupalliselta puolelta kuin avoimen lähdekoodin maailmasta. Aion myös selventää muutamia alalle laadittuja standardeja ja spatiaalisen datan tiedostomuotoja sekä eri vaihtoehtoja karttadatan hankkimiseksi paikkatietojärjestelmän käyttöön.

Tämän lisäksi tutkin MapServer for Windowsin käyttöä yksinkertaisilla esimerkeillä, joilla käytän ohjelmistoa renderöimään karttakuvia vapaasti saatavilla olevan spatiaalisen datan pohjalta. Lisään mukaan myös muutamia kontrolleja, joilla käyttäjä voi hallita karttakuvan näyttämistä selaimessa. Toteutan esimerkin myös MapServerille ominaisella MapScriptillä käyttäen PHP-ohjelmointikieltä. Lisäksi asennan valmiin MapServeriä varten laaditun GeoMOOSE-ohjelmiston ja esittelen lyhyesti sen ominaisuuksia. Toteutan myös yksinkertaisen Internet-sivulle upotetun kartan kustomoiduin merkinnöin käyttäen OpenLayers-javascript-kirjastoa, jota hyödynnetään myös GeoMOOSE-ohjelmistossa. Lopuksi esittelen työni tuloksia, johtopäätöksiäni ja omia pohdintojani paikkatietojärjestelmistä ja niiden käyttötarkoituksista niin nykyisessä olomuodossaan kuin myös tulevaisuudessa.

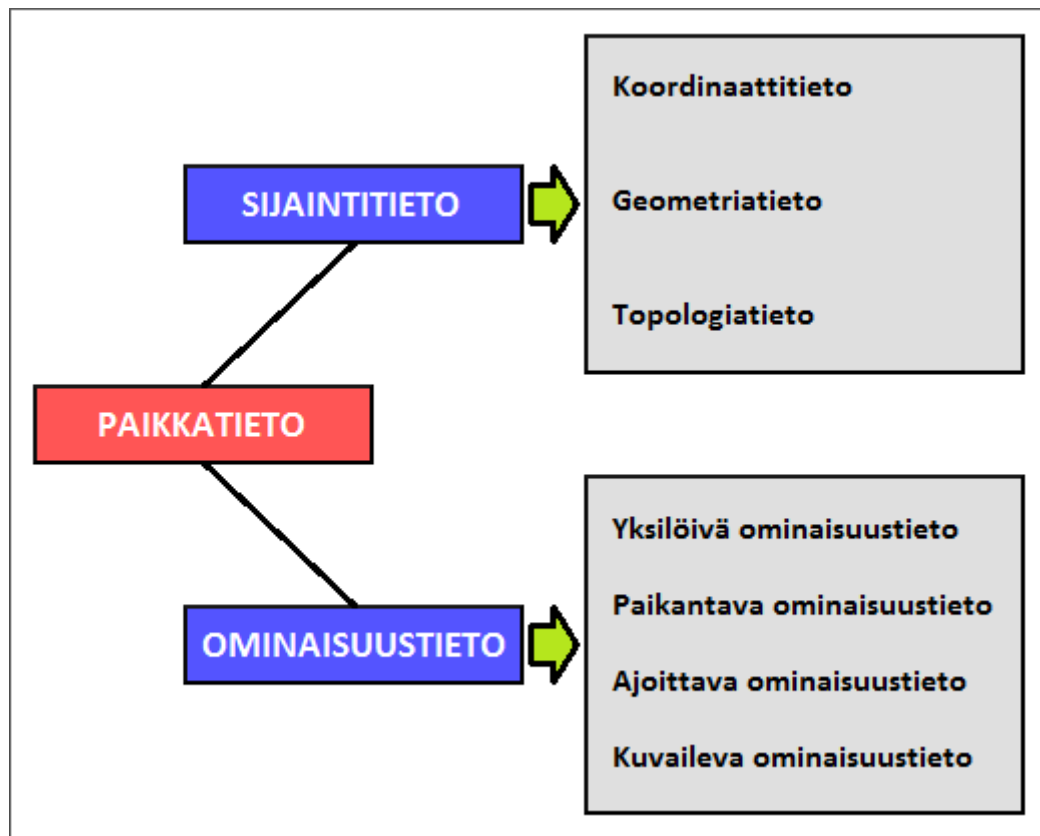
## 2 PAIKKATIETO

Mitä on paikkatieto? Ensimmäisenä mieleen tulevan vaihtoehdon perusteella sen voisi uskoa tarkoittavan yksinkertaisesti jonkin asian, esineen tai rakennelman sijaintia. Toki edellä mainittu määritelmä on osa paikkatietoa, mutta varsinaisesti paikkatiedon voidaan katsoa koostuvan kahdesta suuremmasta kokonaisuudesta, joiden alle mahtuu useita pienempiä osa-alueita. Paikkatiedolle on annettu erilaisia määritelmiä lähteestä riippuen:

- tieto kohteista, joiden sijainti Maan suhteen tunnetaan [1, s.22].
- paikannettua kohdetta kuvaavan sijaintitiedon ja kohteen ominaisuuksia kuvaavien tietojen muodostama kokonaisuus [2, s.12].

Todellisuudessa siis paikkatieto koostuu sijaintitietoon yhdistetystä kohteen ominaisuustiedosta, jolla kohde määritellään tarkemmin jollakin vapaavalintaisella ja hyödylliseksi katsotulla tavalla. Vaikka sijaintitieto on hyödyllistä ja käyttökelpoista sellaisenaan monessa yhteydessä, on siihen liitetty kohteen ominaisuustieto nimenomaan se oleellinen tekijä, joka muodostaa yhdessä sijaintitiedon kanssa paikkatiedoksi kutsutun kokonaisuuden ja määrittää paikkatiedon käyttötarkoituksen. Esimerkiksi asunnonvälityksessä on oleellista tuntea asunnon sijainti ja lähiympäristö, mutta vähintään yhtä tärkeää on tietää esimerkiksi kiinnostavan kohteen koko, kuntoluokitus, rakennusvuosi ja hinta.

Molemmat paikkatiedon pääosat, sekä sijainti- että ominaisuustieto, voidaan edelleen jakaa useampiin osiin. Seuraavassa havainnollistava kaavio (kuva 1), joka kuvaa paikkatiedon osakokonaisuuksien keskinäiset suhteet:



**KUVA 1. Paikkatiedon rakenne [3, s.9; 4, s.4. Muokattu]**

Yllä olevan kuvan mukainen tietorakenne on mahdollista siirtää sähköiseen muotoon tietokannaksi tai tiedostorakenteeksi. Näitä voidaan taas käyttää paikkatietojärjestelmässä tai -sovelluksessa karttakuvien muodostamiseen ja näyttämiseen. Se, minkä tyyppisiä tietoja tarkalleen tullaan sijoittamaan sijaintitiedon ja ominaisuustiedon alle, riippuu järjestelmän käyttötarkoituksesta, joten yllä olevaa kaavioita on pidettävä siltä osin suuntaa antavana.

Paikkatieto esitetään graafisesti kartan muodossa. Siitä käyvät ilmi kohteiden sijainnit (koordinaattitieto), muodot (geometriatieto) ja niiden suhteet toisiinsa (topologiatieto), sekä kohteiden tyyppi (ominaisuustieto: onko kyseessä rakennus, katu, rautatie jne.). Paikkatietoa voidaan esittää myös yksinkertaisesti tekstinä, esimerkiksi Itellan lähetystenseurannan voi sanoa olevan ominaisuuksiltaan karsittu paikkatietosovellus, joka ilmoittaa asiakkaalle hänen paketinsä sijainnin kyseisellä hetkellä kohtuullisella tarkkuudella sekä muutamia asiakkaalle mahdollisesti olennaisia ominaisuustietoja (paketin koko ja paino).



Sijaintitieto on jo itsessään hyödyllistä, mutta kun siihen yhdistetään omia tarkoituksellisia ajavia sopivasti valittuja kohteiden ominaisuustietoja, saadaan varsinaista paikkatietoa, jota voidaan analysoida ja käyttää päätöksenteon pohjana. Tämä tiedon analysointi on ehdottomasti oleellisimpia asioita paikkatiedon käytössä. Äskeisessä postipakettiesimerkissäni asiakas voi lähetyssurannan tietojen perusteella tehdä päätöksen paketin hakemisen ajankohdasta, sekä valita, lähtekö autolla vai saisiko paketin kuljetettua polkupyörällä.

Paikkatiedon käyttötarkoitukset vaihtelevat suuresti. Yksinkertaisimmillaan halutaan vain tietää jonkin tietyn kohteen sijainti. Eräs yleinen hyödyntämistapa on käyttää paikkatietojärjestelmää tai -sovellusta vastaamaan kysymykseen: mikä on lyhin reitti päämääräni? Hieman tätä pidemmälle vietyä erikoistunutta tietoa (ihmisten ikä, mielipide jostain tietystä asiasta) keräämällä ja liittämällä se ominaisuustiedoksi sijaintitiedon yhteyteen saadaan paikkatietoa, jolla on ominaisuustietojensa mukainen käyttötarkoitus. Tällöin voidaan paikkatietojärjestelmää varten luodulla sopivalla sovelluksella havainnollistaa todella laaja kirjo asioita.

Paikkatietojärjestelmä tai -sovellus voidaan asettaa piirtämään paikkatiedon pohjalta graafinen esitys, jossa erilaiset ominaisuudet on asetettu omille karttakerroksilleen väreillä ja kuvioilla koodattuna. Analysoimalla saatua graafista tulosta voidaan esimerkiksi havaita jonkin palvelun suhteen katvealue kohtuullisen lähellä suurehkoa väestökeskittymää. Järjestelmää voidaan siis käyttää apuna analyysissä etsittäessä vastauksia monimutkaisempiin kysymyksiin, kuten [5, s.59]:

- Millaisia kohteet ovat suhteessa toisiinsa? Miten tonttien hinta muuttuu kuljettaessa kaupungin keskustasta pois päin?
- Mikä on jonkin ominaisuuden arvo paikassa X? Mikä on väestön keski-ikä valitulla alueella?
- Mitä on tietyt ominaisuudet omaavien kohteiden vieressä? Kuinka paljon asukkaita on kävelymatkan päässä kaupunkijun myymälöistä? Miten palvelut ovat sijoittuneet kaupungissa?

”Paikkatiedon ammattilaiset ja puoli ammattilaiset ovat jo oppineet ja tottuneet käyttämään verkon kartta- ja paikkatietopalveluja. Yksi selailee kaavoja, toinen muinaismuistoja, kolmas maaperätietoja - kuka mistäkin syystä. Käyttäjä voi valita verkosta ne karttatason, jotka tuntuvat tehtävän kannalta oleellisilta ja tehdä kyselyjä tietokantoihin. Yksi kyselee vanhoja rakennuksia, toinen tyhjiä tontteja, kolmas toimipaikkoja. Käyttäjä voi tallentaa ja jakaa löytämänsä. Yksi tallentaa karttanäkymän itselleen, toinen lähettää karttalinkin kumppanille, kolmas tulostaa kartan paperille. Tämä on ollut arkipäivän sähköistä asiointia jo jonkin aikaa, ainakin maksaville asiakkaille.” [6.]

### **3 PAIKKATIETOJÄRJESTELMÄT**

Paikkatietojärjestelmä on suomen kielen vastine englannin kielen sanoille Geographic(al) Information System, lyhennettynä GIS. Paikkatietojärjestelmiin kuuluviksi voidaan laskea kaikki sellaiset järjestelmät, joiden avulla voidaan tuottaa, tallentaa, hallita, analysoida ja/tai esittää paikkatietoa. Paikkatietojärjestelmä koostuu laitteistoista, ohjelmistoista, paikkatietoaineistoista, käyttäjistä ja käytänteistä. Järjestelmät ovat siis kartoituksen, tilastollisen analyysin ja tietokantateknologian yhdistelmä. Paikkatietojärjestelmät ovat hyvin monipuolisia käyttötarkoituksissaan, niitä voidaan soveltaa muun muassa kartoitukseen, arkeologiaan, luonnonresurssien hallintaan, kaupunkisuunnitteluun, liiketoiminnan suunnitteluun, navigoimiseen, ympäristönsuojeluun ja hakukonetulosten paikallistamiseen. [7.]

Täydellinen paikkatietojärjestelmä yleensä koostuu useasta ohjelmistosta. Näihin ohjelmistokokonaisuuden muodostaviin osiin yleensä kuuluu mm. kerätyn karttadatan syöttö ja manipulointi, sijainti- ja ominaisuustiedon yhdistäminen ja niiden päivittäminen, tallennetun paikkatiedon tulostaminen graafisesti halutulla tavalla, sekä paikkatiedon analysointityökalut graafisen esityksen tulkitsemiseksi. Paikkatietojärjestelmät yleensä kootaan sen mukaisesti, millä tavalla sitä käyttävä aikoo hyötyä järjestelmästä. Erilaisia ratkaisuja painotuksineen on siis lukuisia.

Yleisesti ottaen paikkatietojärjestelmät voidaan jakaa luokkiin keskitetyimmistä ratkaisusta hajautetumpaan seuraavasti [8, s. 6.]:

- Mainframe GIS
- Desktop GIS
- Distributed GIS (Internet GIS & Mobile GIS)

Mainframe GIS tarkoittaa paikkatietojärjestelmää, joka toimii yhdellä keskustietokoneella, jonka tietoihin ja ohjelmistoihin päästään käsiksi päätelaitteilla. Desktop GIS on paikkatietojärjestelmä, joka on asennettuna kaikkine ohjelmistoineen ja mahdollisine lisälaitteineen jollekin työasemalle. Järjestelmää pääsee käyttämään käynnistämällä tarvittava sovellus työasemalla. Tässä tapauksessa paikkatietojärjestelmän käyttämä paikkatieto voi sijaita joko samalla työasemalla kuin itse paikkatietojärjestelmä tai jollakin tarkoitusta varten pystytetyllä palvelimella. Edellä mainituissa tapauksissa paikkatieto ei välttämättä ole julkisesti suuren yleisön saatavilla, vaan lähinnä muutamien ammattilaisten välineenä heidän työnantajiansa tarkoituserien ajamiseksi. Käytössä olevat paikkatietosovellukset ovat myös monimutkaisia useine kehittyneine työkaluineen, ja niiden hallinta vaatii käyttäjältään asiantuntemusta.

Distributed GIS on näistä paikkatietojärjestelmätyypeistä nykyaikaisin, hajautetuin ja myös mielenkiintoisin. Distributed GIS voidaan jakaa kahteen alaluokkaan, Internet GIS:iin ja Mobile GIS:iin, yhteystavan perusteella (langallinen vai langaton). Mielenkiintoisen näistä kahdesta tekee se, että ne voivat tuoda paikkatiedon käytännössä lähes jokaisen ulottuville yksinkertaisessa muodossa. Toteutustavasta riippuen ei ole edes tarpeellista asentaa minkäänlaista erillistä ohjelmistoa tai apuvälinettä (esimerkiksi Java Runtime Environment), riittää kun navigoi laitteensa Internet-selaimen tiettyyn osoitteeseen. Tämän jälkeen käytettävissä voi olla mm. hakukenttä ja muutama yksinkertainen kontrolli karttakuvan muodostumisen ohjaamiseksi. Tällainen Internet-selaimen kautta toteutettu käyttöliittymä on ymmärrettävästi monelta osin rajoittunut, mutta suurimmalle osalle sen toiminnot ovat tarpeisiin nähden riittävät, eikä useille monimutkaisille työkaluille ole kysyntää. On

myös ehdottomasti huomattava, että yllä mainitsemani kahtiajako perinteiseen Internet-puoleen ja mobiilipuoleen on hämärtyvässä älypuhelinien, tablettietokoneiden, pienikokoisten kannettavien ja monentyyppisten nopeiden langattomien verkkojen yleistytessä ja korvaten lankaverkkoja.

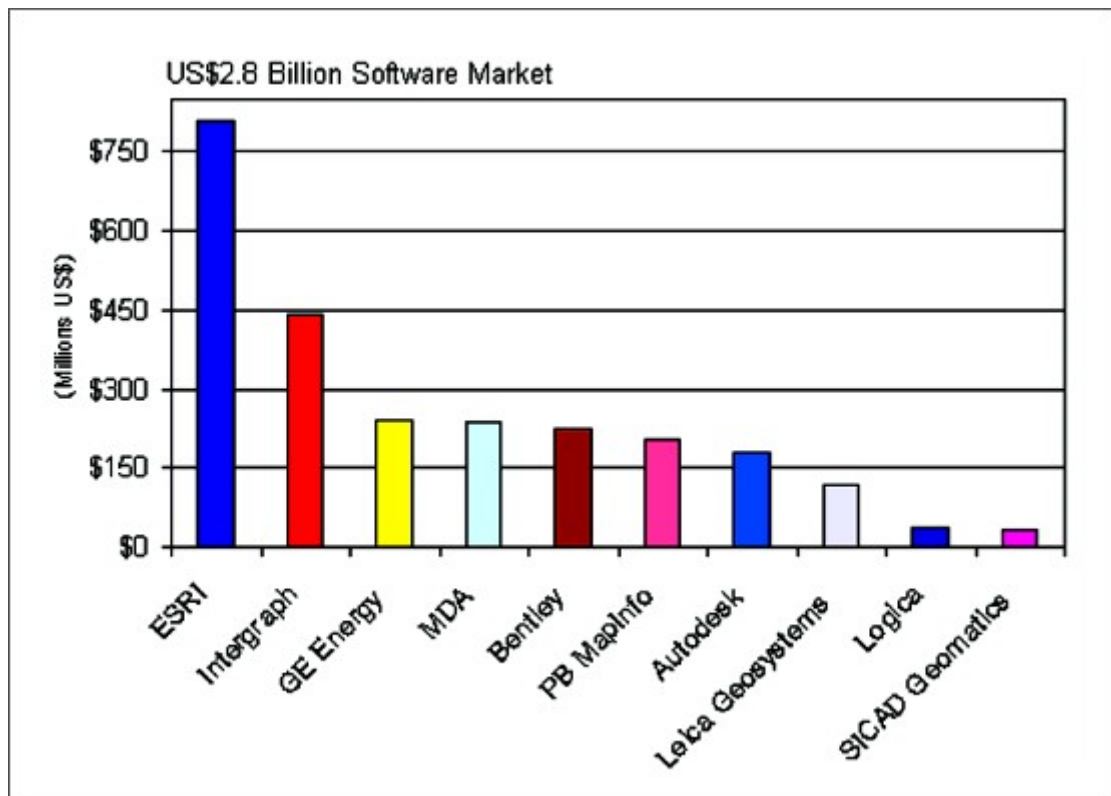
Vaikka työasema-GIS onkin omalla sarallaan hyödyllinen ja vartenotettava työkalu, se on kuitenkin monilta osin soveltumaton tavallisille ihmisille. Desktop-GIS onkin lähinnä työkalu ammattilaisille, jotka työskentelevät päivittäin spatiaalisen datan kanssa, usein yksityisissä yrityksissä tai julkishallinnon piirissä. Internet-GIS on lähempänä kenen tahansa vaatimuksia ja käyttöä selkeytensä ja helppoutensa ansiosta, ja sillä on selkeämmin havaittavissa oleva massatuotepotentiaali. On kuitenkin hyvä muistaa, että useimmiten verkossa toimivat kartta- ym. paikkatietoon pohjautuvat palvelut eivät myöskään ole täydellisiä paikkatietojärjestelmiä käyttäjilleen, vaan ne ovat paikkatietosovelluksia, koska niistä puuttuu täydelliseen paikkatietojärjestelmään olennaisesti kuuluviksi laskettuja osakokonaisuuksia, kuten uuden spatiaalisen datan syöttäminen.

”Tulevaisuudessa paikkatiedon hyödyntämisen aloituskynnys alenee. Käyttäjän ei tarvitse enää välttämättä hankkia paikkatieto-ohjelmistoa. Internet tulee tarjoamaan erilaisia paikkatiedon analysoinnin ja visualisoinnin palveluja, joita voi käyttää selaimella.” [6.]

”Se mikä on tapahtunut toimisto-ohjelmistoille, tapahtuu myös gis-ohjelmille. Verkossa voi luoda ja muokata dokumentteja - miksi ei siis myös paikkatietoja. Nykyaikaisissa tekstinkäsittelyohjelmissä on runsaasti ominaisuuksia, joista käyttäjällä ei ole hajuakaan. Samoin paikkatieto-ohjelmissä. Toki toiminnot ovat hyödyllisiä, etenkin ammattilaiselle. Mutta vähemmälläkin tulee toimeen - tai ainakin pääsee alkuun. Kaikkien ei tarvitse ostaa muutaman tai kymmenen tuhannen euron ohjelmistopakettia eikä asentaa eikä opetella sen käyttöä.” [6.]

### 3.1 Kaupalliset ohjelmistot

Paikkatietojärjestelmien alalla on useita kaupallisia toimijoita. Monet näistä tarjoavat asiakkailleen täydellistä GIS-järjestelmää, johon siis aiemmin mainitun perusteella kuuluu ohjelmistot karttadatan syöttämisestä sen esittämiseen.



**KUVA 2. Kaupallisten paikkatietojärjestelmien markkinaosuudet vuonna 2009 [9. Muokattu]**

Kuten yllä olevassa kuvassa 2 näkyy, on ESRI ja sen tuotteet varsin selkeä markkinajohtaja alan ohjelmistojen toimittajana.

ESRI (Environmental Systems Research Institute) on yhdysvaltalainen yritys, jonka alana on GIS-järjestelmien ja niihin liittyvien sovellusten kehittäminen ja markkinointi. ESRI käyttää tuoteperehestään nimeä ArcGIS, ja sen alle kuuluu monia paikkatietoon liittyviä ohjelmistoja palvelinympäristöistä työasemasovelluksiin, mobiililaiteratkaisuihin, verkkopalveluihin ja sovelluskehittäjille tarkoitettuihin työkaluihin. [10; 11.]

### 3.2 Ilmaisia ohjelmistoja

Myös paikkatietojärjestelmien ja -sovellusten alalla on ilmaisia open source -vaihtoehtoja. Esittelen nyt suppeasti yhden kokonaisen avoimen lähdekoodin paikkatietojärjestelmän ja kaksi paikkatietopalvelinohjelmistoa, jotka renderöivät karttakuvia spatiaalisen datan pohjalta web-käyttöä varten.

GRASS GIS (Geographic Resources Analysis Support System) on yksi huomattavimmista avoimen lähdekoodin paikkatietojärjestelmistä. GRASS on Open Source Geospatial Foundationin virallinen projekti, jonka tarkoituksena on ylläpitää ja kehittää paikkatietojärjestelmää spatiaalisen datan hallintaa, kuvien ja karttojen prosessointia, mallinnusta ja visualisointia varten. GRASS GIS tukee useita käyttöjärjestelmäalustoja. [12.]

MapServer ja Mapnik ovat avoimen lähdekoodin ohjelmistoja, joiden päämääränä on renderöidä käyttäjien pyyntöjen mukaan karttakuvia spatiaalisen datan pohjalta web-käyttöä varten. Ne eivät siis ole kokonaisia paikkatietojärjestelmiä, mutta molempia voidaan käyttää sellaisen rakentamisessa. Kumpikin ohjelmisto on saatavilla useille käyttöjärjestelmäalustoille. Mapnik-ohjelmistoa käyttää mm. OpenStreetMap-projekti (<http://www.openstreetmap.org/>). MapServerin Windows-versioon perehdyn itse tarkemmin myöhemmässä vaiheessa tätä työtä. [13; 14.]

### 3.3 Standardeja

Paikkatietojärjestelmien ja -sovellusten kehittyessä ja alan vallatessa lisää markkinoita alkoi ilmetä ongelmia. Käyttäjät pitivät kartoitukseen ja analysointiin kehitettyjen työkalujen potentiaalista, mutta näiden kalliiden ohjelmistojen laajennettavuuden ongelmat, käytön joustamattomuus ja rajoittuneisuus (tai pahimmillaan suoranainen kykenemättömyys) jakaa ja siirtää dataa systemistä toiseen alkoivat aiheuttaa kasvavassa määrin hankaluuksia. Tässä vaiheessa alalle alettiin kehittää standardeja, joilla pyrittiin yhtenäistämään käytäntöjä ja paremmin mahdollistamaan monien eri

valmistajien ja kehittäjien luomien järjestelmien keskinäinen tiedonsiirto. Tällä hetkellä näiden standardien ylläpidosta ja kehittämisestä on vastuussa Open Geospatial Consortium (OGC). Seuraavaksi esittelen suppeasti muutaman oleellisin Open Geospatial Consortiumin standardin.

Web Map Service -standardi (WMS) määrittää yksinkertaisen HTTP-rajapinnan karttakuvien pyytämiseen yhdestä tai useammasta geospaatiallisesta tietokannasta. WMS-pyyntö määrittää karttakerrokset ja kartta-alueen prosessointia varten. Vastaus WMS-pyyntöön on yksi tai useampia karttakuvia formaatissa, jonka voi näyttää web-selaimessa (JPEG, PNG jne, useimmiten rasterikuvia). Rajapinta tukee myös mahdollisuutta määrittää, tuleeko spatiaalisen datan pohjalta renderöityjen kuvien olla läpinäkyviä useilta palvelimilta tulevien karttakerrosten yhdistämiseksi. [15.]

Web Feature Service -standardi (WFS) määrittää rajapinnan, joka sallii pyynnöt maantieteellisiä ominaisuuksia varten käyttäen alustariippumattomia kutsuja. Maantieteellisillä ominaisuuksilla tarkoitetaan kartan muodostavia itsenäisiä objekteja, kuten vaikkapa maantien muodostava vektori tietokannassa. Erona WMS-standardiin on se, että nyt jokainen karttakuvan renderöintiin käytettävä objekti palautetaan pyynnön lähettäjälle erikseen, eikä valmiiksi renderöitynä kuvana. Tämä mahdollistaa loppukäyttäjälle kuvan muokkaamisen ja tehokkaan työkalupohjaisen spatiaalisen analysoinnin. [15.]

Web Coverage Service -standardi (WCS) määrittää standardirajapinnan ja -operaatiot jotka mahdollistavat alustariippumattomat pyynnöt maantieteellisistä peittoalueista. Peittoalueilla tarkoitetaan tässä yhteydessä objekteja tai kuvia samoin ominaisuuksin kuin WFS-standardissa, eli loppukäyttäjä voi muokata ja analysoida niitä. Standardissa mainituilla objekteilla ja kuvilla voidaan tarkoittaa mm. satelliitti- tai ilmakuvia, maanpinnan korkeusvaihteluita tai muuta dataa, joka voidaan ilmaista jollakin arvolla jokaisessa mittauspisteessä. [15.]

Open Geospatial Consortium on laatinut useita muitakin standardeja, kuten Web Map Tile Service (WMTS), mutta en käsittele niitä tässä työssä.

### 3.4 Karttadata

Paikkatietojärjestelmä tai -sovellus tarvitsee paikkatietoa digitaalisessa muodossa. Spatiaalisen datan hankkimiseen on monta tapaa, ja riippuu tavoitteesta ja resursseista, mikä niistä on kuhunkin tapaukseen sopivin menetelmä.

- Pisteiden kerääminen GPS-laitteistolla ja saadun tiedon syöttö tietokantaan tai halutun tiedostorakenteen luonti
- Karttojen digitointi
- Vapaassa levityksessä oleva itselle ladattava data. Monet tahot ovat laittaneet kartta-aineistojaan Internetiin kenen tahansa saataville. Joissakin tapauksissa on asetettu ehtoja aineistojen käytölle. Muutamia esimerkkejä aineistoista:
  - <http://www.naturalearthdata.com/> - täysin vapaasti käytettävissä olevaa karttadataa mihin tahansa tarkoitukseen. Vektoridata on ESRI shapefile-formaatissa, rasteridatan muotona TIFF + TFW world file.
  - <http://www.esri.com/data/free-data/index.html> – ladattavissa karttoja Yhdysvaltojen alueelta shapefile-formaatissa.
  - <http://www.mapcruzin.com/> - karttadataa monilta maailman alueilta useassa formaatissa, mm. shapefile. Käyttöehdot (jos niitä on) mainitaan tiedostojen yhteydessä.
  - <http://www.maanmittauslaitos.fi/ilmaisetaineistot> - saatavilla tiettyjä vektori- ja rasteriaineistoja sekä paikannimituotteita. Ainoana rajoitteena on aineiston alkuperäislähteen ja aineiston vuosiversion mainitseminen. Aineistojen lataaminen vaatii yhteystietojen luovuttamista.
- Jo olemassa olevan karttapalvelun ilmaisen rajapinnan käyttö. Google Maps API (Application programming interface) on ilmaiseksi käytettävissä niin kaupallisiin kuin epäkaupallisiin tarkoituksiin sillä ehdolla, että sivuston/palvelun on oltava julkisesti kaikkien saatavilla eikä se vaadi käytöstään maksua. Google kuitenkin pidättää itsellään oikeuden lisätä



palveluun mainoksia. [16.] Eräs vaihtoehto on OpenStreetMap, jota voidaan kutsua spatiaalisen datan wikipediaksi (kuka tahansa voi muokata ja lisätä karttadataa). OpenStreetMapin karttadataa voidaan käyttää mm. OpenLayers javascriptkirjaston avulla. [17.]

- Datan tai käyttöoikeuden osto. Kaupallisen rajapinnan käyttö, mm. Google Maps API premier ja Maanmittauslaitoksen rajapinta.

Erikoisuutena voidaan mainita Google Streetview, jonka toteutuksesta on vastannut Google. Muilla toimijoilla ei voi sanoa olevan vastaavaa yhtä laajaa systeemiä. Google Mapsin käyttämä muu karttadata on pääsääntöisesti peräisin Navteqilta ja Tele Atlakselta.

### 3.5 Tiedostoformaatit

Spatiaalista dataa varten on kehitetty monia formaatteja. On olemassa useita tietokantoja, joita voi käyttää spatiaalista tietoa varten joko natiivisti tai päivitysten ja laajennusten kautta. Monet kaupalliset paikkatietojärjestelmien alalla toimijat ovat luoneet järjestelmiään varten omia tiedostorakenteita ja -formaatteja. Lisäksi on olemassa kuvatiedostomuotoja, joihin voidaan liittää osaksi sijaintitietoa. Karttakuvien näyttämiseen web-palvelussa loppukäyttäjälle sovelletaan perinteisempiä kuvaformaatteja.

Paikkatietoa varten käytettävissä olevia tietokantaratkaisuja ovat mm. seuraavat:

- Oracle Spatial, erikseen lisensoitavissa oleva komponentti Oraclen tietokantaan
- Microsoft SQL Server 2008, johon on lisätty tuki spatiaaliselle datalle
- MySQL-tietokantaan kuuluu tuki spatiaaliselle tiedolle
- PostGIS lisää tuen spatiaaliselle datalle PostgreSQL-tietokantaan

Shapefile on ESRIn kehittämä tiedostoformaatti paikkatietojärjestelmiä varten. Shapefile sisältää vektoridataa (pisteitä, viivoja, polygoneja) ja niihin assosioitua ominaisuustietoa. Todellisuudessa shapefile koostuu useasta erillisestä tiedostosta, joilla on sama nimi, mutta eri tiedostomuoto. Shapefilen tärkein komponentti (\*.shp) sisältää spatiaalisen datan. Indeksitiedostoa (\*.shx) käytetään tarvittavan datan löytämiseen sitä vastaavasta \*.shp-tiedostosta. Spatiaalisen datan sisältävän \*.shp-tiedoston objekteihin liittyvä attribuuttitieto on säilötty \*.dbf-tiedostoon. Usein käytetään myös \*.prj-tiedostoa, jossa määritetään shapefilen projektiotiedot ja käytetty koordinaattisysteemi. Edellä mainittujen lisäksi on olemassa useita muita shapefilen yhteyteen kuuluvia tiedostoformaatteja, mutta nämä ovat oleellisimmat. [18.]

TIFF-kuvaformaatti on usein käytössä kun puhutaan rasterimuotoisista satelliitti- tai ilmakuvista. GeoTIFF viittaa TIFF-kuvatiedostoihin, joissa geografista tietoa on sisällytetty tiedostoon kuuluviin metatageihin. Tätä tageissa olevaa dataa voidaan käyttää esimerkiksi kuvan sijoittamiseen oikeaan kohtaan renderöitäessä karttakuvaa. [19.]

Taulukkoon 1 olen listannut vektori- ja rasteriaineistojen hyviä ja huonoja puolia.

#### **TAULUKKO 1. Vektori- ja rasteriaineistojen eroja [5, s. 71. Muokattu]**

	Vektoriaineisto	Rasteriaineisto
Tiedostokokoo	Pieni	Suuri
Yksittäisten kohteiden esittäminen	Helppoa	Kömpelöä
Jatkuvien ilmiöiden esittäminen	Vaikeaa	Helppoa
Luonnollisten muotojen tallentaminen	Helppoa	Vaikeaa
Aineiston muokkaus	Helppoa	Vaikeaa
Aineiston tuottaminen	Työlästä	Vaivatonta
Aineiston hinta	Kallis	Halpa

Kun puhutaan web-karttakuvissa käytetyistä tiedostomuodoista, jolloin tarkoituksena on vain esittää palvelimella sijaitsevan ohjelmiston työn tulos loppukäyttäjälle, käytössä ovat yleisesti selaimissa toimivat rasterikuvaformaattit (png, jpg, gif). Useimmiten hyödynnetään läpinäkyvyyttä tukevia PNG-kuvia, joiden

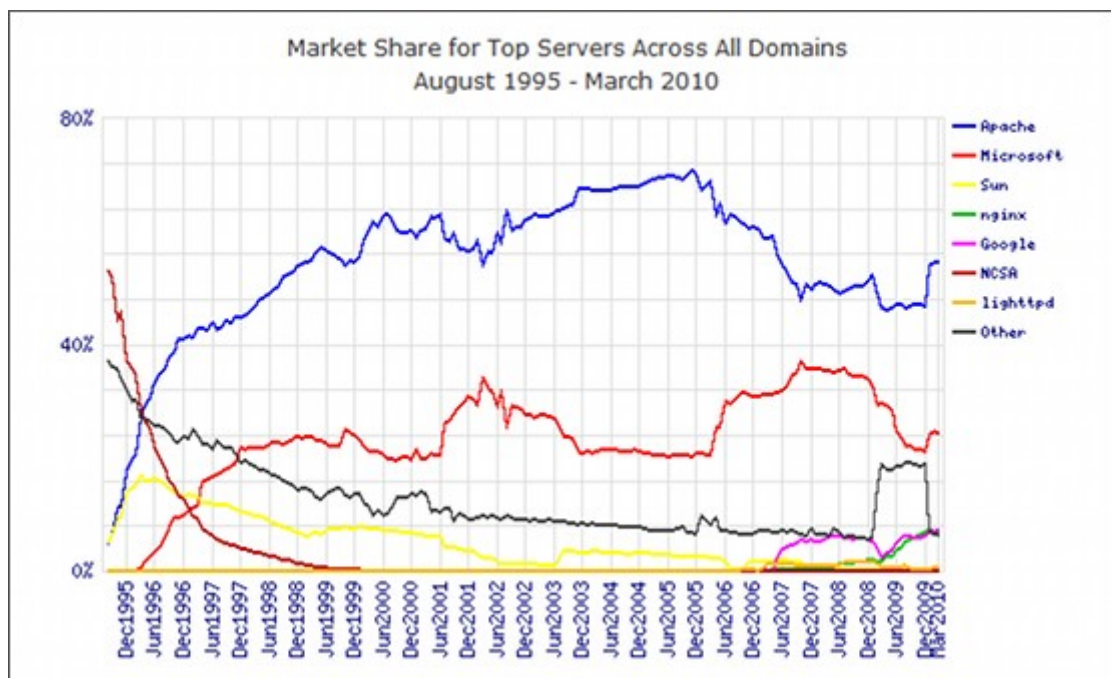
pakkausmenetelmä on häviötön. Koska karttakuvat ovat varsin yksinkertaisia ja niissä käytetään vain muutamia värejä, saadaan ne pakattua tehokkaasti myös png-muotoisiksi, jolloin vältetään jpg-pakkauksen aiheuttamat artifaktit. Tuki läpinäkyvyydelle on myös erittäin hyödyllinen, tällöin voidaan tarvittaessa helposti muodostaa kerrostettuja kuvia ja antaa käyttäjälle kontrolli sen suhteen, mitä näytetään ja mitä ei. Loppukäyttäjälle näkyvissä karttakuvissa voitaisiin myös käyttää SVG-vektorikuvaformaattia (Scalable Vector Graphics), mutta se ei ole vielä kovinkaan yleinen Internet-selainten puutteellisen tuen vuoksi. Yksi vektorikuvan oleellinen etu verrattuna rasterikuvaan on mahdollisuus zoomata rajattomasti laadun kärsimättä.

#### 4 MAPSERVER FOR WINDOWS

Nyt keskityn tarkemmin yksittäiseen valitsemaani open source -karttapalvelinohjelmistoon. Syvällisemmän tutkimuksen kohteeksi valitsin MapServerin Windows-version. Valintani syynä on kyseisen karttapalvelinohjelmiston pitkä kehityshistoria, avoin lähdekoodi, sen soveltuvuus monille käyttöjärjestelmäalustoille sekä sille laaditut suhteellisen laajat valmiit sovellukset palvelinohjelmiston yhteydessä ajettavaksi. MapServer for Windows on myös ilmainen ohjelmisto, mikä on yksi opinnäytetyöni toimeksiantajan, Mikkelin ammattikorkeakoulun, asettama vaatimus.

Tällä hetkellä MapServer eri versioinaan on tunnetuin ja käytetyin ilmainen karttapalvelinohjelmisto. MapServer toimii alustana sovelluksille, joita käytetään paikkatietojen ja interaktiivisten karttojen esitykseen Internetissä. Alun perin MapServer kehitettiin Minnesotan yliopistossa Linux-ympäristöön. MapServer-palvelinohjelmistosta on myös laadittu eri versioita muissa käyttöjärjestelmäympäristöissä ajettavaksi, ja tällä hetkellä se voidaankin asettaa toimimaan kaikilla yleisimmillä käyttöjärjestelmäalustoilla (Windows, Linux, Mac OS X, Solaris ym.). [13.] MapServer tulee useita Open Geospatial Consortiumin standardeja, mukaan lukien aiemmin työssä mainitsemani WMS, WFS ja WCS. MapServer -palvelinohjelmisto on julkaistu MIT-lisenssin alaisuudessa, joka sallii käyttäjän vapaasti muokata, kopioida ja levittää ohjelmistoa sillä ehdolla, että lisenssin teksti säilyy lähdekoodissa. [20.]

MapServer ei ole kokonainen paikkatietojärjestelmä, vaikka sitä voitaisiin käyttää sellaisen rakentamisessa. MapServer on työkalu spatiaalisen datan renderöimiseen web-käyttöä varten. MapServer voi toimia kahdessa eri toimintatilassa, nämä ovat CGI (Common Gateway Interface) ja MapScript. CGI-tilassa MapServerin palvelinohjelmisto delegoi web-sivun luomisen jollekin itsenäiselle sovellukselle. Näitä sovelluksia kutsutaan CGI-skripteiksi. MapServerin tapauksessa toiminta siirretään `mapserv.exe` -sovellukselle. CGI-tilassa voidaan helposti luoda yksinkertaisia sovelluksia. MapScript-tilassa MapServerin API on käytettävissä useilla ohjelmointikielillä, ja tällöin on mahdollisuus rakentaa toiminnaltaan paljon monimutkaisempia sovelluksia, koska karttaelementtien toimintaan voidaan vaikuttaa suoraan. MapServer for Windows tukee tällä hetkellä MapScriptin käyttöä PHP:llä, Javalla, Pythonilla ja CSharpilla. [21, Introduction.]



**KUVA 3. Web-palvelimien markkinaosuudet [22.]**

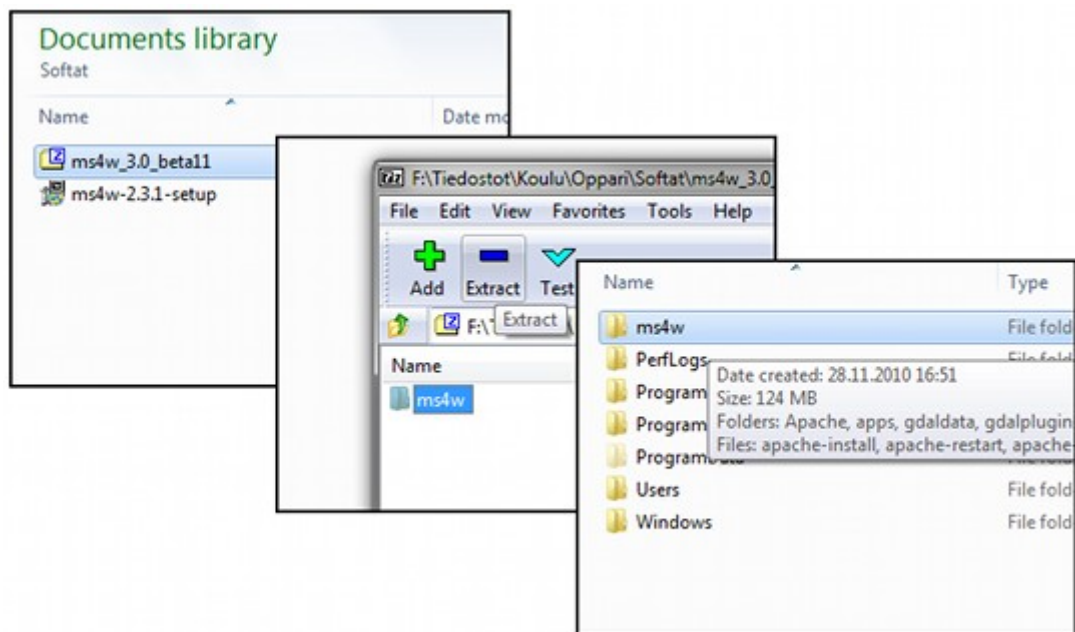
MapServer for Windows perustuu Apache-palvelinohjelmistoon. Tällä hetkellä Apache on selvästi suosituin web-palvelinohjelmisto (kuva 3). Apache on MapServerin tavoin luokiteltavissa open source-tyyppiseksi ohjelmistoksi, jota ylläpitää ja kehittää avoin ohjelmistokehittäjien yhteisö. Apache on ominaisuuksiltaan ja vakaudeltaan riittävä useimpiin tarkoituksiin, joten ilmaisuutensa vuoksi se on

ymmärrettävästi noussut varsin suosituksi palvelinalustaksi Internet-sivustoille. Apachelle on myös laadittu lukuisia lisäosia ja laajennuksia, joilla palvelinohjelmiston toimintoja voidaan tarvittaessa monipuolistaa entisestään.

#### 4.1 MapServer for Windowsin asentaminen

Seuraavaksi käydään läpi uusimman MapServer for Windows-palvelinohjelmiston kehitysversion (versio 3.0 Beta 11, 20 huhtikuuta 2010) asentaminen 64-bittisellä Windows 7 Home Premium -käyttöjärjestelmällä varustettuun PC-koneeseen, jossa ei ole aikaisemmin ollut asennettuna kyseistä ohjelmistoa. Ohjelmiston asentamiseen on kaksi lähestymistapaa, joko perinteinen asennustiedosto tai zip-paketti, joka sisältää kaikki ohjelmistoon kuuluvat tiedostot.

Käsittelen ohjelmiston asentamisen zip-paketista, koska se on testikäyttöä ajatellen yksinkertaisin vaihtoehto. Asentaminen tapahtuu yksinkertaisesti purkamalla ladattu zip-paketti, mikä sisältää valmiiksi konfiguroidun palvelinohjelmiston. Paketin voi purkaa minne tahansa, mutta on suositeltavaa purkaa paketti jonkin kiintolevyaseman juureen (kuva 4), jolloin vältetään ylimääräiseltä asetusten muokkaamiselta, eli käytännössä muutamien asetustiedostojen polkumäärittämiä joutuu muuttamaan manuaalisesti osoittamaan oikeaan paikkaan.



**KUVA 4. MS4W:n asentaminen zip-paketista**

Kun paketti on purettu halutun kiintolevyaseman juureen, yleensä voidaan käydä ajamassa ms4w-kansiossa sijaitseva apache-install.bat -komentojonotiedosto. Tämä asentaa MS4W Apache Web server -palvelun, joka tästä lähtien käynnistyy aina käyttöjärjestelmän käynnistyksen yhteydessä. Tässä vaiheessa on kuitenkin huomioitava eri Windows-käyttöjärjestelmien erot. Windows 95/98/ME käyttöjärjestelmissä Apachen asentaminen palveluksi ei onnistu, ja palvelin on joka kerta käynnistettävä erikseen ajamalla ms4w/Apache/bin/httpd.exe. Jos palvelinta on tarkoitus pitää päällä aina koneen ollessa käynnissä, on järkevintä asettaa httpd.exe ajettavaksi automaattisesti joka kerta käyttöjärjestelmän käynnistymisen yhteydessä.

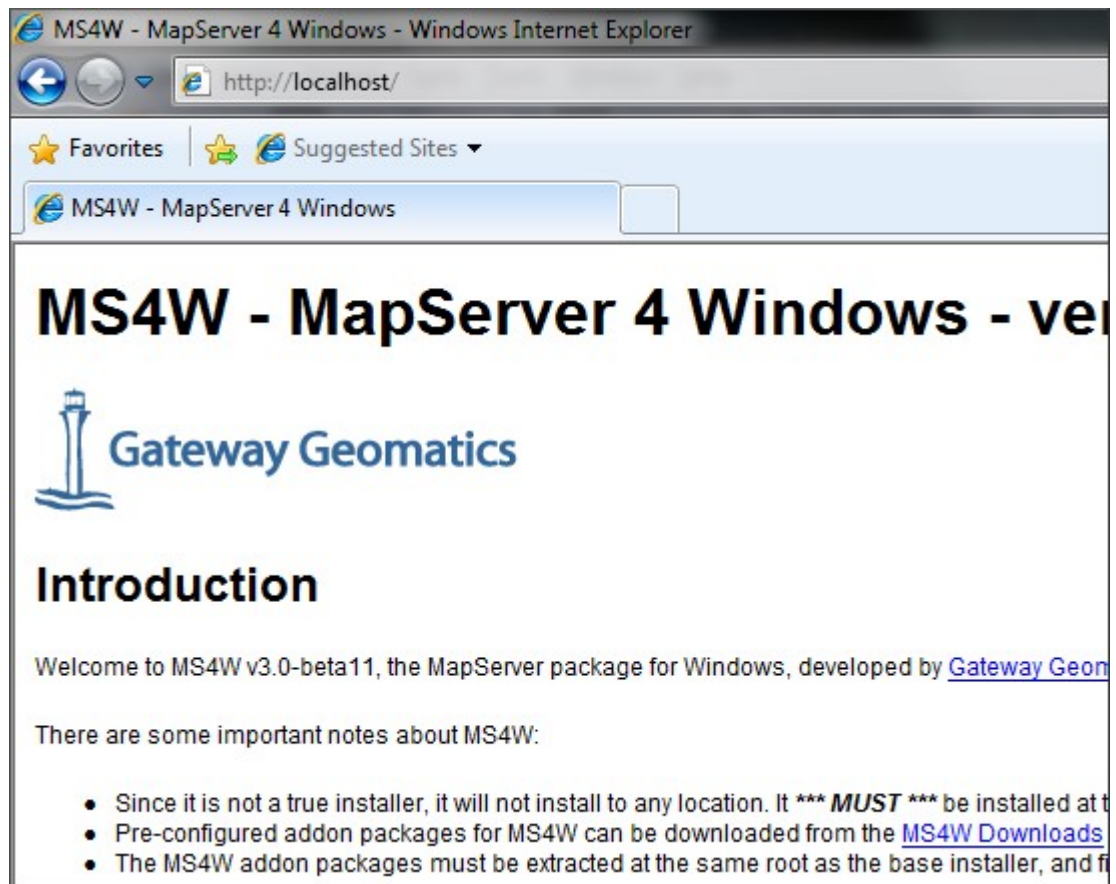
Windows 7 ja Windows Vista ovat myös poikkeustapauksia asentamisen suhteen. Saadakseen Apachen asennettua palveluksi taustalle on käyttäjän ajettava komentokehote järjestelmänvalvojana, navigoitava sitä käyttäen ms4w-kansioon ja ajettava apache-install.bat komentokehotteen kautta. Mielestäni paras vaihtoehto satunnaiseen testikäyttöön (joka koskee myös itseäni) on toimia kuten Windows 98/95/ME tapauksissa, eli ajaa manuaalisesti ms4w/Apache/bin/httpd.exe joka kerta, kun palvelinohjelmistoa halutaan käyttää.

Toinen mahdollinen tapa asentaa MS4W on sen kotisivuilta saatavissa oleva valmis .exe-muotoinen asennuspaketti. Sen avulla on helposti mahdollista:

- asentaa palvelinohjelmisto mihin tahansa kansioon
- määrittää Apachen käyttämä portti verkkoliikennettä varten
- luoda tarpeelliset pikakuvakkeet Käynnistä-valikkoon ja/tai työpöydälle
- luoda palvelinohjelmiston asennukselle oma merkintänsä Windowsin lisää- tai poista sovelluksia -valikkoon helppoa asennuksen poistoa varten.

Valmiin ajettavan asennuspaketin huono puoli on kuitenkin sen verrattain hidas päivitystahti, joten sen mukana tulee yleensä jonkin verran vanhempia komponentteja. Tämän vuoksi valitsin zip-paketin. Minulla ei myöskään ole tarvetta automaattisesti

luoduille kuvakkeille tai muutoksiin kansiorakenteessa ja verkkoyhteysasetuksissa, koska ohjelmisto on minulla testikäyttöä varten.



**KUVA 5. MS4W-asennuksen toimivuuden tarkistus**

MS4W:n asennuksen toimivuuden voi tarkistaa avaamalla osoitteen <http://localhost/> Internet-selaimessa (kuva 5), kun `ms4w/Apache/bin/httpd.exe` on käynnissä. Palvelimen toimiessa oikein avautuu näytölle MS4W infosivu, jossa mm. kerrotaan, mitä komponentteja kuuluu palvelimen perusasennukseen ja kuinka monta lisäkomponenttia on asennettuna.

## 4.2 Palvelinohjelmiston komponentit

MapServer for Windows (MS4W) on alkuperäisen Linux-pohjaisen MapServer-ohjelmiston käännös Windows-ympäristössä käytettäväksi. Palvelinohjelmiston tarkoitus on toimia renderöinti- ja web-palvelinalustana erilaisille sovelluksille, joita voidaan käyttää paikkatietojen ja interaktiivisten karttojen esitykseen Internetissä.

Apache-palvelimen lisäksi pakettiin siis luonnollisesti kuuluu muita komponentteja, joilla lisätään muun muassa tuki useille karttadataa varten käytössä oleville spatiaalisen datan tiedostoformaateille ja mahdollisuus käyttää useita yleisiä ohjelmointikieliä sovellusten luomiseen ja käyttämiseen palvelinohjelmiston kanssa.

Seuraavassa on lueteltuna ja tarkemmin selitettynä kaikki MapServer for Windows -peruspakettiin (v3.0 Beta 11) kuuluvat komponentit [23]:

- Apache version 2.2.15 – MapServerin pohjana toimiva avoimeen lähdekoodiin perustuva HTTP-palvelinohjelma.
- PHP version 5.3.2 – PHP on ohjelmointikieli, jota käytetään dynaamisten web-sivustojen luontiin.
- MapServer 5.6.3 CGI and MapScript (CSharp, Java, PHP, Python) – viimeisimmät versiot MapServerin omasta CGI-ohjelmasta sekä tätä kehittyneemmästä MapScriptistä.
- GDAL 1.7.1 – GDAL-kirjasto (Geospatial Data Abstraction Library) lisää MapServeriin tuen useille geospaatialisen datan rasteriformaateille, esimerkkinä mainittakoon GeoTIFF.
- mapserver utilities
- gdal/ogr utilities – GDAL ja OGR apuohjelmia. OGR on kirjasto monien vektoriformaattien lukemiseen, mukaan lukien ESRI shapefile.
- proj.4 utilities – proj.4 on kirjasto kartografisten projektoiden ja koordinaattisysteemien käsittelyyn.
- shp2tile utility -apuohjelma, joka jakaa suurikokoisen shapefilen "ruudukoksi" useisiin pienempiin shapefileihin.
- shapelib utilities - shapelib-kirjasto tarjoaa mahdollisuuden kirjoittaa yksinkertaisia C-ohjelmia lukemaan, kirjoittamaan ja (rajoitetusti) päivittämään ESRI Shapefilejä ja niiden attribuuttitietoja.
- shpdiff utility -apuohjelma, joka vertaa kahta shapefileä ja tulostaa yhteenvedon löydetyistä eroavaisuuksista.



- avce00 utilities - kirjasto, joka mahdollistaa Arc/Info vektorimuotoisten peittoalueiden lukemisen ja kirjoittamisen kuin ne olisivat E00-tiedostoja. E00 on ESRI ArcInfon käyttämä tiedostomuoto peittoalueiden kuvaamiseen.
- PHP\_OGR Extension 1.1.1 - lisää mahdollisuuden käyttää OGR-kirjaston toimintoja PHP:llä.
- OWTChart 1.2.0 - OWTChart Engine muodostaa GIF-muotoisen kaavion sarjasta syöttöparametrejä.
- msTileCache utilities.

Luonnollisesti palvelinohjelmistoon voidaan lisätä vielä enemmän ominaisuuksia ja toiminnallisuutta laittamalla mukaan uusia komponentteja. MapServer on avoimen lähdekoodin ohjelmisto, joten kuka tahansa voi halutessaan rakentaa siitä itselleen perusversiosta poikkeavan ohjelmiston.

### 4.3 Tiedosto- ja hakemistorakenne

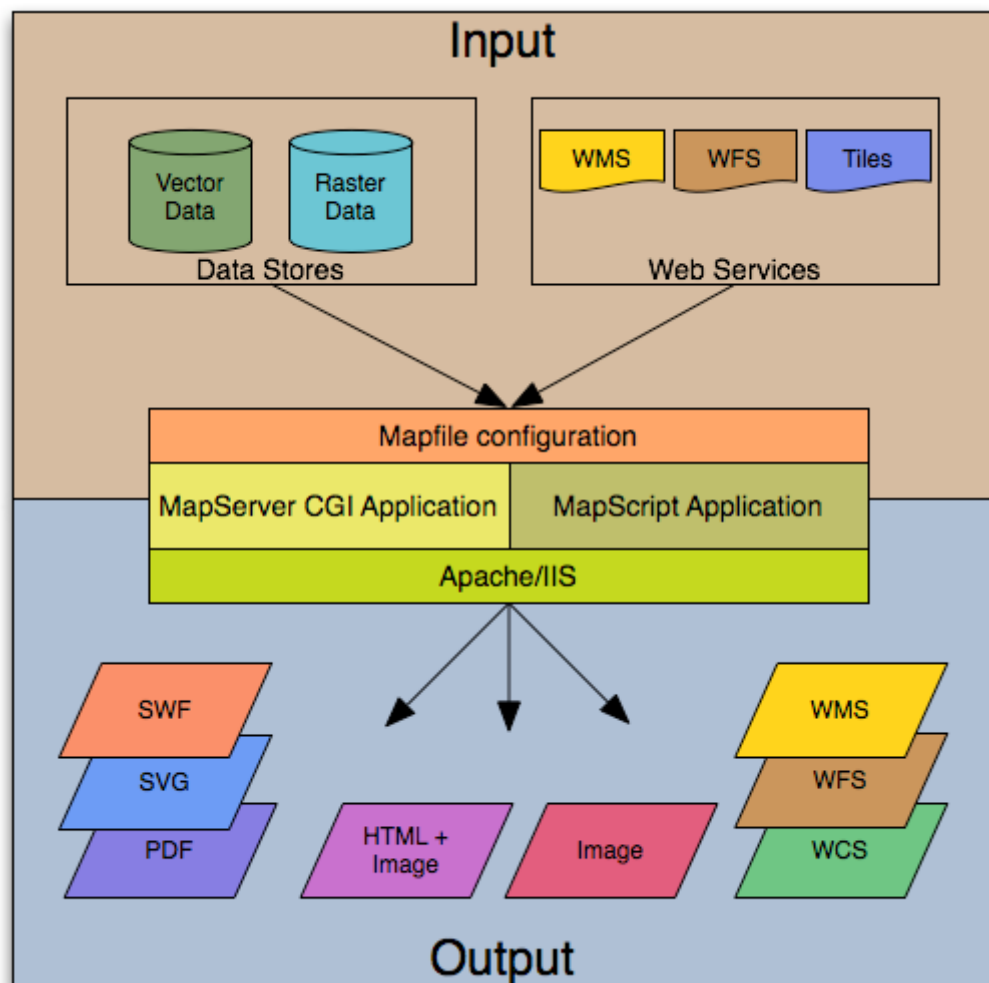
Nyt selvennän hieman MapServerin tiedosto- ja hakemistorakennetta [23]. Käyn lävitse hakemistorakenteen oleellisimmilta osiltaan ja mainitsen MapServerin toiminnan kannalta tärkeimpien tiedostojen sijainnit.

- ms4w/Apache/bin - Apachen binääritiedostot Windowsia varten käännettynä, mm. httpd.exe.
- ms4w/Apache/cgi-bin - mapserv.exe ja sitä tukevat kirjastot löytyvät täältä, kuten myös php.ini ja php.exe.
- ms4w/Apache/cgi-bin/ignored-libmap - libmap.dll Oraclea varten. Tiedosto on otettava erikseen käyttöön, jos sitä tarvitaan.
- ms4w/Apache/cgi-bin/mapsript - Mapsript-tiedostot Javaa, Pythonia ja CSharpia varten.

- ms4w/Apache/conf - Apachen konfiguraatiotiedostot. Näitä käyttäjän ei tarvitse muokata. Sovelluskohtaiset ja paikalliset httpd.conf -tiedostot tulee sijoittaa /ms4w/httpd.d -hakemistoon.
- ms4w/Apache/error - tiedostoja Apachen käytettäväksi virhetilanteissa.
- ms4w/Apache/htdocs - Apachen web-juurihakemisto. Tässä hakemistossa oleviin tiedostoihin pääsee käsiksi webin kautta.
- ms4w/Apache/logs - Apachen lokitiedostot menevät tänne.
- ms4w/Apache/modules - Apachen moduuleja, jotka ovat vastuussa palvelinohjelmiston toiminnallisuudesta.
- ms4w/Apache/php - PHP:n asennus, täältä löytyy mm. php\_mapscript.dll.
- ms4w/Apache/specialplugins - plugin-tiedostoja MapServeriä varten. Näillä saadaan tuki Microsoft SQL Server 2008:lle ja ESRI:n ArcSDE-tietokannoille.
- ms4w/apps - MapServeriä varten laaditut sovellukset tulevat tänne.
- ms4w/gdaldata - tiedostoja GDAL-formaattien tukemiseksi.
- ms4w/gdalplugins - hakemisto dll-tiedostoja varten, joita GDAL-pluginit vaativat.
- ms4w/httpd.d - sovelluskohtaiset ja paikalliset httpd.conf -tiedostot kuuluvat tähän hakemistoon.
- ms4w/proj - PROJ.4-kirjastoon liittyvät tiedostot.
- ms4w/python - tiedostot Pythonia varten.
- ms4w/tmp - hakemisto väliaikaistiedostoille.
- ms4w/tmp/ms\_tmp - hakemisto väliaikaistiedostoille, joihin on tarve päästä käsiksi webin kautta, esimerkiksi MapServerin renderöimät karttakuvat. Hakemisto kannattaa tyhjentää ajoittain, sillä suhteellisen lyhyessä testikäytössäkin tänne kertyy useita satoja megatavuja kuvatiedostoja.
- ms4w/tools - sisältää hyödyllisiä työkaluja, kuten GDAL/OGR- ja mapserv-apuohjelmia.

#### 4.4 Ohjelmointiesimerkkejä

MapServerille voi laatia sovelluksia käyttäen CGI-rajapintaa tai MapScriptiä (kuva 6). Havainnollistan nyt hieman MapServerin käyttöä yksinkertaisten esimerkkien avulla. Esimerkit perustuvat lähteisiin [21] ja [24]. Ensin luon CGI-rajapintaa hyödyntävän sovelluksen, joka piirtää kartan ja näyttää sen käyttäjälle muutaman kontrollitoiminnon kera. Samalla selvennän MapServerille ominaisen mapfilen käyttöä, joka sisältää kartan piirtämisessä käytettäviä asetuksia ja tyylimäärittelyitä.



KUVA 6. MapServer-sovelluksen perusarkkitehtuuri [24, s.7]

Tämän jälkeen muunnan luomani sovelluksen käyttämään PHP MapScriptiä. MapScript mahdollistaa monipuolisemmat ja vaativammat sovellukset, koska kartan piirtoon vaikuttaviin elementteihin voidaan vaikuttaa suoraan, esimerkiksi käytettävää mapfileä voitaisiin vaihtaa dynaamisesti zoomaustason mukaan. Esimerkeissä käyttämäni kartta-aineisto on peräisin kahdelta sivustolta:

- <http://www.naturalearthdata.com/>
- <http://www.mapcruzin.com/free-finland-arcgis-maps-shapefiles.htm>.  
Tiedostojen käyttöehtoihin kuuluu vaatimus linkittää [www.openstreetmap.org](http://www.openstreetmap.org) ja [www.mapcruzin.com](http://www.mapcruzin.com) verkkosivustoille.

Sivustoilta lataamani kartta-aineistot käsittävät rasteri- ja vektoridataa. Vektorimuotoinen karttadata on shapefile-formaatissa ja rasteridata TIFF-kuvana, jonka mukana tulee TFW-muotoinen tiedosto (world file, ESRIn spesifikaatio), joka määrittelee rasterikuvan sijainnin, koon ja kulman kartalla.

Aluksi luon ms4w/Apache/htdocs -hakemistoon uuden kansion, jolle annan nimeksi ”maps”. Tähän hakemistoon siirrän kaikki kartan renderöintiin käytettävän datan, niin shapefilet kuin rasterikuvat aputiedostoineen. Itse käytän htdocs-hakemistoa karttadatalle kahdesta syystä: käytän MapServeriä vain testikäyttöön ja on yksinkertaisempaa kokeilujen kannalta pitää data samassa hakemistossa kuin muutkin asiaan liittyvät dokumentit. Jos MapServeriä on tarkoitus käyttää julkisesti karttapalvelimena, suosittelen sijoittamaan karttadatan muualle.

Seuraavaksi luon html-dokumentin ja sille lomakkeen (kuva 7), jonka avulla MapServer saadaan renderöimään halutun tyyppinen karttakuva aineistojen pohjalta.

```

6  <form method=POST action=" ../cgi-bin/mapserv.exe">
7      <input type="submit" value="Käynnistä">
8      <input type="hidden" name="program"
9          value=" ../cgi-bin/mapserv.exe">
10     <input type="hidden" name="map"
11         value="c:/ms4w/Apache/htdocs/kokeilu.map">
12     <input type="hidden" name=zoomsize size=2 value=2>
13     <input type="hidden" name="layers"
14         value="background countries boundarylines
15             waterways urbanareas roads railways buildings
16             places">
17     </form>

```

**KUVA 7. HTML-lomake**

Rivillä 6 määritellään polku lomakkeen tiedot käsittelevälle CGI-skriptille, tässä tapauksessa mapserv.exelle. Lomakkeen tiedot lähetetään käyttämällä POST-metodia. Riveillä 10 ja 11 määritellään mapfile, jota mapserv.exe käyttää karttakuvan renderöintiin. Lisäksi määritellään muutamia muita piilotettuja arvoja, joita tarvitaan myöhemmässä vaiheessa (esimerkiksi karttakerroksien nimet, joiden halutaan olevan käyttäjän valittavissa). Kun äskeisen lomakkeen sisällään pitävän HTML-tiedoston avaa selaimessa, näkyvillä tulee ”Käynnistä” -nappula, jota painamalla lomakkeen tiedot lähetetään mapserv.exelle.

Kun mapserv.exe saa itselleen lomakkeen tiedot, se ensiksi etsii määritellyn mapfilen. Mapfile (\*.map) on tekstitiedosto, joka määrittää karttakuvan piirtämisessä käytetyt asetukset (kuva 8).

```
1 MAP
2 NAME "Kokeilu"
3 SIZE 1400 1000
4 IMAGECOLOR 240 240 240
5 IMAGETYPE png
6 SHAPEPATH "/ms4w/Apache/htdocs/maps/"
7 EXTENT 22.00 60.00 30.00 65.00
8
9 OUTPUTFORMAT
10     NAME png
11     DRIVER "GD/PNG"
12     MIMETYPE "image/png"
13     IMAGEMODE RGBA
14     EXTENSION "png"
15 END
16
17 WEB
18     TEMPLATE "kokeilu2.html"
19     IMAGEPATH "/ms4w/tmp/ms_tmp/"
20     IMAGEURL "/ms_tmp/"
21 END
```

**KUVA 8. Määrittelyitä esimerkissä käytetystä mapfilestä**

- Mapfile aloitetaan aina MAP-objektilla. Kaikki muu sisältö tiedostossa tulee MAP-objektin alle.

- END lopettaa objektin.
- SIZE määrittää renderöitävän karttakuvan koon pikseleissä.
- IMAGECOLOR on kuvan pohjaväri, jolla näkyvät kaikki ne alueet, jotka eivät jää piirrettäessä muodostuvan kuvadatan alle.
- IMAGETYPE kertoo, missä formaatissa renderöidyn, käyttäjälle lähetettävän karttakuvan tulee olla.
- SHAPEPATH määrittää polun käytettävälle karttadatalle.
- EXTENT kuvaa maantieteellistä aluetta, joka karttakuvaan piirretään.
- OUTPUTFORMAT-objektissa määritellään renderöitävän kuvan formaatti tarkemmin.
- WEB-objekti määrittää template-tiedoston, jonne mapserv.exe palauttaa käsittelynsä tulokset. Template voi olla myös sama tiedosto kuin se, jolta käsiteltävät tiedot lähetettiin. IMAGEPATH määrittää, minne mapserv.exe sijoittaa valmiit karttakuvat. IMAGEURL määrittää osoitteen, jolla selain löytää renderöidyn kuvan.

```

117
118 LAYER
119     NAME "roads"
120     STATUS on
121     TYPE line
122     DATA "roads"
123     OPACITY 50
124     LABELITEM "name"
125     LABELCACHE on
126     CLASSITEM "type"
127

```

**KUVA 9. Esimerkki LAYER-objektista mapfilessä**

Jokaisessa mapfilessä pitää olla vähintään yksi LAYER-objekti (kuva 9). Jokainen karttakerros on yhdistelmä karttadataa ja tyylimäärittelyitä. MapServer piirtää mapfilessä määritellyt karttakerrokset järjestyksessä ensimmäisestä alkaen. Jos myöhemmin piirrettävässä karttakerroksessa on objekteja samalla kohdalla kuin jossain aiemmassa kerroksessa, piirretään ne aiemman datan päälle.

- NAME määrittää karttakerroksen nimen.
- STATUS kertoo, piirretäänkö karttakerrosta vai ei. Arvolla default karttakerros piirretään aina. Muita mahdollisia arvoja ovat on ja off.
- TYPE määrittää karttakerroksen piirtotavan. Käytettäviä tapoja ovat mm. point, line, polygon ja raster. Piirtotavan ei tarvitse välttämättä olla sama kuin karttakerroksessa käytettävän karttadatan tyyppi. Jos karttakerroksen datalle on määritetty sopimaton tyyppi, antaa MapServer virheilmoituksen.
- DATA määrittää karttakerroksen piirtoon käytettävän karttadatan. Jos tiedostopäätettä ei ole merkitty, MapServer olettaa sen olevan .shp.
- OPACITY määrittää karttakerroksen läpinäkyvyyden. Arvolla 0 kerros on täysin läpinäkyvä, arvolla 100 kerros ei ole lainkaan läpinäkyvä.
- LABELITEM määrittää, mitä ominaisuustietoa käytetään karttakerroksen objektien nimeämiseen. Kartta-aineistoni ovat varsin yksinkertaiset, joten voin yleensä valita nimeämisen vain piirrettävien objektien ID:n, nimen (esim. Pasilan asema, Herttoniemen sairaala) ja tyyppin (residential, sport, public\_building, jne.) välillä.
- LABELCACHE määrittää, piirretäänkö karttakerroksen objektien nimittekstit yhtä aikaa objektien kanssa, vai jätetäänkö ne muistiin ja piirretään viimeisenä, yhdessä kaikkien muiden kerrosten nimitekstien kanssa. Tätä toimintoa hyödyntämällä voidaan määrittää mitä tekstejä piirretään milloinkin ja asemoida niitä järkevämmiin.
- CLASSITEM määrittää mitä objektin ominaisuustietoa käytetään CLASS-objektien sisällä niiden suodatukseen eri kategorioihin.

```

127
128     CLASS
129         EXPRESSION "motorway"
130
131         STYLE
132             COLOR 255 0 0
133             WIDTH 3
134         END
135
136         LABEL
137             COLOR 255 255 255
138             SIZE small
139             ANGLE auto
140             REPEATDISTANCE 100
141             MINFEATURESIZE 15
142             PRIORITY 5
143             BUFFER 15
144         END
145     END
146

```

**KUVA 10. Esimerkki CLASS-objektista LAYER-objektin sisällä**

CLASS-objektia käytetään yksittäisten kohteiden piirtoon (kuva 10). Jokaista piirrettävää kohdetta (esim. vektori) verrataan jokaiseen CLASS-objektiin ensimmäisestä alkaen, kunnes löytyy sellainen, jossa määritelyihin ehtoihin kohde sopii. Tässä tapauksessa LAYER-objektissa määriteltyä ”type”-ominaisuutta verrataan kussakin CLASS-objektissa käyttämällä EXPRESSION-komentoa. Rivillä 128 alkaneen CLASS-objektin mukaisesti piirretään ne objektit roads.shp -tiedostosta, joiden tyyppi on määritelty ”motorway” shapefilen \*.dbf-komponentissa.

Jokainen CLASS-objekti voi pitää sisällään useita STYLE-objekteja, joita käytetään symboleiden piirtämiseen. Tässä tapauksessa käytössä on vain yksi STYLE-objekti, jossa määritän moottoriteiden piirtoon käytettävän värin RGB-arvon ja viivan leveyden. LABEL-objektia käytetään tekstin piirtämiseen kartalle. LAYER-objektin määrittelyissä olin valinnut LABELITEM arvoksi ”name”, joten tekstinä käytetään tieosuuden nimeä.

- COLOR määrittää piirrettävän tekstin värin RGB-arvona

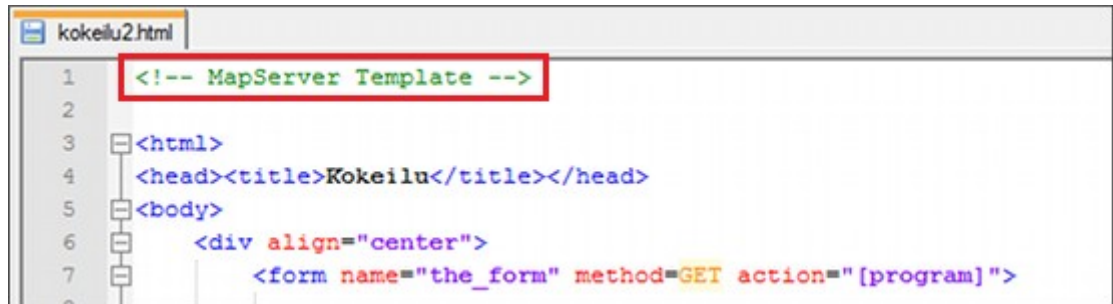


- SIZE määrittää piirrettävän tekstin kirjainten koon. Bitmap-fonteille voidaan käyttää avainsanoja tiny, small, medium, large ja giant. Truetype-fontteja käytettäessä voidaan määrittää fontin pikselikokoa ilmaiseva lukuarvo.
- ANGLE määrittää piirrettävän tekstin kulman. Arvolla ”auto” MapServer päättää itse, minkä verran tekstiä kallistetaan.
- REPEATDISTANCE määrittää, kuinka usein identtinen teksti (kuten tien nimi) toistetaan. Arvo ilmoitetaan pikseleinä.
- MINFEATURESIZE määrittää, kuinka suuri (pikseleinä) objektin tulee olla kartalla, ennen kuin siihen liittyvä teksti piirretään.
- PRIORITY määrittää objektin tekstin prioriteetin. Kun karttakerrosten tekstit jätetään muistiin (LABELCACHE on), ne piirretään lopuksi prioriteetin mukaisessa järjestyksessä.
- BUFFER määrittää, kuinka suuri tyhjä alue (pikseleinä) tulee jättää kahden tekstin väliin.

Nyt MapServer on siis saanut käyttäjän lähettämän lomakkeen tiedot, lukenut mapfilen ja sijoittanut määritysten mukaisesti renderöidyn karttakuvan /ms4w/tmp/ms\_tmp/ -hakemistoon. Tämän jälkeen MapServer lukee mapfilessä määritetyn template-tiedoston ja korvaa siitä tietyt avainsanat, jotka on kirjoitettu hakasulkujen sisään. Esimerkkejä avainsanoista:

- [img] korvataan polulla renderöityyn karttakuvaan
- [map] korvataan polulla mapfileen
- [program] korvataan polulla käytettävään CGI-skriptiin
- [mapext] korvataan näytettävän kartta-alueen koordinaateilla
- [layername\_check] käytetään apuna määritettäessä mitkä karttakerrokset piirretään. Käyttäjälle on näkyvissä valintaruudut.
- [zoomdir\_-1|0|1\_check] käytetään apuna zoomaussuunnan määrittämisessä karttaa päivitettäessä. Käyttäjälle näkyy 3 vaihtoehtoa.

Template-tiedoston ensimmäisen rivin tulee sisältää ns. magic string, eli merkkijono 'mapserver template'. Kätevintä on tehdä siitä kommentti käytetyn ohjelmointikielen syntaksin mukaisesti (kuva 11).



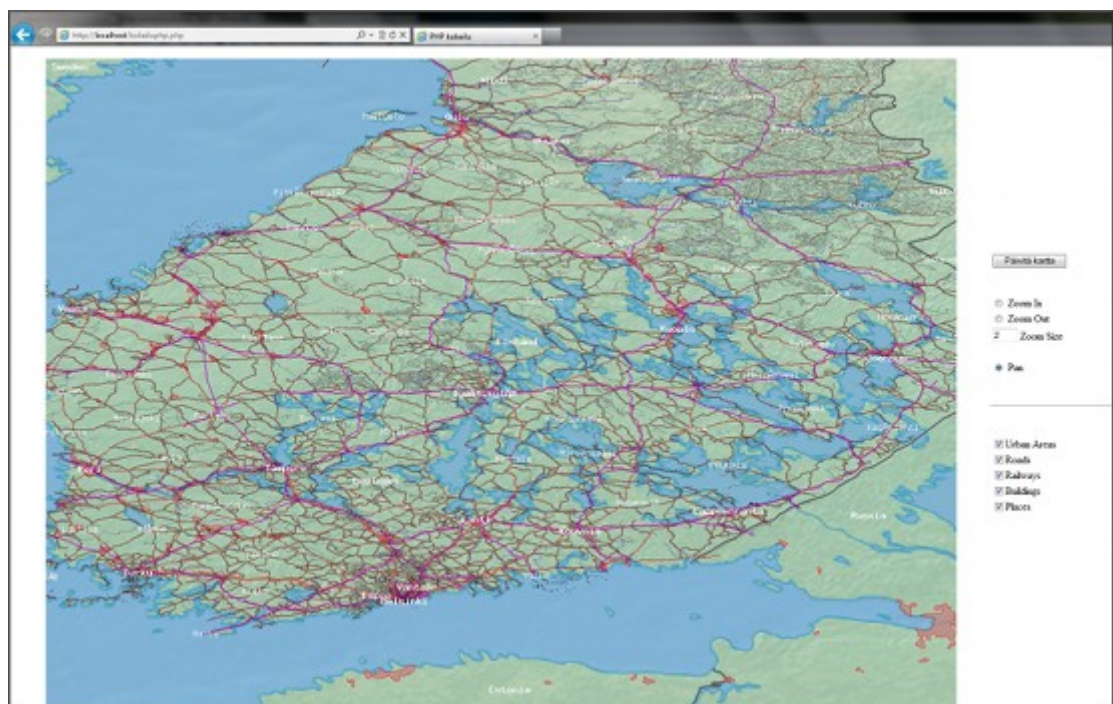
```

1 <!-- MapServer Template -->
2
3 <html>
4 <head><title>Kokeilu</title></head>
5 <body>
6 <div align="center">
7 <form name="the_form" method=GET action="[program]">
8

```

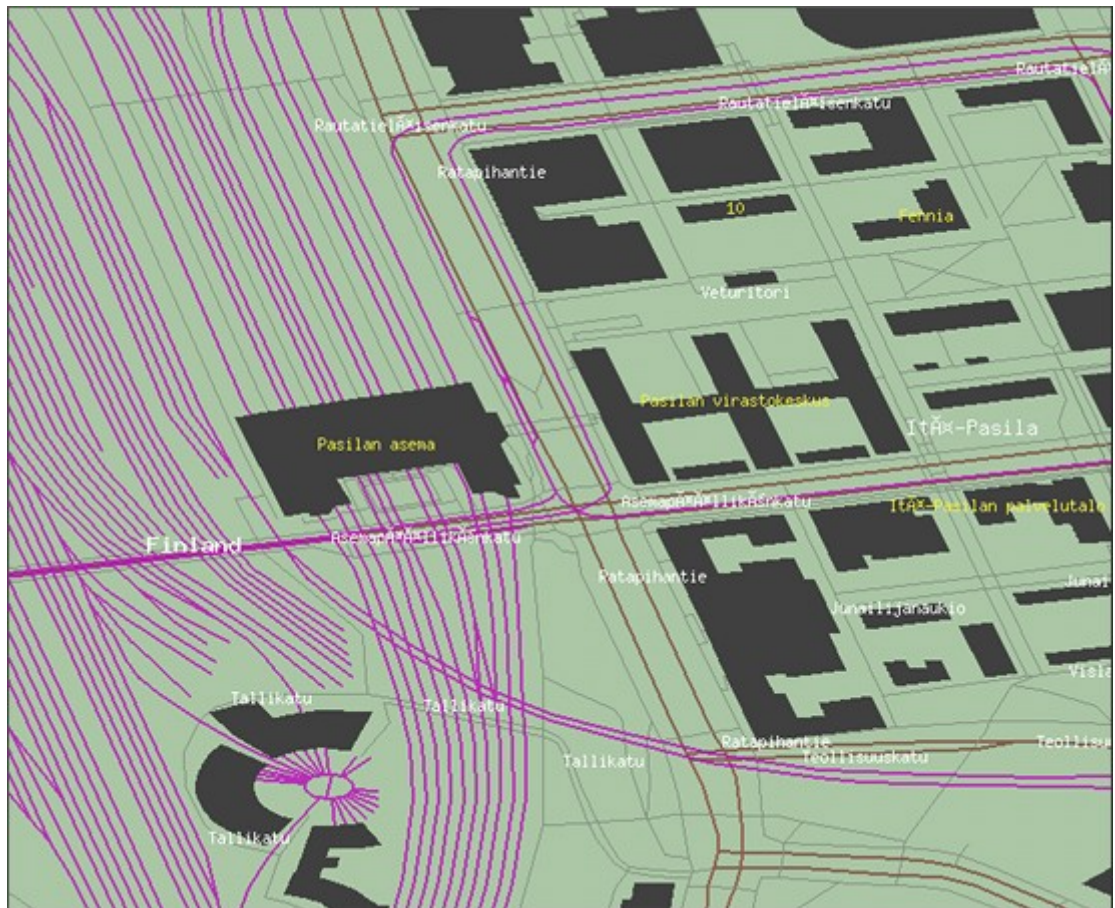
**KUVA 11. HTML-dokumenttiin lisätty ”magic string” -tunniste**

Jos MapServer ei löydä merkkijonoa, se antaa virheilmoituksen. Jos template-tiedosto on kunnossa, korvataan siellä olevat avainsanat ja web-palvelinohjelmisto palauttaa dokumentin käyttäjän selaimelle. Esimerkkitapauksessani selaimelle palautetaan suhteellisen suurikokoinen karttakuva, jonka oikealla puolella on muutama yksinkertainen kontrolli karttakuvan hallitsemiseksi (valinnat sille, mitkä kerrokset piirretään, zoom ja pan -kontrollit, painike karttakuvan päivittämiseksi) (kuva 12).



**KUVA 12. Renderöity esimerkkikartta template-tiedostoon palautettuna**

Poistamalla valinnat joiltakin karttakerroksilta ja joko painamalla ”Päivitä kartta” -painiketta tai klikkaamalla kuvan päällä lähetetään CGI-skriptille uudet ohjeet karttakuvan piirtämiseksi. Hetken kuluttua sivu päivittyy, ja sillä on uusi annettujen tietojen perusteella piirretty karttakuva. Pan-kontrollin ollessa valittuna kartan voi keskittää uuteen kohtaan klikkaamalla haluttua uutta keskipistettä. Zoomauskontrolleja käytettäessä kuva keskitetään klikkauskohtaan ja sitä samalla zoomataan valittuun suuntaan. Jos jompikumpi zoomaustyökalu on valittuna ”Päivitä kartta” -painiketta klikatessa, kuva zoomautuu, mutta keskikohta ei muutu. Näitä yksinkertaisia työkaluja käyttämällä päästään tarkastelemaan karttaa huomattavasti yksityiskohtaisemmalla tasolla (kuva 13).



**KUVA 13. Osa lähemmäs zoomatusta karttakuvasta**

Seuraavaksi muunnan juuri tehdyn sovelluksen käyttämään PHP MapScriptiä. Käytän sovelluksessa samaa mapfileä, joten sen puolesta lopputuloksessa ei ole mitään eroa. Aluksi luon muuttujia ja sijoitan niihin arvoja käyttötarkoituksen mukaan.

```
1 <?php
2 $skripti = "kokeiluphp.php";
3 $mapfile = "kokeilu.map";
4 $polku = "/ms4w/tmp/ms_tmp/";
5
```

**KUVA 14. Tärkeiden muuttujien arvoja**

Kuvan 14 muuttujat sisältävät oleellista tietoa sovelluksen toiminnan kannalta. \$skripti -muuttujaan on tallennettu sovelluksen sisältämän PHP-tiedoston nimi, jota tarvitaan lähetettäessä lomakkeelta tietoja eteenpäin. \$mapfile määrittää sovelluksen käyttämän mapfilen nimen. Se voisi sisältää hakemistopolun, mutta tässä tapauksessa mapfile sijaitsee samassa kansiossa PHP-tiedoston kanssa. \$polku määrittää sijainnin, johon MapServer sijoittaa piirtämänsä karttakuvat.

```
8 $zoomsize = 2;
9 $span = "CHECKED";
10 $zoomout = "";
11 $zoomin = "";
12 $urbanareas = "CHECKED";
13 $roads = "CHECKED";
14 $railways = "CHECKED";
15 $buildings = "CHECKED";
16 $places = "CHECKED";
```

**KUVA 15. Kontrollimuuttujien arvoja**

Muuttujat kuvassa 15 sisältävät tiedon kartan hallintaan käytettävien kontrollien oletustiloista. \$zoomsize määrittää, kuinka paljon MapServer zoomaa lähemmäs tai kauemmas käytettäessä zoomaustoimintoa. Zoomaustyökaluista panorointi on valittuna oletuksena. Rivien 12-16 muuttujat määrittävät kaikki karttakerrokset valituiksi.

```

20 $klikkaus_x = 700;
21 $klikkaus_y = 500;
22 $klikkauskohta = ms_newPointObj();
23 $value_vanha = ms_newRectObj();

```

**KUVA 16. Klikkauskohdan määrittäminen ja muodostimien käyttöä**

Joka kerta kun uusi karttakuva piirretään, sen keskikohta määrätään joko sen karttakuvan pisteen mukaan, jota käyttäjä on klikannut, tai määriteltyjen oletusarvojen mukaan. Muuttujat \$klikkaus\_x ja \$klikkaus\_y sisältävät nämä arvot (kuva 16). Koska mapfilessä olen määrittänyt piirrettävän karttakuvan kooksi 1400x1000 pikseliä, käytän muuttujille oletuksena arvoja 700 ja 500, jotka osoittavat täsmälleen kartan keskikohtaan. Tällöin karttakuva ei siirry, jos käyttäjä esimerkiksi poistaa jonkin karttakerroksen valinnan ja päivittää kartan painikkeella. Samoja arvoja käytetään myös karttakuvan ensimmäisellä piirto-kerralla. Muuttuja \$klikkauskohta rivillä 22 on ensimmäinen kosketus MapScriptiin. Siinä käytetään PointObj-luokan ms\_newPointObj-muodostinta. \$value\_vanha-muuttujan luomisessa käytetään muodostinta, joka kuuluu MapScriptin RectObj-luokkaan. Muodostimilla luotuja muuttujia kutsutaan olioiksi.

```

27 $value = array(20, 59, 32, 72);
28 $value_maksimi = ms_newRectObj();
29 $value_maksimi->setExtent(-90, -90, 90, 90);
30 $kartta = ms_newMapObj($mapfile);

```

**KUVA 17. Kartta-alueiden määrittelyitä**

Taulukkomuuttuja \$value määrittää aluksi piirrettävän kartta-alueen (kuva 17). Siihen sijoitetut arvot vastaavat mapfilessä olevia EXTENT-arvoja. Rivillä 28 luodaan muodostimella uusi RectObj-luokan olio \$value\_maksimi. Olion jäsenmuuttujiin sijoitetaan arvot käyttäen setExtent-metodia. Rivillä 30 luodaan MapObj-luokan olio \$kartta, johon luetaan tiedot muuttujan \$mapfile määrittämästä mapfilestä.



```

34     if (($_POST['img_x'] and $_POST['img_y'])
35         or $_POST['refresh'])
36     {
37         if ($_POST['refresh'])
38         {
39             $klikkaus_x = 700;
40             $klikkaus_y = 500;
41         }
42         else
43         {
44             $klikkaus_x = $_POST['img_x'];
45             $klikkaus_y = $_POST['img_y'];
46         }
47
48         $klikkauskohta->setXY($klikkaus_x,$klikkaus_y);

```

**KUVA 18. Tutkitaan sovelluksen tila ja asetetaan klikkauskohta**

Seuraavaksi tutkin, ollaanko karttaa piirtämässä ensimmäistä kertaa, vai käytetäänkö lomakkeen tietoja (kuva 18.). Jos lomakkeelta on tullut tietoja käsiteltäväksi POST-metodilla, käsittelen ne lomakkeen päivitystavan mukaan. Jos on käytetty ”Päivitä kartta” -painiketta, sijoitan \$klikkaus\_x- ja \$klikkaus\_y -muuttujiin oletusarvot 700 ja 500, jolloin kartan keskipiste ei muutu. Muussa tapauksessa sijoitan niihin lomakkeelta saadut arvot (img\_x ja img\_y). Olioon \$klikkauskohta sijoitan klikkauskohdan koordinaatit setXY-metodilla.

```

52     if ($_POST['layer'])
53     {
54         $layers = join(" ", $_POST['layer']);
55     }
56     else
57     {
58         $layers = "";
59     }

```

**KUVA 19. Sijoitetaan lomakkeen tiedot muuttuun**

Kuvassa 19 olevalla koodilla yhdistän lomakkeelta tulevan tiedon aktivoituista karttakerroksista \$layers-muuttuun PHP:n join-funktiolla. Muuttujaa \$layers käytetään määrittämään, mitkä karttakerrokset kuvaan piirretään.

```

61     $karttakerros = 0;
62
63     if (preg_match("/urbanareas/", $layers))
64     {
65         $urbanareas = "CHECKED";
66         $karttakerros = $kartta->getLayerByName('urbanareas');
67         $karttakerros->set('status', MS_ON);
68     }

```

**KUVA 20. Karttakerrosten tilan asettaminen**

Nyt \$layers-muuttujan sisältöä käsitellään PHP:n preg\_match -funktiolla, joka etsii ensimmäisen sääntöön sopivan merkkijonon (kuva 20). Tässä tapauksessa etsittävä teksti on karttakerroksen nimi. Jos etsitty merkkijono löytyy, lopettaa preg\_match toimintansa, palauttaen arvon 1 ja karttakerros määritetään piirrettäväksi (MS\_ON). Jos haettua merkkijonoa ei löydy, statukseksi laitetaan MS\_OFF ja "CHECKED"-arvon sijaan käytetään tyhjää. Sama toistetaan jokaiselle karttakerrokselle, jotka käyttäjä voi valita tai jättää valitsematta.

```

133     if ($_POST['extent'])
134     {
135         $value = split(" ", $_POST['extent']);
136     }
137
138     $kartta->setExtent($value[0], $value[1], $value[2], $value[3]);
139     $value_vanha->setExtent($value[0], $value[1], $value[2], $value[3]);
140     $zoomauskerroin = $_POST['zoom'] * $_POST['zsize'];
141     $zoomsize = abs($_POST['zsize']);

```

**KUVA 21. Näytettävä alue ja zoomaukseen liittyviä arvoja**

Kuvan 21 mukaisesti pilkon lomakkeelta tulleen tiedon näytettävästä alueesta osiin split-funktiolla ja sijoitan arvot taulukkomuuttujaan \$value. Nämä arvot sijoitan edelleen myös \$kartta- ja \$value\_vanha-olioihin setExtent-metodilla. Lasken myös arvon zoomauksessa käytettävälle \$zoomauskerroin-muuttujalle lomakkeen tietojen pohjalta. Tämän muuttujan arvon perusteella määritän, mikä zoomaustyökalu (zoom in, zoom out, pan) on valittuna. \$zoomsize-muuttujaan sijoitan itseisarvon lomakkeelta saadusta zsize-arvosta. Klikkauskohtaan koordinaattien muunto karttakoordinaateiksi tapahtuu sitä varten luodun PHP-funktion avulla.

```

165     $kartta->zoomPoint($zoomauskerroin,$klikkauskohta,
166     |     $kartta->width,$kartta->height,$alue_vanha,$alue_maksimi);
167     }
168

```

**KUVA 22. zoomPoint-metodi**

Olen sijoittanut kaikki zoomaamiseen tarvittavat muuttujat attribuuteiksi zoomPoint-metodille, joka hoitaa varsinaisen karttakuvan zoomauksen (kuva 22). Nyt perusmäärittelyt sekä zoom- ja pan-kontrollit ovat kunnossa. Rivillä 34 (kuva 18) alkanut sovelluksen tilan tutkiva ja sopivat toimenpiteet määräävä if-lause loppuu tähän.

```

172     $kartta_id = sprintf("%0.6d",rand(0,999999));
173     $karttakuva = "kokeilu".$kartta_id.".png";
174     $kartta_url="/ms_tmp/".$karttakuva;
175     $image=$kartta->draw();
176     $image->saveImage($polku.$karttakuva);

```

**KUVA 23. Karttakuvan piirto ja tallennus**

Määritän piirrettävälle karttakuvalle yksilöllisen nimen käyttäen apuna PHP:n satunnaislukufunktiota (kuva 23). \$kartta\_url-muuttujaan tulee sisällöksi renderöidyn karttakuvan sijainti tiedostonimen ja -pääteen kera selainta varten. Rivillä 175 annan MapServerille käskyn piirtää karttakuvan olion \$kartta perusteella. Piirretty kuva sijoitetaan muuttujaan \$image. Rivillä 176 muuttujaan \$image talletettu karttakuva tallennetaan tiedostoksi muuttujien mukaisella nimellä ja hakemistosijainnilla.

Tämän PHP-esimerkin tapauksessa web-lomakkeella ei ole käytössä hakasuluin ympäröityjä avainsanoja, vaan tarvittavat tiedot muodostetaan PHP:n avulla tulostamalla muuttujien arvoja tarvittaviin paikkoihin (kuva 24).



```

mg" type="image" src="<?php echo $kartta_url;?>"></td>

t name="refresh" value="Päivitä kartta" <br /><br /> </td></tr>
name="zoom" value="1" <?php echo $zoomin;?> Zoom In </td></tr>
name="zoom" value="-1" <?php echo $zoomout;?> Zoom Out </td></tr>
name="zsize" value="<?php echo $zoomsize;?>" size="2" Zoom Size </td>
.
name="zoom" value="0" <?php echo $pan;?> Pan </td></tr>

```

**KUVA 24. Lomaketietojen täydentämistä PHP:llä**

Käyttäjälle tämä PHP MapScriptillä luotu esimerkkisovellus vastaa ulkoasultaan ja toiminnoiltaan täysin aikaisemman esimerkin CGI-pohjaista sovellusta. Käyttäjän saapuessa käyttämään sovellusta, ensimmäinen karttakuva piirretään annettujen oletusarvojen perusteella. Tästä eteenpäin sovelluksen toiminta on käyttäjän valintojen varassa ja tarvittava tieto uusien karttakuvien piirtämiseksi saadaan HTML-lomakkeelta.

#### 4.5 Valmiita sovelluksia

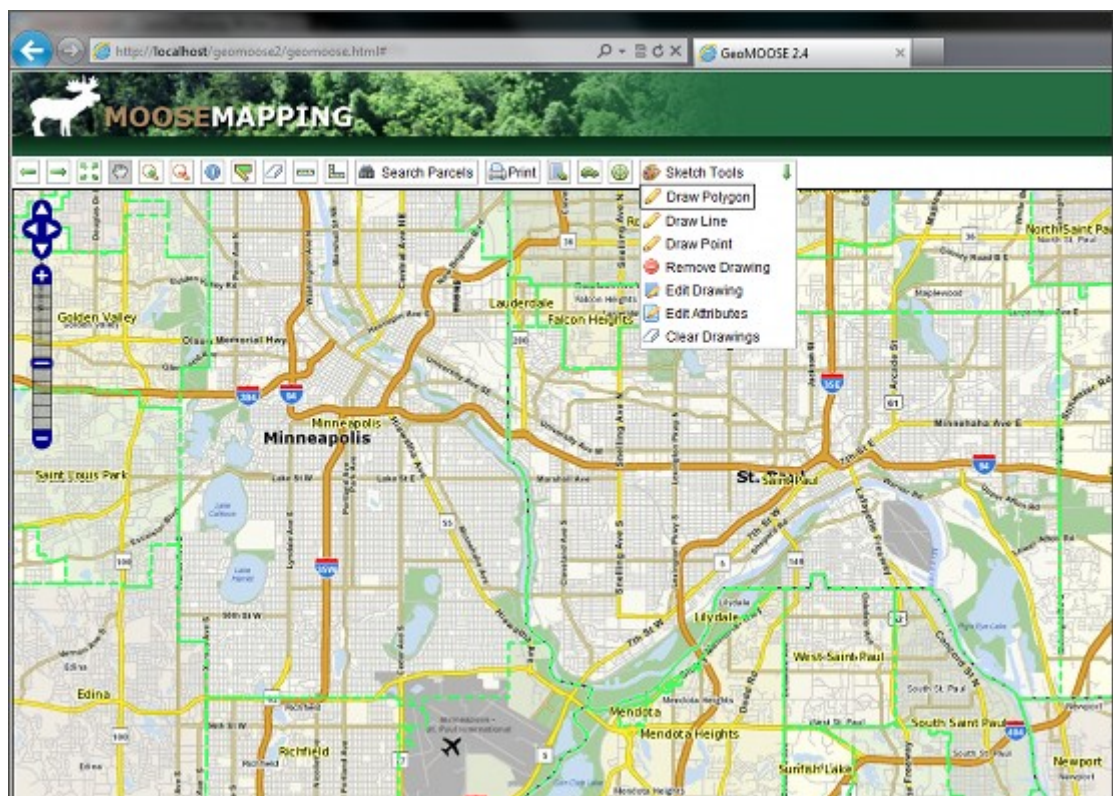
MapServeriä varten on laadittu useita open source -sovelluksia, jotka ovat vapaassa levityksessä. MapServer for Windowsia varten laaditut sovellukset asennetaan yleensä purkamalla kaikki ohjelman tarvitsevat tiedostot sisältävä zip-paketti. Nämä paketit puretaan yleensä ms4w-kansion sisälle. Asennetut sovellukset tulevat automaattisesti käyttöön, kun Apache käynnistetään seuraavan kerran uudelleen. Tällä hetkellä saatavilla olevia MapServer for Windowsia varten valmiiksi konfiguroituja sovelluksia [23]:

- ArcIMS Emulator - toteuttaa ESRI ArcIMS-tyylisen käyttöliittymän MapServeriin, jolloin sitä voidaan käyttää ESRI:n client-sovelluksilla ja kaikilla muilla ArcIMS ArcXML-rajapintaa hyödyntävillä sovelluksilla.
- Chameleon - suurelta osin muokattavissa oleva ympäristö kartta- ja paikkatietosovellusten kehittämiseen, jonka ytimenä toimii MapServer.

- dBox MapServer - kokoelma DHTML-pohjaisia kirjastoja interaktiivisten karttasovellusten laatimiseksi Internetiin.
- FIST - Flexible Internet Spatial Template - avoimen lähdekoodin PHP-pohjainen sovellus, jota käytetään karttojen esittämiseen Internetissä.
- GeoMOOSE - javascript-pohjainen, OpenLayers-kirjastoon ja MapServeriin pohjautuva sovellus kartografisen datan esittämiseen.
- Ka-Map - avoimen lähdekoodin projekti, jonka päämäärä on tuottaa javascript-rajapinta interaktiivisten web-karttasovellusten kehittämiseen.
- MapBender - Open Source Geospatial Foundationin projekti OGC:n standardien mukaisten karttapalveluiden käyttämiseen. Toteutuksessa käytetty Javascriptiä, PHP:tä ja XML:ää.
- MapServer Itasca Demo Application - demosovellus, jolla voidaan testata MapServerin toiminta asennuksen jälkeen ja tutkia palvelimen peruskäyttöä.
- OGC Workshop - tämä ei varsinaisesti ole MapServeriä varten laadittu sovellus, vaan yhdistelmä ohjeistuksia, testejä ja harjoituksia OGC:n standardien toteuttamisesta MapServerillä.
- OpenLayers - avoimen lähdekoodin javascript-kirjasto karttadatan esittämiseen moderneissa Internet-selaimissa.
- GMap Sample PHP/MapScript Application - PHP/MapScript esimerkkiohjelma
- phpPgAdmin - web-pohjainen hallintatyökalu PostgreSQL-tietokantaa varten.
- p.mapper - helpottaa PHP/MapScript-pohjaisen MapServer-sovelluksen kehitystä ja käyttöönottoa tarjoamalla monia uusia funktioita hyödynnettäväksi.

GeoMOOSE on HTML- ja javascript-pohjainen sovellus kartografisen datan esittämiseen, ja se perustuu kahteen muuhun open source-projektiin: MapServeriin ja OpenLayers-javascript-kirjastoon. GeoMOOSEa on mahdollista ajaa normaalillakin web-palvelimella. Sovellukseen kuuluu useita PHP:llä ohjelmoituja työkaluja.

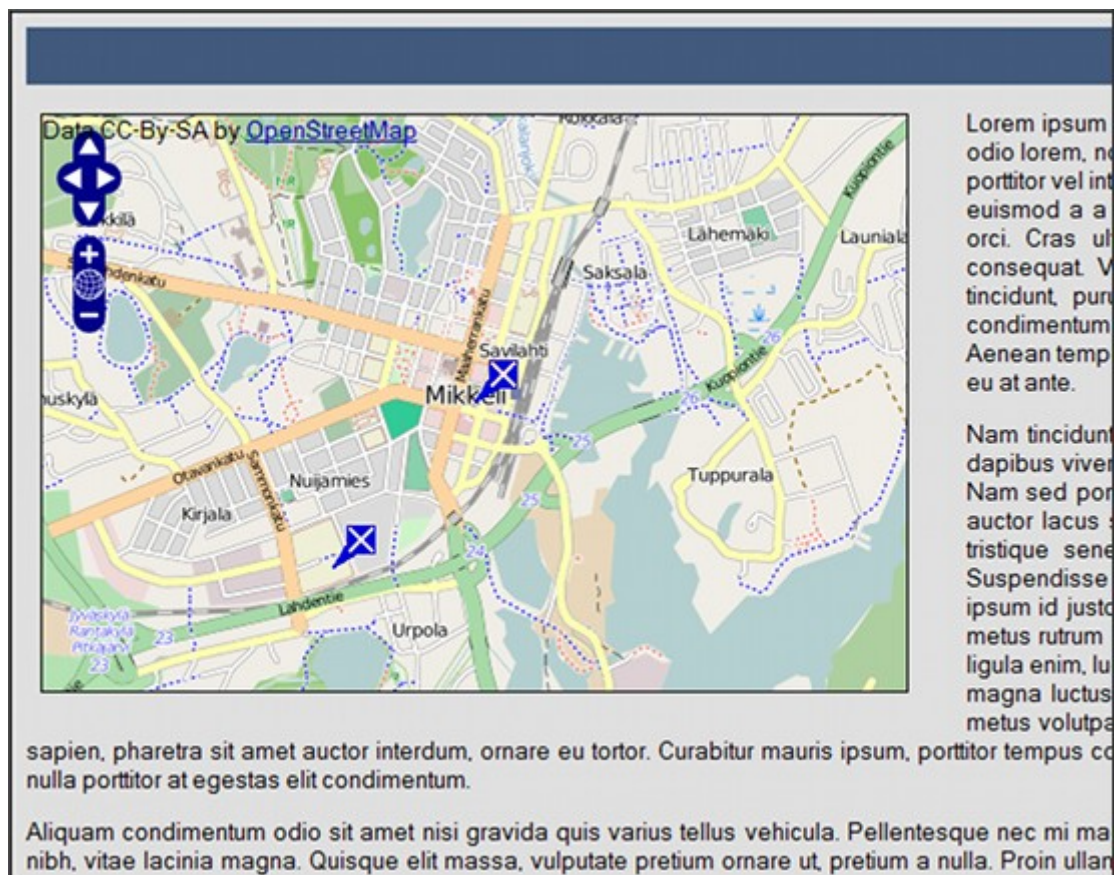
Asennan GeoMOOSE 2.4RC1 for MS4W -sovelluksen yksinkertaisesti purkamalla lataamani zip-paketin sen levyaseman juureen, mihin olen sijoittanut MapServer for Windowsin. Paketin purkamisen jälkeen varmistetaan, että Apache on käynnissä (jos Apache oli käynnissä sovelluksen asentamisen ajan, on tarpeen käynnistää Apache uudelleen), minkä jälkeen voidaan navigoida Internet-selain osoitteeseen <http://localhost/>. Jos kaikki toimii oletetusti, saadaan näytölle MapServerin infisivu, jossa Applications-kohdan alla on mainittuna juuri asentamamme GeoMOOSE ja linkit itse sovellukseen, sovelluksen verkkosivustolle ja sen dokumentaatioon.



**KUVA 25. GeoMOOSE-sovellus toiminnassa esimerkkikarttoineen**

GeoMOOSE on jo varsin pitkälle kehitetty paikkatietosovellus. Siihen on laadittu monipuoliset työkalut niin zoomausta, mittauksia kuin hakujakin varten (kuva 25). Lisäksi sovellukseen on integroitu mahdollisuus valita kartalta kohta, joka sitten näytetään Googlen Streetview-palvelussa tai Microsoftin Bing Mapsin Bird's Eye view-palvelussa. Sovelluksen asetustiedostot löytyvät `ms4w/apps/geomoose2/conf/` hakemistosta `.ini-` ja `.xml-`muodoissa.

OpenLayers puolestaan on avoimen lähdekoodin AJAX-kirjasto (Asynchronous JavaScript And XML) geografisen datan käsittelyyn, ja sen tarkoituksena on mahdollistaa karttadatan näyttäminen moderneissa Internet-selaimissa. OpenLayers on Open Source Geospatial Foundationin projekti, ja se tukee OGC:n standardeja (esimerkiksi WMS ja WFS). Kirjastoa apuna käyttäen on helppo lisätä dynaaminen kartta mille tahansa web-sivulle. Monissa tapauksissa tällainen pikkukartta onkin todella kätevä, eikä resursseja tarvitse uhrata raskaisiin kokonaisuuksiin. Näkemällä hieman enemmän vaivaa kartalle voi lisätä haluamiaan pisteitä, joita osoittamaan voi asettaa jonkin symbolin.



**KUVA 26. Web-sivulle upotettu kartta OpenLayers-kirjastoa käyttäen**

Laadin pienen esimerkkisivun, jolle sijoitin 480x320 pikselin kokoisen karttaobjektin (kuva 26). Sivulle on linkitetty OpenLayers.js -kirjasto ja sen lisäksi on laadittu pieni javaskripti, joka käyttää kirjaston toimintoja hyödykseen. Skripti koostuu funktiosta, joka suoritetaan heti kun sivu on latautunut kokonaisuudessaan. Funktiossa määritellään karttaobjekti, kaksi karttakerrosta (normaali kartta OpenStreetMapin dataan perustuen ja omat merkinnät sisältävä kerros), kartan keskipiste, karttaprojektion muunnos ja zoomaustaso. Omien karttamerkintöjen tiedot haetaan

tekstitiedostosta, jossa on määritelty merkinnän kohta kartalla, merkinnän kuvaus sekä sitä varten käytettävä ikoni. Klikkaamalla kartalla olevaa ikonia käyttäjä saa avattua merkinnälle kirjoitetun kuvauksen pieneen laatikkoon, joka voidaan sulkea klikkaamalla ikonia uudelleen.

## 5 POHDINTA

Opinnäytetyöni tavoitteena oli saattaa avoimen lähdekoodin MapServer-karttapalvelinohjelmiston Windows-versio käyttökuntoon ja selvittää sen toimintaa ja käyttöä. Mielestäni saavutin asetetut tavoitteet ja sain havainnollistettua ohjelmiston käyttöä spatiaalisen datan esittämisessä laatien yksinkertaisen esimerkkisovelluksen käyttäen MapServerin CGI-rajapintaa ja PHP MapScriptiä. Tämän lisäksi päätin selvittää, mitä valmiita paikkatietosovelluksia MapServeriä varten on laadittu, ja asensin niistä yhden lyhyttä tutustumiskäyttöä varten. Useimmissa tapauksissa näin raskaiden ja monimutkaisten sovellusten käyttö ei kuitenkaan ole vaatimusten tai tarkoitusperien mukaista, joten rakensin esimerkin web-sivulle upotetusta karttaikkunasta OpenLayers-javascript-kirjastoa hyödyntäen. Samaa avoimen lähdekoodin javascript-kirjastoa on myös käytetty kokeilemassani GeoMOOSE-sovelluksessa.

Jatkokehityksen kannalta olisi mielenkiintoista nähdä, millaisia sovelluksia on mahdollista laatia hyödyntäen MapServerin kaikkia ominaisuuksia laadukkaiden paikkatietoaineistojen pohjalta. MapScriptiä ja useita huolellisesti koostettuja mapfilejä käyttämällä voidaan varmasti saada aikaan mielenkiintoisempi, dynaamisempi ja visuaalisesti miellyttävämpi sovellus. Tutustumista varten asentamaani GeoMOOSE-sovellus on mielestäni selkeä merkki siitä, että MapServeriä voidaan käyttää alustana kehittyneille ja monipuolisille paikkatietosovelluksille.

Web-tekniikoiden jatkuva kehitys mahdollistaa uusia mielenkiintoisia asioita, jotka luultavasti tulevat jollakin tapaa näkymään myös karttapalveluissa sekä muussa paikkatiedon hyödyntämisessä Internetissä. Näihin uusiin uusiin mielenkiintoisiin tekniikoihin kuuluu mm. HTML5 ja WebGL, joiden canvas-elementtiä voisi

hyödyntää tavalla tai toisella karttojen esittämisessä. Myös Internetin mahdollistama vuorovaikutteisuus ja suuren suosion saaneet sosiaalisen median palvelut voisivat toimia jonkinlaisena hyötyjänä paikkatiedon tarjoamista mahdollisuuksista, erityisesti pitäen silmällä nykyisiä älypuhelimia ja niiden integraatiota edellä mainittuihin palveluihin.

Internetin karttapalveluista Google Maps on tunnetuin ja suosituin. Google palvelun rajapinta on nyky muodossaan monipuolinen, joten tietyssä määrin voidaankin kyseenalaistaa MapServerin tarpeellisuus. Vaikka tällä hetkellä suurimassa osassa Googlen ilmaista palvelua ei ole rajoituksia käyttömäärien suhteen, niitä saatetaan kuitenkin asettaa. Tämän lisäksi Google pidättää itsellään oikeuden lisätä mainoksia ilmaispalveluun. MapServer on ympäristönä erilainen kuin Google Maps, ja mahdollisuuksia kustomointiin on lähes rajattomasti. Monessa muodossa olevien omien kartta-aineistojen hyödyntäminen on myös kätevää MapServeriä käyttäen. Järkevintä onkin miettiä tarkasti, mitä toteutettavalta systeemiltä vaaditaan, kuinka paljon resursseja on käytettävissä ja millaiselle karttadatalle on tarvetta. Nämä asiat viime kädessä määräävät sen, onko MapServerin käyttö perusteltua. Joka tapauksessa on mielenkiintoista nähdä, kuinka esimerkiksi OpenStreetMap-projekti tulee kehittymään jatkossa.

**LÄHTEET**

- [1] Sanastokeskus TSK ry. Geoinformatiikan sanasto. 2005.  
Pdf-dokumentti.  
<http://www.tsk.fi/tiedostot/pdf/GeoinformatiikanSanasto.pdf>
- [2] Tekniikan sanastokeskus, Paikannussanasto. 2002.  
Pdf-dokumentti.  
<http://www.tsk.fi/fi/info/paikannussanasto.pdf>
- [3] Nuora, Arto. Paikkatietojen käyttö kunnissa.  
Suomen Kuntaliitto, Kuntaliiton painatuskeskus. Helsinki. 1995.  
ISBN 951-598-305-3.
- [4] Hannus, Esa. Hajautettu paikkatietojärjestelmä oppilaitoskäytössä:  
Internet-GIS. Opinnäytetyö, Mikkelin ammattikorkeakoulu. 2007.  
ISBN 978-951-588-225-7. ISSN 1455-3678.
- [5] Löytönen Markku, Toivonen Tuuli, Kankaanrinta Ilta-Kanerva.  
Globus GIS: Paikkatietojärjestelmä. 2003.  
WS Bookwell Oy, Porvoo. ISBN 951-0-26089-4.
- [6] Rainio, Antti. Paikkatietoa selaimella. 3.5.2010  
WWW-dokumentti.  
<http://www.maanmittauslaitos.fi/tiedotteet/2010/05/paikkatietoa-selaimella>.  
Viitattu 6.2.2011
- [7] Wikipedia. Geographic information system.  
WWW-dokumentti.  
[http://en.wikipedia.org/wiki/Geographic\\_information\\_system](http://en.wikipedia.org/wiki/Geographic_information_system)  
Viitattu 12.1.2011
- [8] Peng, Zhong-Ren & Tsou, Ming-Hsiang. Internet GIS: Distributed  
Geographic Information Services for the Internet and Wireless  
Networks. 2003.  
John Wiley & Sons, Inc. New Jersey, U.S.A. ISBN 0-471-35923-8.



- [9] Daratech, Inc. GIS/Geospatial Industry Worldwide Growth Slows to 1% in 2009. 20.8.2009.  
WWW-dokumentti.  
<http://www.directionsmag.com/pressreleases/gisgeospatial-industry-worldwide-growth-slows-to-1-in-2009/119103>  
Viitattu 5.1.2011.
- [10] Wikipedia. Esri. 8.4.2011  
WWW-dokumentti  
<http://en.wikipedia.org/wiki/Esri>  
Viitattu 13.4.2011
- [11] ESRI. Products.  
WWW-dokumentti  
<http://www.esri.com/products/index.html>  
Viitattu 13.4.2011
- [12] GRASS Development Team. GRASS: Introduction. 4.8.2010  
WWW-dokumentti.  
<http://grass.osgeo.org/intro/general.php>  
Viitattu 23.4.2011.
- [13] Regents of the University of Minnesota. Welcome to MapServer. 2010.  
WWW-dokumentti.  
<http://www.mapserver.org/>.  
Viitattu 19.11.2010.
- [14] Artem Pavlenko. Mapnik C++/Python GIS Toolkit | Welcome. 2010.  
WWW-dokumentti.  
<http://www.mapnik.org/>  
Viitattu 2.5.2011.
- [15] Open Geospatial Consortium. Welcome to the OGC Website.  
<http://www.opengeospatial.org/>  
WWW-dokumentti  
Viitattu 17.3.2011



- [16] Google. Google Maps API Family. 2011.  
WWW-dokumentti.  
<http://code.google.com/intl/fi-FI/apis/maps/index.html>  
Viitattu 2.4.2011
- [17] OpenStreetMap Wiki. API. 17.12.2010.  
WWW-dokumentti.  
<http://wiki.openstreetmap.org/wiki/API>  
Viitattu 2.4.2011
- [18] Wikipedia. Shapefile. 3.4.2011.  
WWW-dokumentti  
<http://en.wikipedia.org/wiki/Shapefile>  
Viitattu 28.4.2011.
- [19] Ruth, Mike. GeoTIFF FAQ Version 2.4. Helmikuu 2011.  
WWW-dokumentti  
<http://remotesensing.org/geotiff/faq.html>  
Viitattu 23.3.2011
- [20] Open Source Initiative. The MIT License. 2010.  
WWW-dokumentti.  
<http://www.opensource.org/licenses/mit-license.php>.  
Viitattu 22.11.2010.
- [21] Kropla, Bill. Beginning MapServer: Open Source GIS Development. 2005.  
Pdf-dokumentti.  
<http://geografi.ums.ac.id/ebook/GIS/WebGIS/B.Kropla%20-%20Beginning%20MapServer%20-%20Open%20Source%20GIS%20Development.%202005.pdf>
- [22] Netcraft. March 2010 Web Server Survey. 17.3.2010.  
WWW-dokumentti.  
[http://news.netcraft.com/archives/2010/03/17/march\\_2010\\_web\\_server\\_survey.html](http://news.netcraft.com/archives/2010/03/17/march_2010_web_server_survey.html)  
Viitattu 31.1.2011.

- [23] McKenna Jeff, Gateway Geomatics. MapServer for Windows (MS4W).  
20.4.2010.  
README\_INSTALL.html  
MS4W:n mukana tuleva readme-tiedosto.
- [24] The MapServer Team. MapServer 5.6.6 Documentation. 5.2.2011.  
Pdf-dokumentti.  
<http://mapserver.org/MapServer.pdf>

# LIITE 1. CGI-esimerkin käynnistävä HTML-lomake

```
<html>
<head> <title>Kokeilu</title></head>
<body>
  <div align="center">
    <form method=POST action=" ../cgi-bin/mapserv.exe">
      <input type="submit" value="Käynnistä">
      <input type="hidden" name="program"
        value=" ../cgi-bin/mapserv.exe">
      <input type="hidden" name="map"
        value="c:/ms4w/Apache/htdocs/kokeilu.map">
      <input type="hidden" name="zoomsize" size=2 value=2>
      <input type="hidden" name="layers"
        value="urbanareas roads railways buildings
        places">
    </form>
  </div>
</body>
</html>
```

## LIITE 2. CGI-esimerkin template-tiedosto

```
<!-- MapServer Template -->
<html>
<head><title>Kokeilu</title></head>
<body>
  <div align="center">
    <form name="the_form" method=GET action="[program]">
      <table width="1800px">
        <tr>
          <td width="1450px">
            <input name="img" type="image"
              src="[img]" width=1400 height=1000></td>

          <td width="350px">
            <table>
              <tr><td> <input type="submit"
                value="Päivitä kartta"> <br /></td></tr>

              <tr><td> <input type="radio" name="zoomdir"
                value=1 [zoomdir_1_check]> Zoom In </td></tr>

              <tr><td> <input type="radio" name="zoomdir"
                value=-1 [zoomdir_-1_check]> Zoom Out </td></tr>

              <tr><td> <input type="text" name="zoomsize" size=2
                value=[zoomsize]> Zoom Size </td></tr>

              <tr><td> <input type="radio" name="zoomdir"
                value=0 [zoomdir_0_check]> Pan </td></tr>
            </table>

            <br /> <hr /> <br />

            <table>
              <tr><td> <input type="checkbox" name="layer"
                value="urbanareas" [urbanareas_check]> Urban areas </td></tr>

              <tr><td> <input type="checkbox" name="layer"
                value="roads" [roads_check]> Roads </td></tr>

              <tr><td> <input type="checkbox" name="layer"
                value="railways" [railways_check]> Railways </td></tr>

              <tr><td> <input type="checkbox" name="layer"
                value="buildings" [buildings_check]> Buildings </td></tr>

              <tr><td> <input type="checkbox" name="layer"
                value="places" [places_check]> Places </td></tr>
            </table>

          </td>
        </tr>
      </table>

      <input type="hidden" name="imgxy" value="700 500">
      <input type="hidden" name="imgext" value="[mapext]">
      <input type="hidden" name="map" value="[map]">
      <input type="hidden" name="program" value="[program]">
    </form>
  </div>
</body>
</html>
```

```
<?php
//PHP-tiedoston nimi, polkuasetukset mapfilelle ja
//renderöidyille kuville
$skripti = "kokeiluphp.php";
$mapfile = "kokeilu.map";
$polku = "/ms4w/tmp/ms_tmp/";

//Navigointikontrollien oletusasetukset
$zoomsize = 2;
$span = "CHECKED";
$zoomout = "";
$zoomin = "";

//Oletuksena näytettävät karttakerrokset joihin
//käyttäjä voi vaikuttaa
$urbanareas = "CHECKED";
$roads = "CHECKED";
$railways = "CHECKED";
$buildings = "CHECKED";
$places = "CHECKED";

//Oletettu klikkauskohta "päivitä" -nappulaa
//käytettäessä (= keskellä kuvaa)
$klikkaus_x = 700;
$klikkaus_y = 500;
$klikkauskohta = ms_newPointObj();
$alue_vanha = ms_newRectObj();

//Aluemäärittelyitä
$alue = array(20, 59, 32, 72); //Mapfilen extent
$alue_maksimi = ms_newRectObj();
$alue_maksimi->setExtent(-90, -90, 90, 90);

//Luetaan mapfile ja luodaan kartta sen perusteella
$kartta = ms_newMapObj($mapfile);

//Tutkitaan, käytetäänkö lomakkeen tietoja vai
//ollaanko luomassa ensimmäistä karttaa
if (($_POST['img_x'] and $_POST['img_y']) or $_POST['refresh']) {
    //Päivitä-napin käyttö, simuloidaan painalluksen
    //koordinaatit (kuvan keskikohta)
    if ($_POST['refresh']) {
        $klikkaus_x = 700;
        $klikkaus_y = 500;
    }
    //Karttakuvan klikkaus, käytetään oikeita koordinaatteja
    else {
        $klikkaus_x = $_POST['img_x'];
        $klikkaus_y = $_POST['img_y'];
    }

    //Sijoitetaan klikkauskohta muuttujiin setXY-metodilla
    $klikkauskohta->setXY($klikkaus_x,$klikkaus_y);

    //Karttakerrokset
    if ($_POST['layer']) {
        $layers = join(" ", $_POST['layer']);
    }
    else {
        $layers = "";
    }

    $karttakerros = 0;

    if (preg_match("/urbanareas/", $layers)) {
        $urbanareas = "CHECKED";
        $karttakerros = $kartta->getLayerByName('urbanareas');
        $karttakerros->set('status', MS_ON);
    }
}
```

```
else {
    $urbanareas = "";
    $karttakerros = $kartta->getLayerByName('urbanareas');
    $karttakerros->set('status', MS_OFF);
}
if (preg_match("/roads/", $layers)) {
    $roads = "CHECKED";
    $karttakerros = $kartta->getLayerByName('roads');
    $karttakerros->set('status', MS_ON);
}
else {
    $roads = "";
    $karttakerros = $kartta->getLayerByName('roads');
    $karttakerros->set('status', MS_OFF);
}
if (preg_match("/railways/", $layers)) {
    $railways = "CHECKED";
    $karttakerros = $kartta->getLayerByName('railways');
    $karttakerros->set('status', MS_ON);
}
else {
    $railways = "";
    $karttakerros = $kartta->getLayerByName('railways');
    $karttakerros->set('status', MS_OFF);
}
if (preg_match("/buildings/", $layers)) {
    $buildings = "CHECKED";
    $karttakerros = $kartta->getLayerByName('buildings');
    $karttakerros->set('status', MS_ON);
}
else {
    $buildings = "";
    $karttakerros = $kartta->getLayerByName('buildings');
    $karttakerros->set('status', MS_OFF);
}
if (preg_match("/places/", $layers)) {
    $places = "CHECKED";
    $karttakerros = $kartta->getLayerByName('places');
    $karttakerros->set('status', MS_ON);
}
else {
    $places = "";
    $karttakerros = $kartta->getLayerByName('places');
    $karttakerros->set('status', MS_OFF);
}

//Näytettävän kartan alue
if ($_POST['extent']) {
    $value = split(" ", $_POST['extent']);
}

//Asetetaan kartan alue lomakkeen tietojen mukaan
$kartta->setExtent($value[0], $value[1], $value[2], $value[3]);

//Pistetään vanha alue talteen (rectObj), tätä
//tarvitaan zoomaamiseen
$value_vanha->setExtent($value[0], $value[1], $value[2], $value[3]);

//lasketaan zoomauskerroin zoomPoint-metodille
//0 = pan, <0 = zoom out, >0 = zoom in
$zoomauskerroin = $_POST['zoom']*$_POST['zsize'];

//Asetetaan valintaruutujen tila zoomaussuunnan mukaisesti
if ($zoomauskerroin == 0) {
    $zoomauskerroin = 1;
    $pan = "CHECKED";
    $zoomout = "";
    $zoomin = "";
}
elseif ($zoomauskerroin < 0) {
```

## LITE 3(3). PHP MapScript -esimerkin koodi

```
        $pan = "";
        $zoomout = "CHECKED";
        $zoomin = "";
    }
    else {
        $pan = "";
        $zoomout = "";
        $zoomin = "CHECKED";
    }

    $zoomsize = abs($_POST['zsize']);

    //Zoomataan
    $kartta->zoomPoint($zoomauskerroin,$klikkauskohta,
    $kartta->width,$kartta->height,$alue_vanha,$alue_maksimi);
}
//Zoom/pan ja layerit nyt kunnossa

//Luodaan yksilöllinen tiedostonimi karttakuvalle
$kartta_id = sprintf("%0.6d",rand(0,999999));
$karttakuva = "kokeilu".$kartta_id.".png";
$kartta_url = "/ms_tmp/".$karttakuva;

//Piiirretään ja tallennetaan karttakuva
$image = $kartta->draw();
$image->saveImage($polku.$karttakuva);

//Tallennetaan zoom/pan jälkeinen alue lomakemuuttujaan
//arvot erotettu välilyönnein
$alue_uusi = sprintf("%3.6f",$kartta->extent->minx)." "
            .sprintf("%3.6f",$kartta->extent->miny)." "
            .sprintf("%3.6f",$kartta->extent->maxx)." "
            .sprintf("%3.6f",$kartta->extent->maxy);

//Klikkauskoordinaattien muunto karttakoordinaateiksi
list($x,$y) = img2map($kartta->width,$kartta->height,
$klikkauskohta,$alue_vanha);

//Muunto kuvasta karttakoordinaateiksi
function img2map($width,$height,$point,$ext)
{
    $minx = $ext->minx;
    $miny = $ext->miny;
    $maxx = $ext->maxx;
    $maxy = $ext->maxy;

    if ($point->x && $point->y){
        $x = $point->x;
        $y = $point->y;

        $dpp_x = ($maxx-$minx)/$width;
        $dpp_y = ($maxy-$miny)/$height;

        $x = $minx + $dpp_x*$x;
        $y = $maxy - $dpp_y*$y;
    }

    $pt[0] = $x;
    $pt[1] = $y;
    return $pt;
}

?>

<html>
<head><title>PHP kokeilu</title></head>
<body>
<div align="center">
    <form method=POST action="<?php echo $skripti;?>">
```

## LIITE 3(4). PHP MapScript -esimerkin koodi

```
<table width="1800px">
<tr>
<td width="1450px">
  <input name="img" type="image"
    src="<?php echo $kartta_url;?>"></td>

<td width="350px">
  <table>
<tr><td> <input type="submit" name="refresh"
  value="Päivitä kartta"> </td></tr>

<tr><td> <input type="radio" name="zoom"
  value="1" <?php echo $zoomin;?>> Zoom In </td></tr>

<tr><td> <input type="radio" name="zoom"
  value="-1" <?php echo $zoomout;?>> Zoom Out </td></tr>

<tr><td> <input type="text" name="zsize"
  value="<?php echo $zoomsize;?>" size="2"> Zoom Size </td></tr>

<tr><td> <input type="radio" name="zoom"
  value="0" <?php echo $pan;?>> Pan </td></tr>
</table>

<br /> <hr /> <br />

<table>
<tr><td> <input type="checkbox"
  name="layer[]" value="urbanareas"
  <?php echo $urbanareas;?>>Urban Areas</td></tr>

<tr><td> <input type="checkbox"
  name="layer[]" value="roads"
  <?php echo $roads;?>>Roads</td></tr>

<tr><td> <input type="checkbox"
  name="layer[]" value="railways"
  <?php echo $railways;?>>Railways</td></tr>

<tr><td> <input type="checkbox"
  name="layer[]" value="buildings"
  <?php echo $buildings;?>>Buildings</td></tr>

<tr><td> <input type="checkbox"
  name="layer[]" value="places"
  <?php echo $places;?>>Places</td></tr>

<tr><td> <input type="hidden"
  name="extent" value="<?php echo $alue_uusi;?>"> </td></tr>
</table>
</td>
</tr>
</table>
</form>
</div>
</body>
</html>
```