

LAPPEENRANNAN TEKNILLINEN YLIOPISTO
TIETOTEKNIIKAN OSASTO

**ASIAKASLÄHTÖISEN PAIKKATIEDON HYÖDYNTÄMINEN
AUTOMAATTISESSA KARTTATUOTANTOJÄRJESTELMÄSSÄ**

Diplomityön aihe on hyväksytty Tietotekniikan osaston osastoneuvostossa 13.12.2006.

Työn tarkastajina toimivat professori Heikki Kälviäinen ja FT Pellervo Kokkonen.

Työn ohjaajana toimii Jyrki Pulkkinen.

Sulkavalla 20.11.2007

Pekka Repo

Reposuontie 191

58700 SULKAVA

puh. +358 40 7273 986

pekka.repo@karttaikkuna.fi

TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto
Tietotekniikan osasto

Pekka Repo

Asiakaslähtöisen paikkatiedon hyödyntäminen automaattisessa karttatuotantojärjestelmässä

Diplomityö

2007

40 sivua, 19 kuvaa ja 7 taulukkoa

Tarkastajat: Professori Heikki Kälviäinen
FT Pellervo Kokkonen

Hakusanat: kartta, paikkatieto, vektorigrafiikka
Keywords: map, geographic information, vector graphics

Tässä työssä on kehitetty järjestelmä, jonka avulla asiakas voi lisätä omia kohteitaan kartan päälle. Tuotantoketju kohteiden lisäämisen jälkeen on automaattinen, joten kohteiden lisääminen karttaan ei aiheuta ylimääräistä työtä karttaa tulostettaessa.

Kohteiden lisäämistä varten kehitettiin yksinkertaiset piirtotyökalut sekä mahdollisuus tuoda geometriatietoa joistain paikkatietotiedoista. Kohteet myös talletetaan paikkatietokantaan myöhempää tarkastelua ja muokkausta varten. Lisätyt kohteet syötetään karttatuotantojärjestelmään SVG-tiedostona.

ABSTRACT

Lappeenranta University of Technology
Department of Information Technology

Pekka Repo

Using customer oriented geographic information in automatic map production system

Thesis for the Degree of Master of Science in Technology

2007

40 pages, 19 figures and 7 tables

Examiners: Professor Heikki Kälviäinen
PhD Pellervo Kokkonen

Keywords: map, geographic information, vector graphics

In this work a system to add customers own geographic information on a map was developed. After the objects are added the map is produced completely automatically. Adding the objects does not add any extra work when maps are printed.

Simple drawing tools to add the objects were implemented. Tools to import geometry data from some geographic information files were also added. The objects are stored in a spatial database so they can be viewed and edited later. The added objects are passed to the map production system as a SVG-file.

ALKUSANAT

Tämä diplomityö on tehty Karttaikkuna Oy:lle, joka on sulkaavalainen, karttoihin ja paikkatietoon erikoistunut yritys. Työ on osa suurempaa projektia, jonka tarkoituksena on kehittää yrityksen toimintaa ja palveluja.

Haluan kiittää Karttaikkuna Oy:tä, erityisesti toimitusjohtaja Kristian Liljeströmiä ja työn ohjaajaa Jyrki Pulkista, tuesta ja ohjeista työn aikana sekä diplomityön tarkastajia Heikki Kälviäistä ja Pellervo Kokkosta.

SISÄLLYSLUETTELO

LYHENNE- JA SYMBOLILUETTELO.....	2
1. JOHDANTO	3
1.1 TAUSTA.....	3
1.2 TAVOITTEET JA RAJAUKSET	3
1.3 TYÖN RAKENNE	4
2. SKAALATTAVA VEKTORIGRAFIikka.....	5
2.1 SVG:N RAKENNE	5
2.2 GRAFIIKAN LISÄÄMINEN	5
3. KOORDINAATISTOT.....	7
3.1 KARTTAKOORDINAATTIJÄRJESTELMÄ.....	7
3.2 KKJ:N MUODOSTAMINEN	8
4. PAIKKATIETO JA PAIKKATIEtOTIEDOSTOT	10
4.1 PAIKKATIETO	10
4.2 ESRI-MUOTOTIEDOSTO.....	10
4.2.1 <i>Indeksitiedosto</i>	11
4.2.2 <i>Päätiedosto</i>	11
4.2.3 <i>Tietokantatiedosto</i>	14
4.3 MAPINFO-TIEDONSIIRTOFORMAATTI.....	15
5. PAIKKATIEtOKANNAT.....	17
5.1 PAIKKATIEDON KÄSITTELY TIEtOKANNASSA	17
5.2 GEOMETRIAN KUVAAMINEN TEKSTINÄ JA BINÄÄRIMUODOSSA	19
5.5 MYSQL-PAIKKATIEtOLAAJENNUS.....	20
5.5.1 <i>R-puuindeksointi</i>	21
6. TOTEUTUS.....	23
6.1 JÄRJESTELMÄN RAKENNE.....	23
6.2 PIIRTOSOVELMA.....	23
6.1.1 <i>Pisteiden piirtäminen</i>	25
6.1.2 <i>Viivojen piirtäminen</i>	27
6.1.3 <i>Alueiden piirtäminen</i>	28
6.1.4 <i>Tietojen tallentaminen ja lataaminen</i>	28
6.1.5 <i>Tietojen tuominen tiedostosta</i>	29
6.2 PALVELINSOVELMA	30
7. YHTEENVETO	32
LÄHTEET	33
LIITE 1. ESIMERKKI OHJELMAN TUOTTAMASTA SVG-KOODISTA	36

LYHENNE- JA SYMBOLILUETTELO

ESRI	Environmental Systems Research Institute, Inc.
KKJ	Kartastokoordinaattijärjestelmä
MIF	MapInfo data interchange format
OGC	Open GIS Consortium, Inc.
SVG	Scalable Vector Graphics
WKB	Well-Known Binary
WKT	Well-Known Text
XML	Extensible Markup Language
YKJ	Yhtenäiskoordinaattijärjestelmä

1. Johdanto

1.1 Tausta

Tietotekniikan ja Internetin kehittyminen on viime vuosina avannut uusia mahdollisuuksia karttatuotantoalalle. Karttaikkuna Oy avasi vuonna 2003 verkkokaupan, jonka kautta jokainen pystyi tilaamaan kartan haluamastaan paikasta haluamassaan koossa. Aikaisemmin karttoja sai lähinnä valmiiksi painettuina karttalehtinä. Vaikka kartan saikin haluamastaan paikasta, ei sen sisältöön voinut vaikuttaa kuin tilaamalla erikseen käsin muokatun kartan. Käsityönä tehdyn kartan hinta taas saattoi aiheuttaa ongelmia, varsinkin pienempien tilausten yhteydessä. Työn aiheena olikin kehittää järjestelmä, jonka avulla asiakas voisi halutessaan itse lisätä omia paikkatietokohteita kartalle.

Verkkokaupan toiminnan alusta alkaen ideana on ollut mahdollisimman vähäinen käsityö. Asiakkaan tilaaman kartan tuotanto on siis pyritty automatisoimaan mahdollisimman pitkälle. Toiminnan sydämenä on Anssi Pulkkinen kehittämä automaattinen karttatuotantojärjestelmä nimeltä PDFMapper. PDFMapper on käytännössä Java-palvelinsovelma, joka luo tulostettavan kartan syötettyjen tietojen pohjalta. Jo alkuperäisessä järjestelmässä oli mahdollisuus lisätä tietoja kartan päälle, mutta kuitenkin hyvin rajallisesti. Karttaan pystyi lähinnä lisäämään viivoja ja tekstejä. Tämän puutteen korjaamiseksi uusimpaan kehitysversioon lisättiinkin mahdollisuus SVG-tiedostojen (Scalable Vector Graphics) [1] liittämiseen. SVG-tiedostot taas mahdollistavat hyvin monipuolisen grafiikan lisäämisen karttoihin. Monipuolisuutensa lisäksi SVG valittiin, koska sen käsittelyyn löytyy avoimeen lähdekoodiin perustuva ilmainen Java-kirjasto Batik [2]. Sitä käyttäen ei varsinaiseen grafiikan muodostamisesta tarvinnut huolehtia.

1.2 Tavoitteet ja rajaukset

Tavoitteena oli siis kehittää järjestelmä, jonka avulla asiakas voi verkkokaupassa lisätä tietoa tilaamalleen kartalle. Työssä tehty järjestelmä koostuu Java-sovelmasta, jolla tiedot syötetään ja Java-palvelinsovelmasta, joka hoitaa tiedon tallentamisen

tietokantaan sekä PDFMapperille menevän SVG-tiedoston tallentamisen palvelimelle. Tietojen syöttämistä varten päätettiin ohjelmaan luoda yksinkertaiset piirtotyökalut sekä mahdollisuus tuoda tietoja kahdesta paljon käytetystä paikkatietoformaattista.

1.3 Työn rakenne

Työn toisessa luvussa kerrotaan SVG-tiedostomuodosta. Kolmannessa luvussa tutustutaan kehitetyssä järjestelmässä ja kartoissa käytettävään koordinaattijärjestelmään. Neljännessä luvussa käsitellään yleisesti paikkatietoa sekä esitellään työssä käytetyt paikkatietotiedostot. Viidennessä luvussa käsitellään paikkatietokantoja. Kuudennessa luvussa esitellään valmis järjestelmä. Yhteenveto esitetään seitsemännessä luvussa.

2. Skaalattava vektorigrafiikka

2.1 SVG:n rakenne

SVG on kaksiulotteisen grafiikan esittämiseen suunniteltu kieli. Se on XML:n pohjautuva ja sen luominen ja editointi onnistuu tekstieditorilla. SVG:n avulla voidaan kuvata

- vektorigrafiikkaa,
- rasterigrafiikkaa,
- tekstejä.

Kaikki graafiset komponentit tulevat `svg`-elementin sisään. Elementille annetaan parametrina alue, minkä kyseinen ryhmä kattaa. Alue voidaan antaa pelkästään leveytenä ja pituutena mutta sille voi antaa myös vasemman yläkulman koordinaatit. Elementin attribuutteihin lisätään myös yleensä `svg`-nimiavaruus. `Svg`-elementti voi olla esimerkiksi seuraavanlainen: `<svg width="5cm" height="4cm" version="1.1" xmlns=http://www.w3.org/2000/svg>`. SVG-dokumentin koordinaatiston origo on vasemmassa yläkulmassa, eli `x`-koordinaatti kasvaa oikealle ja `y`-koordinaatti alaspäin mentäessä [1].

2.2 Grafiikan lisääminen

Graafisen kohteiden lisäämistä varten kielessä määritellään joukko valmiita perusmuotoja:

- `rect`.
- `circle`.
- `ellipse`.
- `line`.
- `polyline`.
- `polygon`.

Valmiiden muotojen lisäksi kohteita voi muodostaa poluista. Polkua muodostettaessa toimitaan, kuin kohdetta piirrettäisiin kynällä. Ensin siirrytään haluttuun aloituspaikkaan `moveTo` (M) komenolla, jonka jälkeen voi piirtää esimerkiksi suoran viivan komennolla `lineTo` (L) tai jonkun mahdollisista kaarista. Viivan piirto lopetetaan

kirjaimella z. Esimerkiksi koodi `<path d="M 100 100 L 300 100 L 200 300 z" fill="red" stroke="blue" stroke-width="3" />` piirtää kolmion [1].

Graafisille elementeille voi antaa attribuutteina erilaisia muotoiluja, kuten piirtotyyli, piirtoväri ja täyttöväri. Muotoilut voidaan antaa omina attribuutteinaan tai ne voidaan kerätä yhden tyyliattribuutin sisään. Tyylit voidaan myös kuvailla omassa tiedostossaan tai erillään dokumentin alussa. Esimerkiksi koodit `<rect x="200" y="100" width="600" height="300" style="fill:red; stroke:blue; stroke-width:3"/>` ja `<rect x="200" y="100" width="600" height="300" fill="red" stroke="blue" stroke-width="3"/>` tuottavat samanlaisen lopputuloksen [1].

Tekstin lisäämistä varten on oma elementtinsä nimeltä text. Tekstielementille voi attribuutteina antaa mm. tekstin paikan, fontin, koon sekä tekstin tasauksen. Tekstielementti voi olla esimerkiksi seuraavanlainen [1]:

```
<text x="250" y="150" font-family="Verdana" font-size="55" fill="blue" >  
    Hello, out there  
</text>
```

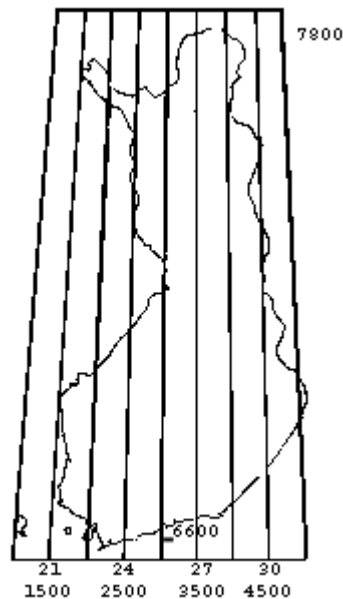
SVG-dokumenttiin voi myös tarvittaessa lisätä rasterikuvia. Tätä varten kielestä löytyy elementti image, jonka attribuuttina annetaan linkki kuvatiedostoon: `<image x="200" y="200" width="100px" height="100px" xlink:href="myimage.png">`[1].

SVG:ssä kohteille voi myös tehdä erilaisia muunnoksia. Kohteita voi siirtää, käänellä ja skaalata käyttämällä transform-attribuuttia, jonka arvossa kerrotaan mitä muunnoksia halutaan tehdä (translate, rotata, scale) ja annetaan niille arvot. Esimerkiksi koodi `<g transform="translate(50,30)">` siirtää ryhmäelementin sisässä olevia kohteita 50 yksikköä vaakasuunnassa ja 30 yksikköä pystysuunnassa [1].

3. Koordinaatit

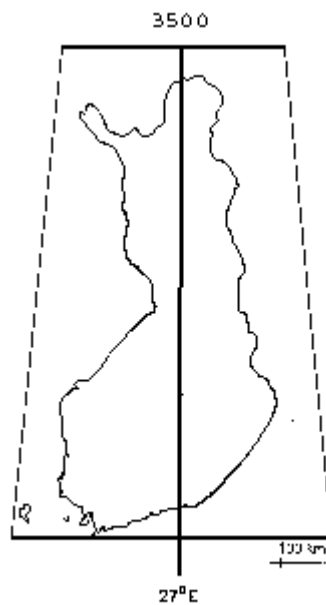
3.1 Karttakoordinaattijärjestelmä

Suomessa peruskarttojen koordinaatistojärjestelmänä on pitkään käytetty kartastokoordinaattijärjestelmää (KKJ). KKJ:ssa Suomi on jaettu kuuteen kaistaan, joista jokaisella on oma koordinaatisto (kuva 1). Jokaisen kaistan leveys on kolme astetta. Koordinaatistojen origot ovat päiväntasaajan ja kyseisen kaistan keskimeridiaanin leikkauskohdassa. Keskimeridiaanilla tarkoitetaan kaistan keskimmäistä pituuspiiriä. KKJ:ssa koordinaatit kerrotaan metreinä etäisyytenä päiväntasaajasta ja kaistan keskimeridiaanista. Jotta negatiivisilta arvoilta vältyttäisiin, keskimeridiaanille on annettu arvoksi 500 000 m. Eri kaistoihin kuuluvien koordinaattiin tunnistamiseksi itäkoordinaatin eteen lisätään kaistan numero [3].



Kuva 1. KKJ-kaistajako ja kaistojen keskimeridiaanit [4].

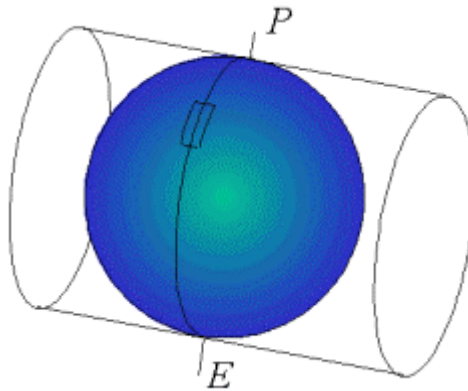
Kaistajako aiheuttaa ongelmia, kun kartta pitäisi saada kahden viereisen kaistan alueelta. Tämän vuoksi käytössä on myös yhtenäiskoordinaatistojärjestelmä (YKJ). Siinä KKJ:n kolmas kaista on levitetty kattamaan koko Suomi ja keskimeridiaanin arvoksi annettu 3 500 000 m (kuva 2). Kaistan levittäminen lisää maapallon muodosta johtuvia projektiovirheitä sen laidoilla, mutta tarkkuus pysyy kuitenkin riittävänä peruskäyttöön [3]. Karttaikkunan järjestelmät käyttävät YKJ-koordinaatistoa.



Kuva 2. YKJ kaista ja sen keskimeridiaani [4].

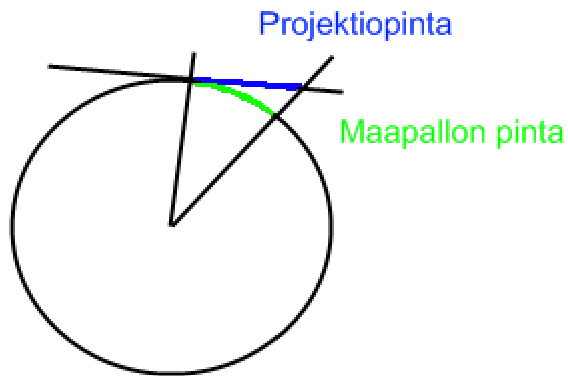
3.2 KKJ:n muodostaminen

Maapallo on kolmiulotteinen kappale, jonka vuoksi pisteen saamiseksi tasolle on käytettävä jotain projisointimenetelmää. Projisointi voidaan tehdä suoraan tasolle, mutta tarkemman tuloksen yleensä saa käyttämällä lieriötä tai kartiota, joka sitten levitetään tasoksi. Projektio-laskuja varten myös maapallon pinnasta on saatava maatemattinen kuvaus. Tätä kuvausta kutsutaan vertausellipsoidiksi [5]. KKJ on suorakulmainen koordinaatisto, jossa projektiona käytetään Gauss-Krüger-karttaprojektiota ja vertausellipsoidina on Hayfordin ellipsoidi. Gauss-Krüger-projektiossa vertausellipsoidin ympärille pyöräytetään lieriö, joka sivuaa jotain pituuspiiriä (kuva 3)[6].



Kuva 3. Gauss-Krüger-projektio [6].

Kohteet projisoidaan lieriön pintaan vetämällä suora viiva ellipsoidin keskipisteestä halutun pisteen kautta lieriölle [6]. Muodostamistapansa vuoksi projektio aiheuttaa projektiovirhettä itäsuunnassa heti, kun poistutaan projektiokaistan keskimeridiaanilta. Tämä virhe on kuitenkin normaali käytössä häviävän pieni. Käytännössä projektiovirhe aiheuttaa sen, että etäisyydet kartalla mitattuna poikkeavat luonnossa mitatuista (kuva 4).



Kuva 4. Projektiovirhe.

4. Paikkatieto ja paikkatietotiedostot

4.1 Paikkatieto

Paikkatieto on johonkin sijaintiin sidottua tietoa. Paikkatietoon läheisesti liittyviä termejä on karttakohde, sijainti- ja ominaisuustieto. Karttakohdeella kertoo, mitä reaali maailman asiaa kyseisellä kohteella kuvataan. Kohde voi olla esimerkiksi kivi, puu, pelto tai tie. Kohteet voivat olla pisteitä, viivoja tai alueita. Sijainnilla tarkoitetaan yleensä kohteen koordinaatteja, mutta se voi olla myös esimerkiksi osoite tai muu vastaava paikan yksiselitteisesti kertova tieto. Jos kyseessä on viiva tai alumuotoinen kohde, paikkatieto koostuu useammasta pisteestä. Jokaiselle kohteelle voi olla olemassa useita kohteeseen liittyviä ominaisuustietoja. Ominaisuus tietona voi olla esimerkiksi alueen pinta-ala tai pituus. Paikkatietoon liittyy myös käsite topologia, joka kuvaa kohteiden välisiä suhteita [5].

4.2 ESRI-muototiedosto

ESRI-muototiedosto (Environmental Systems Research Institute, Inc.) on 1990-luvun alussa kehitetty vektoritiedostomuoto paikkatiedon siirtämiseen ja varastointiin. Yksi muototiedosto koostuu itse asiassa useammasta tiedostosta. Pakolliset tiedostot ovat ns. indeksi-, pää- ja tietokantatiedostot, mutta pakettiin voi kuulua myös muita tiedostoja. Samaan pakettiin kuuluvilla tiedostoilla on oltava päätettä lukuun ottamatta yhtenevä nimi. Muototiedoston tukemat kolme paikkatiedon perustyyppiä ovat piste, viiva ja alue. Näistä perustyypeistä on vielä olemassa myös korkeusarvolla ja mitta-arvolla (pisteZ, pisteM) varustetut versiot. Kun mukaan vielä lasketaan tyhjä-tyyppi, jonka voi tallentaa mihin tahansa tiedostoon, sekä pintapala ja useista pistekohteista muodostuva monipiste, saadaan erilaisten tyyppien yhteismääräksi 14 (taulukko 1) [7].

Kaikki muototiedoston sisältämät kohteet on oltava saman tyyppisiä, eli esimerkiksi pisteille ja viivoille on oltava omat tiedostonsa. Muototiedosto on binääritiedosto, joten sen lukemiseen ja kirjoittamiseen tarvitaan siihen tarkoitukseen tehty ohjelma [7].

Taulukko 1. Muototiedoston kohdetyypit sekä niitä vastaavat koodit [7].

arvo	tyyppi
0	Tyhjä
1	Piste
3	Viiva
5	Alue
8	Monipiste
11	PisteZ
13	ViivaZ
15	AlueZ
18	MonipisteZ
21	PisteM
23	ViivaM
25	AlueM
28	MonipisteM
31	Pintapala

4.2.1 Indeksitiedosto

Indeksitiedosto sisältää eräänlaisen sisällysluettelon. Tiedoston alussa 100 tavua pitkä tiedosto-otsikko, jonka rakenne on kuvattu taulukossa 2. Otsikon jälkeen tiedostossa on jokaista pakettiin sisältyvää kohdetta kohden kahdeksantavuinen tietue, jossa kerrotaan mistä kohtaa kyseisen kohteen tiedot löytyvät päätiedostosta ja kuinka paljon tilaa se käyttää [7].

4.2.2 Päätiedosto

Varsinainen koordinaattitieto on tallennettu päätiedostoon, jonka tunnistaa päätteestä ”shp”. Tiedoston ensimmäiset 100 tavua on varattu tiedosto-otsikolle. Otsikko sisältää tiedostotunnisteen, tiedoston koon, kohteiden tyyppin sekä alueen, missä tiedoston sisältämät kohteet sijaitsevat. Otsikon rakenne on kuvattu taulukossa 2 [7].

Taulukko 2. Muototiedoston otsikkotiedot [7].

Sijainti	Kenttä	Arvo	Tyyppi
Tavu 0	Tiedostotunniste	9994	Kokonaisluku
Tavu 4	Käyttämätön	0	Kokonaisluku
Tavu 8	Käyttämätön	0	Kokonaisluku
Tavu 12	Käyttämätön	0	Kokonaisluku
Tavu 16	Käyttämätön	0	Kokonaisluku
Tavu 20	Käyttämätön	0	Kokonaisluku
Tavu 24	Tiedoston pituus	Tiedoston pituus	Kokonaisluku
Tavu 28	Versio	1000	Kokonaisluku
Tavu 32	Kohteen tyyppi	Kohteen tyyppi	Kokonaisluku
Tavu 36	Alue	Xmin	Liukuluku
Tavu 44	Alue	Ymin	Liukuluku
Tavu 52	Alue	Xmax	Liukuluku
Tavu 60	Alue	Ymax	Liukuluku
Tavu 68	Alue	Zmin	Liukuluku
Tavu 76	Alue	Zmax	Liukuluku
Tavu 84	Alue	Mmin	Liukuluku
Tavu 92	Alue	Mmax	Liukuluku

Varsinainen koordinaattitieto on talletettu vaihtuvanmittaisiin tietueisiin otsikkotiedon jälkeen. Tietueen sisältö ja rakenne riippuu täysin talletettavana olevan kohteen tyypistä. Myös tietuilla on oma otsikkotietonsa, jolle on varattu ensimmäiset 8-tavua tietueesta. Se koostuu tietueen numeroista sekä tietueen pituudesta (taulukko 3) [7].

Taulukko 3. Tietueen otsikkotieto [7].

Sijainti	Kenttä	Arvo	Tyyppi
Tavu 0	Tietueen numero	Tietueen numero	Kokonaisluku
Tavu 4	Tietueen pituus	Tietueen pituus	Kokonaisluku

Yksinkertaisin kohteista on piste, jonka tietueen tieto-osa sisältää vain kohteen tyyppitiedon sekä pisteen x- ja y-koordinaatit (taulukko 4) [7].

Taulukko 4. Pistekohteen tietueen rakenne [7].

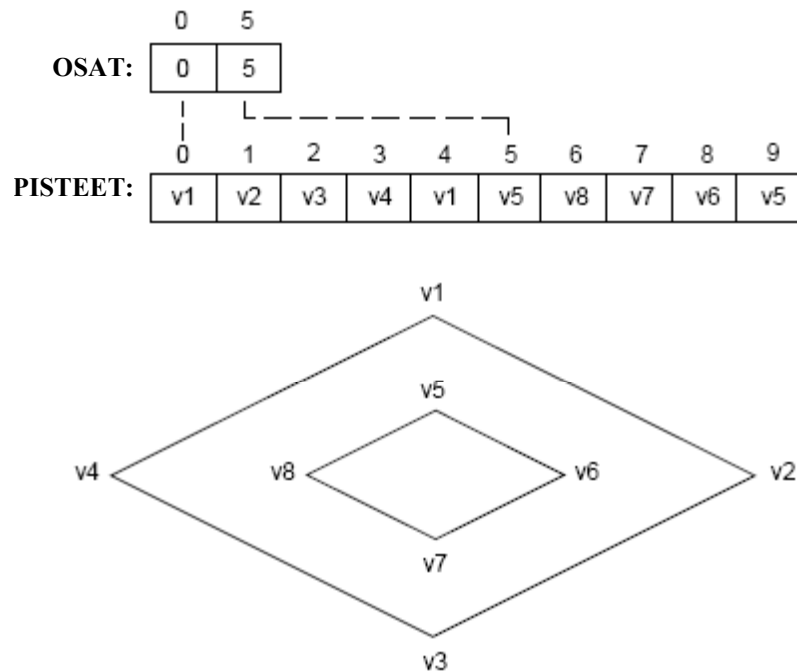
Sijainti	Kenttä	Arvo	Tyyppi
Tavu 0	Kohteen tyyppi	1	Kokonaisluku
Tavu 4	X	X-koordinaatti	Liukuluku
Tavu 12	Y	Y-koordinaatti	Liukuluku

Viiva-kohteen tietue voi sisältää useampia viivoja. Tietueen ensimmäiset neljä tavua on varattu kohteen tyyppi, joka tässä tapauksessa on siis viivalle varattu kolmonen. Kohteen tyyppin jälkeiset 32 tavua on varattu rajaussuorakulmioille, joka kertoo, minkä alueen kohde kattaa. Koska viiva-tietue voi sisältää useita viivoja seuraavaksi kerrotaan viivojen määrä, jonka jälkeen kerrotaan koordinaattipisteiden kokonaismäärä. Tämän jälkeen tietueessa on osien määrän verran kokonaislukuja, jotka kertovat, mistä kohdin seuraavana olevia koordinaattipisteitä tietyn osan koordinaatit alkavat. Lopussa on siis varsinainen koordinaattitieto liukuluku pareina (taulukko 5) [7].

Taulukko 5. Viiva-tyypin tietueen rakenne, jossa $K = 44 + 4 * \text{Osien määrä}$ [7].

Sijainti	Kenttä	Arvo	Tyyppi
Tavu 0	Kohteen tyyppi	3	Kokonaisluku
Tavu 4	Alue	Alue (Xmin, Ymin, Xmax, Ymax)	Liukuluku
Tavu 36	Osien määrä	Osien määrä	Kokonaisluku
Tavu 40	Pisteiden määrä	Pisteiden määrä	Kokonaisluku
Tavu 44	Osat	Osien alkujen sijainnit	Kokonaisluku
Tavu K	Pisteet	Pisteet	Piste

Alue-tyyppisen kohteen tietueen rakenne on, kohteen tyyppiä lukuun ottamatta, yhtenevä viiva-tietueen kanssa (taulukko 6). Sisältöä on vain tulkittava eri tavoin. Alueet koostuvat renkaista, eli viivoista joiden alku- ja loppupiste on yhdistetty. Viivan ensimmäisen ja viimeisen koordinaattiparin on siis oltava samat. Tämän lisäksi alueet voivat sisältää reikiä. Muototiedostossa alueiden ulkoreunat ja reiät erotetaan toisistaan koordinaattipisteiden järjestyksellä. Tietueessa myötäpäivään kulkeva rengas tarkoittaa alueen ulkorajaa ja vastapäivään kulkeva rengas aukkoa alueen keskellä. Kuvassa 5 on esimerkki kuinka alueen ulkoraja ja aukko on talletettu tietueeseen.



Kuva 5: Alueen ja siinä olevan aukon tallennus tietueeseen [7].

Taulukko 6. Alue-tietueen rakenne, jossa $K = 44 + 4 * \text{Osien määrä}$ [7].

Sijainti	Kenttä	Arvo	Tyyppi
Tavu 0	Kohteen tyyppi	5	Kokonaisluku
Tavu 4	Alue	Alue (Xmin, Ymin, Xmax, Ymax)	Liukuluku
Tavu 36	Osien määrä	Osien määrä	Kokonaisluku
Tavu 40	Pisteiden määrä	Pisteiden määrä	Kokonaisluku
Tavu 44	Osat	Osien alkujen sijainnit	Kokonaisluku
Tavu K	Pisteet	Pisteet	Piste

4.2.3 Tietokantatiedosto

Paikkatietoon liittyvät ominaisuustiedot on talletettu dBase-tietokantatauluun [8], niin että jokaiselle kohteelle on oma tietueensa taulussa. Tietueiden on oltava samassa järjestyksessä kuin kohteet päätiedostossa [7].

4.3 MapInfo-tiedonsiirtoformaatti

Toinen työhön mukaan otettu paikkatietomuoto on paikkatietosovelluksia kehittävän, MapInfo:n omaan käyttöön suunnittelema tiedonsiirtoformaatti MapInfo Data Interchange Format eli MIF. MIF-tiedostot ovat tekstitiedostoja, joten niiden lukeminen ja kirjoittaminen on mahdollista tekstieditorilla. MIF koostuu yleensä kahdesta tiedostosta; mif-päätteisestä päätiedostosta sekä mid-päätteisestä ominaisuustiedostosta. MIF-tiedostomuodon suunnittelussa on otettu huomioon tiedon graafinen kuvaaminen, joten siihen talletettavat kohteet ovat useista piirtotyökaluistakin tuttuja graafisia peruselementtejä:

- Piste.
- Viiva.
- Moniosainen viiva.
- Alue.
- Kaari.
- Teksti.
- Suorakulmio.
- Pyöreäkulmainen suorakulmio.
- Ellipsi.

Edellisten lisäksi kohde voi muodostua useammasta pisteestä tai olla kokoelma joistain edellä mainituista. Jokaiselle kohteelle voi myös antaa piirtotyylin, eli millä symbolilla kyseistä kohdetta kuvataan [9].

Päätiedosto koostuu kahdesta osasta, jotka ovat otsikko sekä varsinainen tieto. Otsikko alkaa tiedoston versiolla. Koska tiedostomuoto on muuttunut aikojen saatossa on lukijoille kerrottava, minkä version mukainen kyseinen tiedosto on. Version jälkeen tiedostossa voidaan kertoa käytetty merkistökoodaus, ominaisuustiedostossa käytetty välimerkki ja muita vastaavia tiedoston tulkitsemiseen tarvittavia tietoja. Otsikko-osassa voidaan myös kertoa paikkatiedon paikan esittämiseen käytetty koordinaatistojärjestelmä. Jos koordinaatistoa ei ole määritelty, koordinaattien oletetaan olevan pituus- ja leveysasteina. Otsikko-osiossa on myös lueteltu ominaisuustiedostossa olevien sarakkeiden määrä sekä niiden otsikot ja tyypit [9].

Tieto-osa koostuu varsinaisesta koordinaattitiedosta. Osan alkamisen tunnistaa sanasta DATA. Tämän jälkeen on lueteltu kohteiden tiedot. Kohdetieto alkaa aina kohteen tyyppiä kuvaavalla sanalla, joita on NONE, POINT, LINE, PLINE, REGION, ARC, TEXT, RECT, ROUNDRECT, ELLIPSE, MULTIPOINT sekä COLLECTION. Tyyppiä NONE tarvitaan, koska jokaista ominaisuustiedostossa olevaa riviä on vastattava kohde päätiedostossa ja ominaisuustietoja voi olla myös ilman varsinaista graafista kohdetta. Koordinaattitiedon esittäminen riippuu kohteen tyypistä. Pisteiden koordinaatit ovat suoraan POINT-avainsanan jälkeen, välilyönnillä eroteltuina. Myös viivasymbolin, kaaren, suorakulmion, pyöristetyn suorakulmion ja ellipsin koordinaatit ovat suoraan avainsanan jälkeen. Monimutkaisemmissa kohteissa, kuten moniosainen viiva, alue sekä kokoelma, avainsanan jälkeen kerrotaan kuinka monesta osasta kohde koostuu sekä joissain tapauksissa pisteiden määrä. Tämän jälkeen osat ja pisteet luetellaan rivinvaihdolla eroteltuina. Jokaiseen kohteeseen voi myös liittyä tieto, millä symbolilla tai millaisella viivalla kyseistä kohdetta pitäisi kuvata kartalla [9].

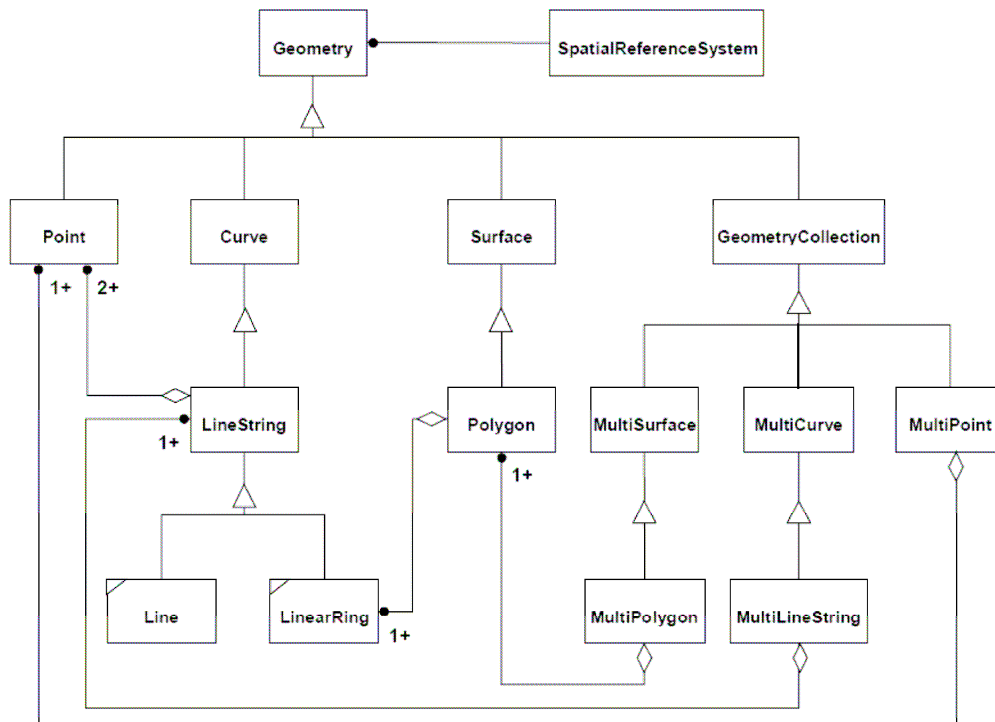
Ominaisuustiedostossa on lueteltu kohteisiin liittyvät ominaisuudet, päätiedostossa määritellyllä erottimella erotettuna. Jokaista päätiedostossa olevaa kohdetta vastaa yksi rivi ominaisuustiedostossa. Ominaisuustiedot on oltava oikeassa järjestyksessä verrattuna kohdetietoon, koska tiettyyn kohteeseen liittyvä ominaisuustieto tunnistetaan ainoastaan järjestyksen mukaan [9].

5. Paikkatietokannat

5.1 Paikkatiedon käsittely tietokannassa

Paikkatietokanta on tietokanta, johon on lisätty moniulotteisen tiedon käsittelyä tehostavia ja helpottavia toimintoja [10]. Paikkatietoa käsiteltäessä tietomäärät voivat kasvaa hyvinkin suuriksi. Luonnollinen valinta suurten tietomäärien tallentamiseen on tietokanta. Perinteiset tietokannat eivät kuitenkaan toimi kovin tehokkaasti paikkatietoa tai muuta useampiulotteista tietoa käsiteltäessä. Paikkatietoa varten joihinkin tietokantoihin onkin kehitetty paikkatietolaajennukset, jotka mahdollistavat paikkatiedon tehokkaan käsittelyn.

OGC (Open GIS Consortium, Inc.) on kansainvälinen satojen yritysten muodostama yhteenliittymä, jonka tarkoituksena on kehittää yhteisiä avoimia standardeja paikkatiedolle. OpenGIS Simple Features Specification for SQL on OGC:n määrittelemä standardi paikkatiedon käsittelyyn tietokannoissa. Paikkatietoon liittyvän koordinaattitiedon käsittelyä varten standardissa määritellään kohteille oliomalli. Perusluokkana mallissa on geometria, josta kaikki kohteet periytetään (kuva 6) [11].



Kuva 6 Geometria-luokan oliomalli [11].

Geometria-luokassa määritellään myös joukko kaikille kohteille ja kohdetyypeille yhteisiä funktioita [11]:

- Yleiset kohteisiin liittyvät funktiot:
 - Dimension.
 - GeometryType.
 - SRID.
 - Envelope.
 - AsText.
 - AsBinary.
 - IsEmpty.
 - IsSimple.
 - Boundary.
- Kohteiden välisiin suhteisiin liittyvät funktiot:
 - Equals.
 - Disjoint.
 - Intersects.
 - Touches.
 - Crosses.
 - Within.
 - Contains.
 - Overlaps.
 - Relate.
- Paikkatiedon analysointia helpottavat funktiot:
 - Distance.
 - Buffer.
 - ConvexHull.
 - Union.
 - Difference.
 - SymDifference.

Standardissa määritellään rakennekuvaus sekä perustietokannoille että tietokannoille, jotka tukevat geometrisia tietotyypppejä. Jos tietokanta ei tue geometrisia tietotyypppejä, koordinaattitieto talletetaan omaan tauluunsa käyttäen perinteisiä numeerisia tietotyypppejä tai WKB-muotoa (Well-Known Binary). Geometrisia tietotyypppejä tukevissa kannoissa niille on varattu oma GEOMETRY-tietotyyppi [11].

5.2 Geometrian kuvaaminen tekstinä ja binäärimuodossa

WKT (Well-Known Text) ja WKB ovat standardisoituja tapoja esittää paikkatietokohteeseen liittyvä geometria. Standardissa määritellään miten eri tyyppiset kohteet tulee kuvata. Taulukossa 7 on esimerkit pisteen viivan ja alueen WKT-esityksistä [12].

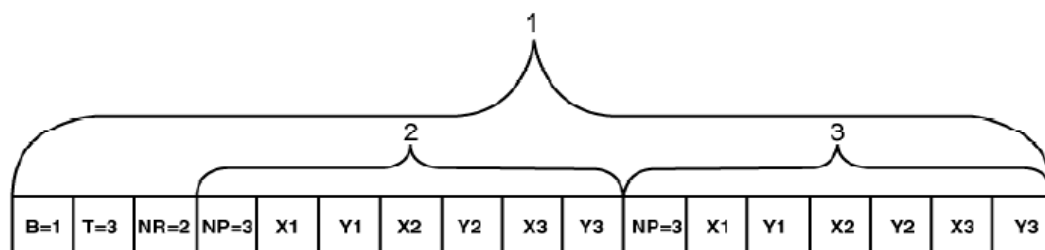
Taulukko 7. Pisteen, viivan ja alueen WKT esitykset.

Piste	POINT(3572000 6854000)
Viiva	LINestring(3572000 6854000,3572500 68545000)
Alue	POLYGON((3572000 6854000,3572500 6854500,3572000 6854500,3572000 6854000))

WKT esityksessä pisteen x- ja y-koordinaatit on erotettu toisistaan välilyönnillä ja eri pisteet taas on erotettu pilkulla. Aluetta kuvatessa ensimmäisen ja viimeisen koordinaattiparin on oltava samat. Jos alueen sisällä on reikiä, niiden koordinaatit lisätään pilkulla erotettuna sulkeissa aina ensimmäisenä olevan ulkorajan perään [12].

WKB mahdollistaa geometrioiden siirtämisen binäärimuodossa. Se on myös yleinen tapa tallentaa geometria paikkatietokantaan. WKB:ssa ensimmäiset neljä tavua sisältävät kokonaisluvun joka kertoo tavujärjestyksen. Seuraavat neljä tavua on varattu kohteen tyyppin kertovalle kokonaisluvulle. Seuraavien tavujen tarkoitus vaihtelee kohteen tyyppin mukaan. Jos kyseessä on piste, on seuraavana kaksi liukulukua, jotka kertovat pisteen koordinaatit. Viivakohteilla seuraavana on kokonaisluku, josta selviää viivaan kuuluvien pisteiden määrä. Alueilla kyseinen kohta on varattu renkaiden määrälle. Tämän jälkeen viivassa on pisteiden määrän verran koordinaattipareja

liukulukuina. Aluekohteessa taas seuraavana on ensimmäisen renkaan (ulkoreuna) viivan sisältämien pisteiden määrä kokonaislukuna, jonka jälkeen seuraa pisteiden koordinaatit liukulukuina. Jos alueessa on reikiä, niitä kuvaavat viivat tulevat ulkoreunan jälkeen. Jokainen reikä alkaa aina kokonaisluvulla, joka kertoo viivaan kuuluvien pisteiden määrän [12]. Kuvassa 7 on esimerkki aluekohteen sisällöstä.



Kuva 7. Yhden reiän sisältämän alueen WKB:n sisältö. Kakkosella merkityt luvut kuuluvat alueen ulkoreunaan ja kolmosella merkityt alueessa olevaan reikään [12].

5.5 MySQL-paikkatietolaajennus

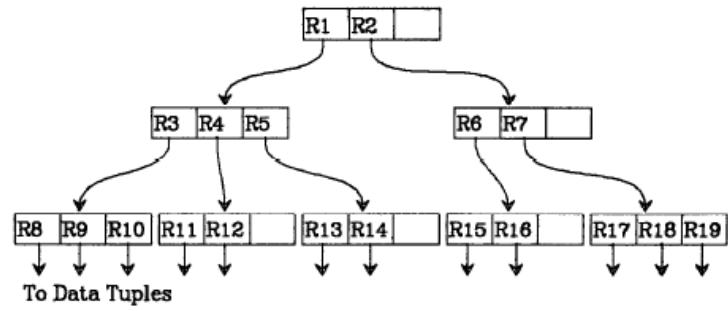
MySQL on ilmainen, avoimeen lähdekoodin perustuva, tietokantojen hallintajärjestelmä, joka tarjoaa myös laajennukset paikkatiedon hyödyntämiselle. MySQL:n paikkatietolaajennus on toteutettu käyttäen edellä esiteltyä OpenGIS Simple Features Specification for SQL -määritelmää, mutta toteutus on hieman vajaa. Kaikkia standardin määrittelemiä funktioita ei ole toteutettu ja osa funktioista ei toimi aivan standardin mukaisesti. MySQL tukee geometrisia tietotyyppejä, joten paikkatietokohteiden koordinaatit talletetaan GEOMETRY-tyyppiseen soluun. Kyseinen solu voi siis sisältää minkä tahansa paikkatietokohteen. Tiedon syöttämiseen voi käyttää WKT- ja WKB-muotoja ja niille tarkoitettuja funktioita `GeomFromText` ja `GeomFromWKB`. Esimerkiksi komento `INSERT INTO taulu VALUES GeomFromText('POINT(3572000 6854000)')` lisää tauluun pisteen [13].

5.5.1 R-puuindeksointi

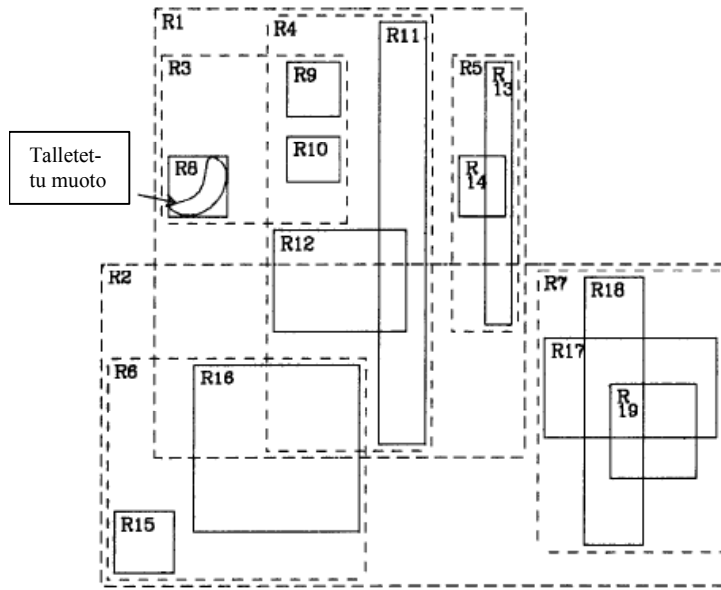
Kuten normaaleissa tietokannoissa paikkatietokannoissa erittäin tärkeää osaa näyttelee tiedon indeksointi. Indeksoinnin avulla tiedon käsittely ja hakemista voidaan nopeuttaa, kun koko kanta ei tarvitse käydä läpi hakujen aikana. Normaalitietokantojen indeksoinnit eivät kuitenkaan sellaisenaan sovellu hyvin moniulotteisen tiedon luettelointiin. Paikkatietoa varten onkin kehitetty useita indeksointimenetelmiä hakujen tehostamiseksi [10]. Kehitettyjä indeksointi menetelmiä ovat esimerkiksi nelipuu [14], kd-puu [15] ja r-puu [16]. MySQL käyttää paikkatiedon indeksointiin r-puuindeksiä [13].

R-puu on puurakenteinen indeksi, joka perustuu minimaalisten rajaussuorakulmioiden käyttämiseen. Minimaalinen rajaussuorakulmio kertoo pienimmän alueen, minkä sisään tietty kohde mahtuu. Puun lehdet sisältävät viittaukset tietokannassa oleviin kohteisiin ja kyseisen kohteen rajaussuorakulmion. Jokainen puun solmu, joka ei ole lehti, taas sisältää viittauksen puussa alempana oleviin solmuihin ja rajaussuorakulmion, minkä sisään kaikki alempana olevien solmujen rajaussuorakulmiot mahtuvat. Kuvassa 8 on esitelty puun rakenne ja sitä vastaavat alueet [16].

Tiedon hakeminen puusta tapahtuu vertaamalla solmujen rajaussuorakulmioita haettavan kohteet rajaussuorakulmioon. Jos rajaukset leikkaavat siirrytään taso alaspäin ja suoritetaan vertailu uudestaan. Tätä jatketaan kunnes puu on käyty läpi [16].



(a)



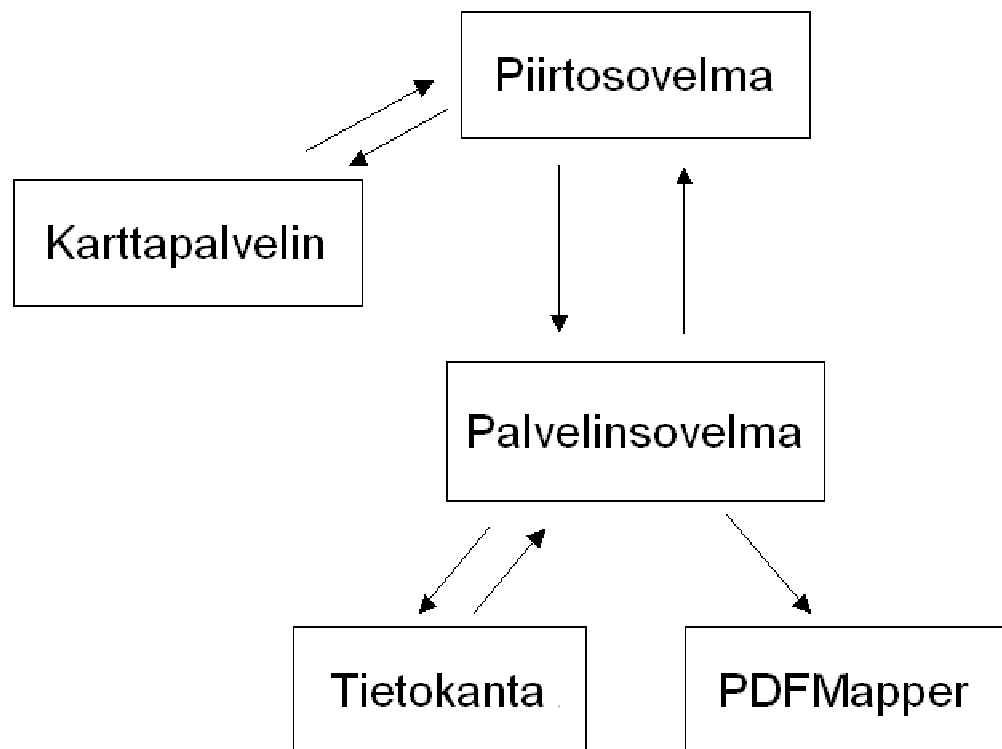
(b)

Kuva 8. R-puu: (a) rakenne; (b) rakennetta vastaavat kohteet rajaussuorakulmioineen [16].

6. Toteutus

6.1 Järjestelmän rakenne

Järjestelmän on toteuttanut tämän diplomityön tekijä omatoimisesti lukuun ottamatta PDFMapperia. Toteutettu järjestelmä koostuu asiakkaan käytössä olevasta piirto-sovelmasta, tietojen tallentamisen hoitavasta palvelinsovelmasta sekä tietokannasta, johon tiedot talletetaan (kuva 9). Järjestelmään kuuluu kiinteänä osana myös jo olemassa olevat, tulostettavat kartat tekevä PDFMapper sekä karttapalvelin, joka tuottaa sovelmassa näytettävät kartat.

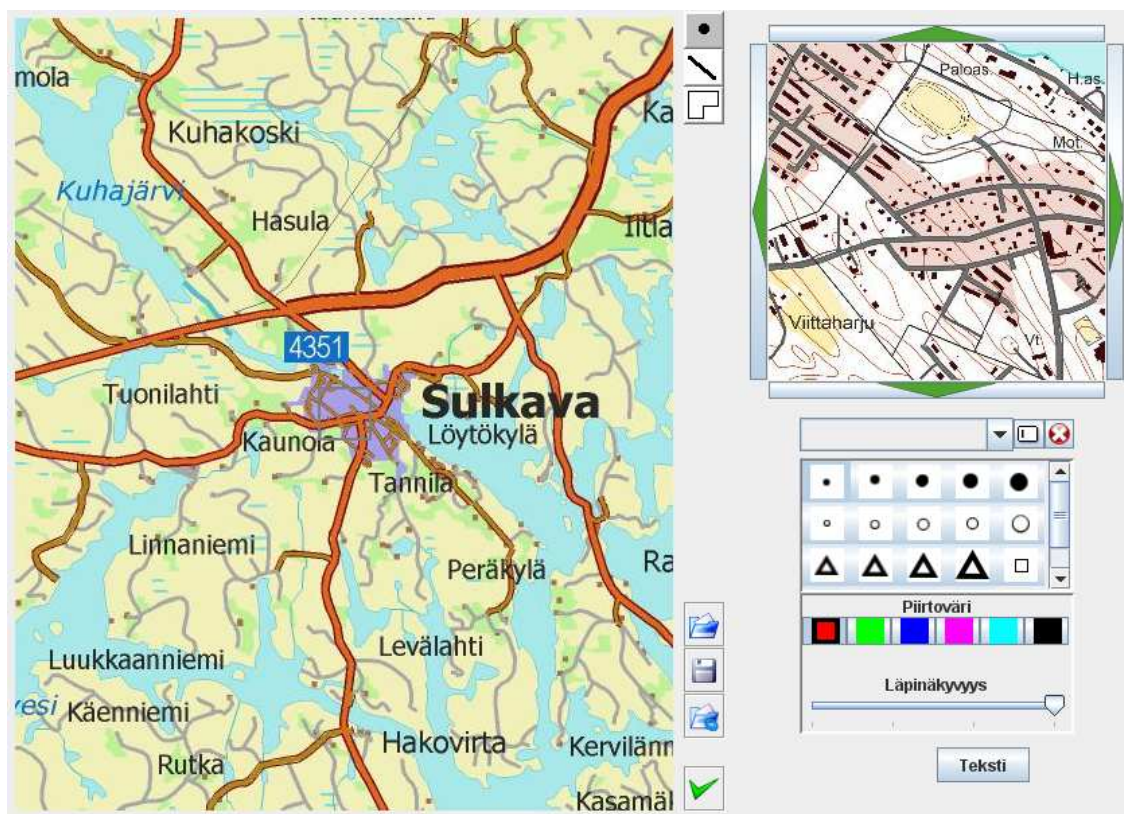


Kuva 9. Järjestelmän rakenne.

6.2 Piirto-sovelma

Paikkatietokohteiden, piste, viiva ja alue, lisäämiseksi kartalle kehitettiin Java-sovelma. Suunnittelun lähtökohtana oli pitää käyttöliittymä ja sovelman toiminta mahdollisimman helppona ja yksinkertaisena. Esimerkiksi eri tyyppisten kohteiden piirtäminen pyrittiin tekemään mahdollisimman samankaltaiseksi. Piirto-sovelma

koostuu vasemmalla olevasta suuresta karttaikkunasta, työkalupalkista sekä työkalupaneelista, jossa on pienempi kartta kohteen tarkempaa sijoittamista varten sekä piirtotyökaluihin liittyviä lisävalintoja (kuva 10).



Kuva 10. Kuva piirtosovelmasta.

Suuremmasta karttaikkunasta käyttäjä saa yleiskuvan käsiteltävästä alueesta ja pystyy siirtämään pienemmän kartan paikka hiiren vasemmalla painikkeella. Yleiskartan mittakaava on kuitenkin niin pieni, että sen päälle piirtäminen tarkasti on käytännössä mahdotonta. Karttakuvan yksi kuvapiste vastaa noin 20 metriä luonnossa. Tästä johtuen piirtäminen tapahtuukin pienemmällä kartalla, jossa käytetään maastokartta-aineistoa. Pienemmässä karttakuvassa yksi kuvapiste vastaa noin kolmea metriä luonnossa.

Työkalupalkissa ylimpänä olevista valintapainikkeista valitaan, minkä tyyppisiä kohteita halutaan piirtää. Ylimpänä on piste seuraavana viiva ja kolmantena alue. Palkin alareunasta löytyvät painikkeet piirrosten tallentamiseen, lataamiseen ja tiedoston

tuontiin sekä alimpana valmis-painike, josta ohjelma sulkeutuu ja lisätyistä kohteista luodaan SVG-tiedosto.

Oikeassa laidassa sijaitsevan työkalupaneelin sisältö riippuu osin valittuna olevasta piirtotyökalusta. Ylimpänä on kuitenkin aina karttaruutu, johon varsinainen kohteiden piirtäminen tehdään. Kartan laidoilla on myös nuolipainikkeet, joista kartta voidaan siirtää haluttuun suuntaan. Kaikille työkaluille yhteistä on myös seuraavana oleva yhdistelmäruutu, johon piirretyt kohteet lisätään. Yhdistelmäruudun sisältö muuttuu valitun piirtotyökalun mukaan. Sen oikealla puolella on painikkeet kohteen nimen muuttamista ja kohteen poistamista varten (kuva 11).



Kuva 11. Lisätyt kohteet sisältävä yhdistelmäruutu.

Piirtämiseen käytettävien symboleihin liittyen suunnittelussa päädyttiin käyttämään eräänlaista symbolikirjastoa. Ideana oli, että piirtosovelmassa olevia symboleita voidaan tarpeen mukaan tehdä helposti uusia ja symbolin ulkoasu olisi mahdollisimman vapaa. Näin voitaisiin tarvittaessa tehdä erilaisia symbolikirjastoja käyttötarkoituksen mukaan. Tästä johtuen kartalle piirrettävät pistekohteet piirretään polygoneina. Jokaiselle symbolille on myös annettu oma yksilöivä koodi. Tähän versioon ei kuitenkaan vielä toteutettu varsinaista ulkoista kirjastoa, mutta sen toteuttaminen tulevaisuudessa on tehty mahdollisimman helpoksi. Suunnitelmissa on myös tutkia mahdollisuutta asiakkaan omien symbolien lisäämiseen järjestelmään.

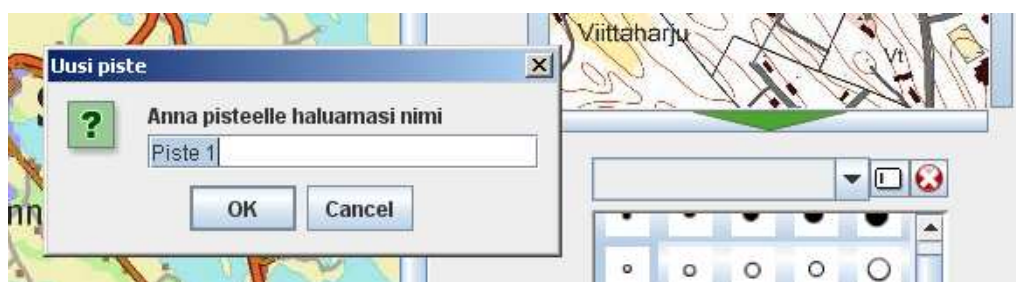
6.1.1 Pisteiden piirtäminen

Kun karttaan halutaan lisätä pistemäinen kohde, työkalupaneelissa (kuva 12) olevasta symbolilistasta valitaan ensin halutunlainen symboli ja piirtoväri.



Kuva 12. Pistekohteen työkalupaneeli.

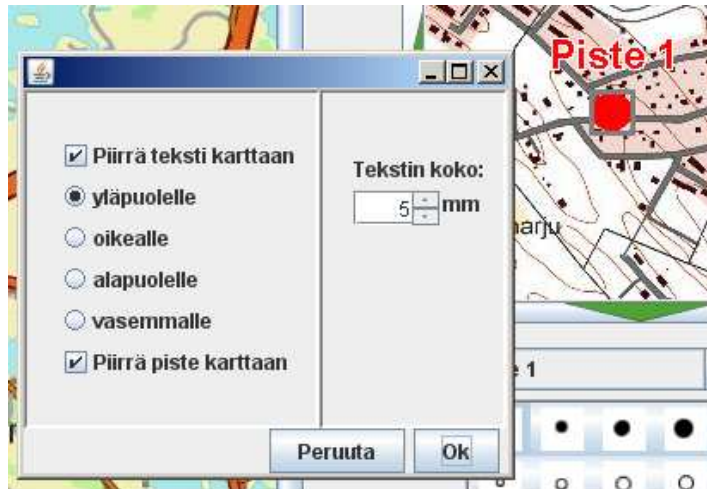
Kun halutut asetukset on valittu napsautetaan karttaa halutusta kohdasta hiiren vasemmalla painikkeella, jolloin aukeaa ikkuna (kuva 13), jossa voidaan pisteelle syöttää nimi. Kun auenneesta ikkunasta painetaan OK-painiketta, piste lisätään kartalle.



Kuva 13. Pisteen nimen syöttäminen.

Jo lisätyn pisteen symbolia ja väriä voi vielä halutessaan vaihtaa valitsemalla piste yhdistelmäruudusta ja haluamansa symbolin tai värin valittavissa olevista vaihtoehdoista. Valitun pisteen paikkaa voi myös muuttaa hiirellä vetämällä. Valittuna olevan pisteen ympärillä näkyy kartalla harmaa neliö.

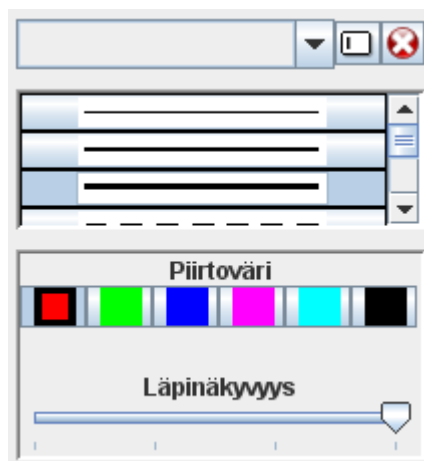
Pisteen yhteyteen voidaan myös haluttaessa liittää teksti. Kun napsautetaan tekstipainiketta aukeaa ikkuna (kuva 14), josta voidaan valita piirretäänkö pisteen nimi kartalle, mihin kohtaan teksti tulee, tekstin koko sekä piirretäänkö piste kartalle. Esimerkiksi jos karttaan haluaa lisätä pelkästään tekstin, voi pisteen piilottaa.



Kuva 14. Teksti lisääminen karttaan.

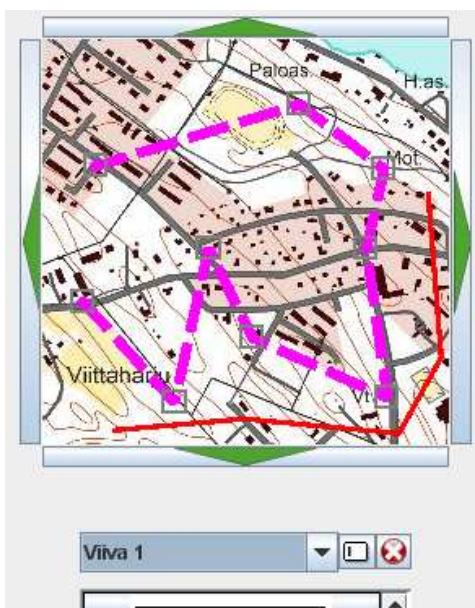
6.1.2 Viivojen piirtäminen

Viivojen piirto tapahtuu pitkälti samalla tavalla kuin pisteiden. Valitaan haluttu väri ja viivasymboli listasta (kuva 15), jonka jälkeen karttaa hiiren vasemmalla painikkeella napsauttelemalla piirretään haluttu viiva. Ensimmäisen painalluksen jälkeen kysytään viivan nimeä. Viivan piirtäminen lopetetaan napsauttamalla hiiren oikeaa painiketta kartalla.



Kuva 15. Viivakohteen työkalupaneeli.

Jo piirretyn viivan piirtosymbolia ja väriä voi myös muuttaa valitsemalla muokattava viiva yhdistelmäruudusta ja valitsemalla haluttu symboli ja väri työkalupaneelin valinnoista. Valittuna olevan viivan kulmapisteitä voi myös siirtää hiiren osoittimella vetämällä. Viivaan voi myös lisätä uusia pisteitä napsauttamalla sitä hiiren vasemmalla painikkeella. Valittuna olevan viivan kulmapisteiden ympärille piirretään harmaa neliö (kuva 16).



Kuva 16. Valittuna olevan viivan kulmapisteiden ympärille piirretään harmaa neliö.

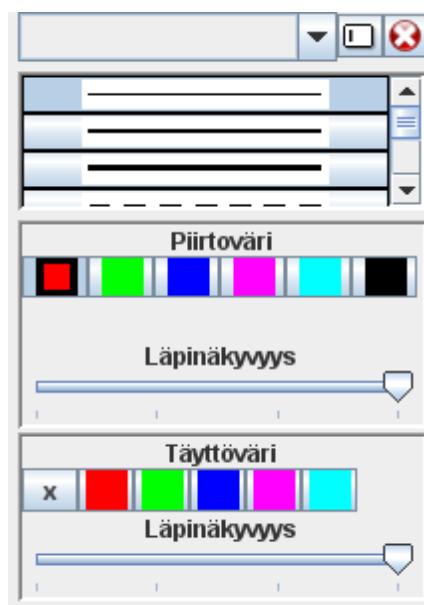
6.1.3 Alueiden piirtäminen

Alueen piirto toimii täysin samalla tavalla viivan piirron kanssa. Viivan ensimmäinen ja viimeinen piste vain yhdistetään, jolloin syntyy suljettu alue. Työkalupaneeliin on kuitenkin lisätty valinta täyttövärille (kuva 17).

6.1.4 Tietojen tallentaminen ja lataaminen

Ohjelman käyttäjä voi halutessaan tallentaa piirtämänsä kohteet palvelimelle ja ladataan ne sieltä. Kun käyttäjä napsauttaa tallennuspainiketta, piirretyistä kohteista muodostetaan tarkoitusta varten suunniteltuja olioita, jotka tallennetaan vektoriin. Tämä vektori sitten lähetetään palvelimelle. Koska myös palvelimena toimii Java-pohjainen

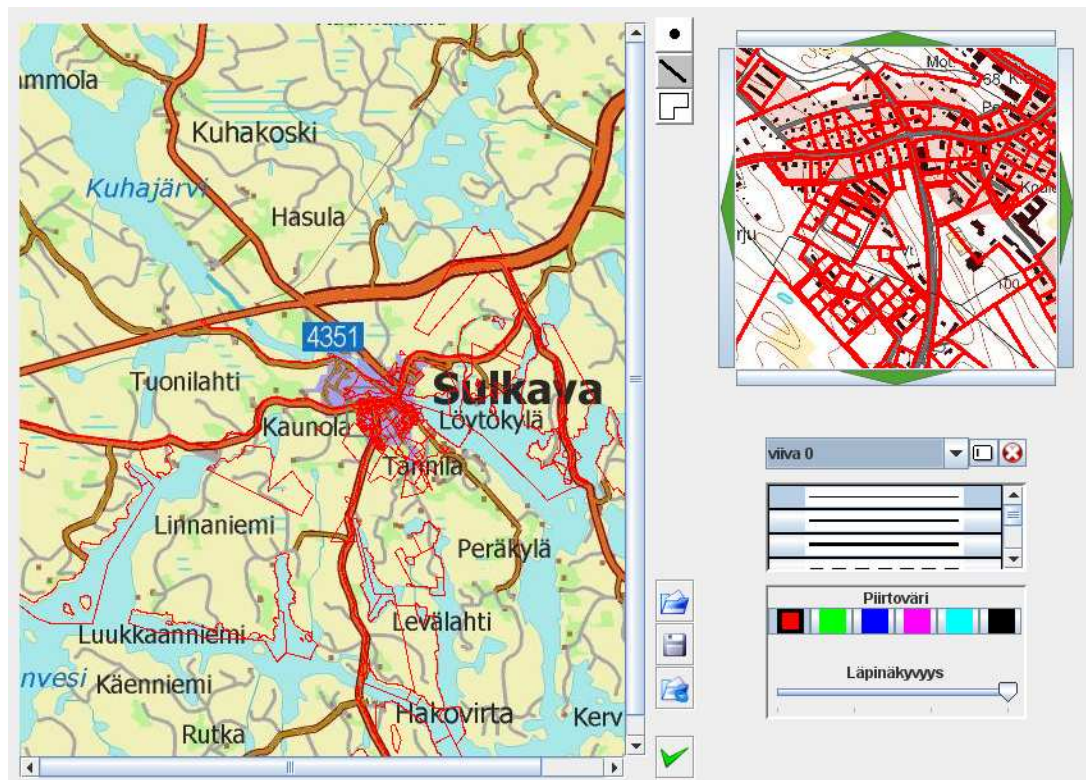
ratkaisu, tiedot voidaan siis lähettää suoraan olioina. Lähetettävät tiedot koostuvat kohteen nimestä, symbolin koodista, koordinaattitiedosta sekä kohteen tyyppin mukaan vaihtuvista lisäominaisuuksista. Tällaisia lisäominaisuuksia on esimerkiksi piirtoväri. Kohteen koordinaattitieto lähetetään WKT-muodossa. Tietoa ladattaessa palvelin muodostaa tietokannassa olevista tiedoista vastaavia olioita ja lähettää ne piirto-sovelmalle, joka sitten parsii saamansa tiedon ja lisää kohteet karttaan.



Kuva 17. Aluekohteen työkalupaneeli.

6.1.5 Tietojen tuominen tiedostosta

Työkaluun toteutettiin mahdollisuus tuoda tietoja kahdesta suositusta paikkatietoformaattista, ESRI-muototiedosto sekä MapInfo-siirtotiedosto. Kun käyttäjä napsauttaa tietojentuontipainiketta, avautuu tiedostojen valinta ikkuna, josta tuotava tiedosto voidaan valita. Ohjelma hyväksyy mif- ja shp-päätteiset tiedostot. Kun haluttu tiedosto on valittu, ohjelma lukee tiedoston läpi ja lisää löytyneet kohteet kartalle. Kohteiden tuonti on tällä hetkellä rajoittunut pisteisiin, viivoihin ja alueisiin. Jos tiedostosta löytyy soveltumattomia kohteita ne ohitetaan. Kuvassa 18 on ohjelmaan tuotu kiinteistörajoja ESRI-muototiedostosta.



Kuva 18. Ohjelmaan on tuotu kiinteistörajatiedot ESRI-muototiedostosta.

Ohjelmaan ei nykyisellään voi tuoda kohteisiin liittyviä ominaisuustietoja ja kaikille kohteille annetaan tuotaessa sama oletussymboli. Tuotavien kohteiden on oltava YKJ-koordinaatistossa. Tulevaisuudessa ohjelmaan on tarkoitus lisätä mahdollisuus myös ominaisuustietojen tuomiseen ja muiden koordinaatistojen käyttämiseen.

6.2 Palvelinsovelma

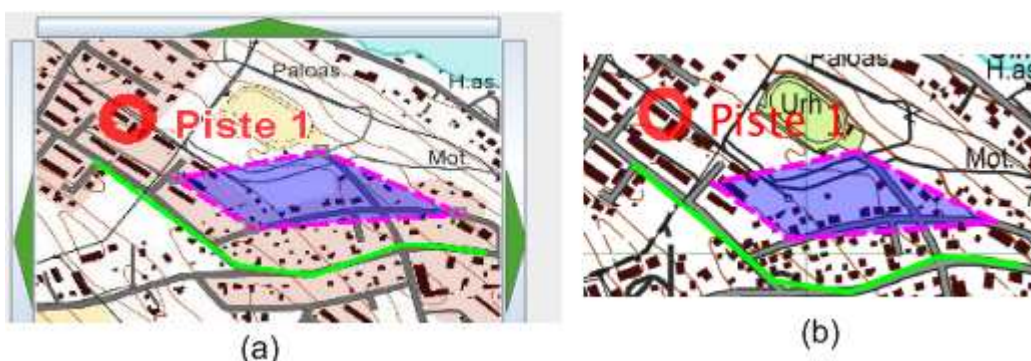
Tietojen tallentamista varten kehitettiin Java-palvelinsovelma. Käytännössä palvelinsovelma ottaa vastaan talletettavat tiedot, parsii ne ja tallentaa MySQL-tietokantaan. Tämän lisäksi se hakee tarvittavat tiedot tietokannasta ja lähettää ne piirto-sovelmalle. Palvelin hoitaa myös SVG-tiedoston tallentamisen.

Tietojen tallentamista varten palvelimelle luotiin tietokanta, jossa on sarakkeet kohteen nimelle, ominaisuustiedoille, symbolille sekä koordinaattitiedolle. Koordinaattitieto talletetaan GEOMETRY-tyyppiseen sarakkeeseen. Tämä mahdollistaa

paikkatietolaajennuksen tarjoamien lisäominaisuuksien hyödyntämisen tietoa käsiteltäessä. Kyseisestä sarakkeesta tehtiin myös indeksisarake.

Palvelin hoitaa myös PDFMapperille menevän SVG-tiedoston tallentamisen palvelimelle. SVG-tiedostoa luotaessa kohteiden mitat skaalataan metreiksi kartan mittakaavan mukaisesti. Symbolitaulussa mitat on ilmoitettu millimetreinä. Viivojen piirtotyylin määrittely Java:ssa [17] on täysin yhtenevä SVG:n kanssa, joten niiden siirtämiseksi riittää pelkkä skaalaaminen. Karttaan tulevat teksti on myös peilattava y-akselin suuntaisesti, koska SVG:ssä koordinaatisto lähtee vasemmasta yläkulmasta, kun taas karttakoordinaatit vasemmasta alakulmasta. Varsinainen piirtoalueen valmistelu tehdään PDFMapperissa, joten luotavassa SVG-tiedostossa on vain piirrettävät graafiset komponentit. Liitteessä 1 on esimerkki ohjelman tuottamasta pisteen, tekstin, viivan ja alueen sisältävästä SVG-koodista. Kuvassa 19 on esitetty piirretyt kohteet piirtosovelmassa sekä valmiilla PDF-kartalla.

Jatkossa olisi tarkoitus kehittää palvelimen paikkatieto-ominaisuuksia. Esimerkiksi lisätä mahdollisuus hakea syötettyjä tietoja halutulta alueelta sekä mahdollisuus alueiden pinta-alojen, viivojen pituuksien ja muiden vastaavien ominaisuuksien näyttämiseen. Tällöin on myös perehdyttävä muihin paikkatietolaajennuksia sisältäviin tietokantoihin, MySQL:n puutteiden vuoksi. Esimerkiksi PostGIS [18] voisi olla tutustumisen arvoinen.



Kuva 19. Esimerkkikohteet: (a) piirtosovelmassa; (b) PDF-kartalla.

7. Yhteenveto

Työn tarkoituksena oli kehittää verkkokaupan asiakkaalle työkalut paikkatiedon lisäämiseksi tilattavan kartan päälle. Tehtävää varten kehitettiin Java-sovelma, johon kuuluu piirtotyökalut paikkatiedon peruskomponenttien, pisteen, viivan ja alueen, piirtämiseksi sekä tuonti työkalut ESRI-muototiedostolle sekä MapInfo-siirtotiedostolle. Piirretyt kohteet talletetaan MySQL-tietokantaan, käyttäen hyväksi sen tarjoamaa paikkatietolaajennusta. Kohteet lisätään kartalle käyttäen PDFMapperin hyväksymää SVG-tiedostoa. Piirtotyökalusta pyrittiin tekemään mahdollisimman yksinkertainen ja helposti omaksuttava.

Alussa asetut toiminnalliset tavoitteet saavutettiin, mutta myös kehitettävää jäi. Järjestelmä on tällä hetkellä vielä koekäytössä ja sitä tullaan muuttamaan tarvittaessa saadun palautteen perusteella. Tulevaisuudessa järjestelmään olisi tarkoitus lisätä paikkatietohakuja ja työkaluja syötetyn paikkatiedon analysointiin. Järjestelmä hyväksyy nykyisellään vain YKJ-koordinaatistossa olevaa paikkatietoa, mutta tarvittavat työkalut koordinaatistomuunnoksille on myös tarkoitus lisätä myöhemmässä vaiheessa.

LÄHTEET

- [1] W3C. Scalable Vector Graphics (SVG) 1.1 specification [verkkajulkaisu]. 2003. [viitattu 9.11.2007]. Saatavilla: <http://www.w3.org/TR/SVG11/>
- [2] The Apache Software Foundation. Batik SVG Toolkit [The Apache Software Foundation www-sivuilla]. Päivitetty 11.7.2007. [viitattu 9.11.2007]. Saatavissa: <http://xmlgraphics.apache.org/batik/>
- [3] Maanmittauslaitos. Koordinaatti- ja korkeusjärjestelmät [Maanmittauslaitoksen www-sivuilla]. Päivitetty 8.11.2007. [viitattu 9.11.2007]. Saatavissa: <http://www.maanmittauslaitos.fi/default.asp?id=915>
- [4] Pohjois-Karjalan AMK, VirtuaaliAMK [Pohjois-Karjalan AMK:n www-sivuilla]. [viitattu 8.11.2007]. Saatavissa: <http://www.ncp.fi/koulutusohjelmat/metsa/PaikkatietoWWW/perusteet/koordinaatistot.htm>
- [5] Tekniikan sanastokeskus. Geoinformatiikan sanasto [pdf-tallenne]. [viitattu 8.11.2007]. Saatavissa: <http://www.tsk.fi/fi/info/GeoinformatiikanSanasto.pdf>
- [6] Raivio, T. Latitudi-longitudi -koordinaatistot ja GPS [verkkajulkaisu]. [viitattu 9.11.2007]. Saatavissa: <http://users.tkk.fi/~traivio/latlong.html>
- [7] Environmental Systems Research Institute, Inc. ESRI Shapefile Technical Description [pdf-tallenne]. [viitattu 26.9.2007]. Saatavissa: <http://www.esri.com/library/whitepapers/pdfs/shapefile.pdf>
- [8] ClicktyClick Software. Xbase Data file (*.dbf) [ClicktyClick Software:n www-sivuilla]. Päivitetty 18.8.2007. [viitattu 9.11.2007]. Saatavissa: <http://www.clicketyclick.dk/databases/xbase/format/index.html>

- [9] MapInfo. Appendix J: MapInfo Data Interchange Format [pdf-tallenne]. [viitattu 8.11.2007]. Saatavissa:
http://resource.mapinfo.com/static/files/document/1074660800077/interchange_file.pdf
- [10] Guting, Ralf. An Introduction to Spatial Database Systems. The International Journal on Very Large Data Bases, vol. 3, no. 4, pp. 357 - 399, 1994.
- [11] Open GIS Consortium, Inc. OpenGIS Simple Features Specification For SQL Revision 1.1 [pdf-tallenne]. [viitattu 26.9.2007]. Saatavissa:
http://portal.opengeospatial.org/files/?artifact_id=829
- [12] Open GIS Consortium, Inc. OpenGIS Implementation Specification for Geographic information – Simple feature access - Part 1: Common architecture [pdf-tallenne]. [viitattu 8.11.2007]. Saatavissa: http://portal.opengeospatial.org/files/?artifact_id=18241
- [13] MySQL AB. Chapter 17. Spatial Extensions [MySQL:n www-sivuilla]. [viitattu 8.11.2007]. Saatavissa: <http://dev.mysql.com/doc/refman/5.0/en/spatial-extensions.html>
- [14] Finkel, R. & Bentley, J. Quad trees a data structure for retrieval on composite keys. Acta Informatica, vol. 4, no. 1, pp. 1-9, 1974.
- [15] Bentley, J. Multidimensional binary search trees used for associative searching. Communications of the ACM, vol. 18, no. 9, pp. 462-472, 1975.
- [16] Guttman, A. R-Trees: A Dynamic Index Structure for Spatial Searching. ACM SIGMOD Conference, pp. 47-57, 1984.
- [17] Sun Developer Network, Learning Java 2D, Part 1 [Sun Developer Network www-sivuilla]. [viitattu 11.11.2007]. Saatavissa:
<http://java.sun.com/developer/technicalArticles/GUI/java2d/java2dpart1.html>

[18] Refrations Research. PostGIS : Home [Refrations Research www-sivuilla]. 2005.
[viitattu 9.11.2007]. Saatavissa: <http://postgis.refrations.net/>

Liite 1. Esimerkki ohjelman tuottamasta SVG-koodista

```
<polygon fill="rgb(0,0,255)" fill-opacity="0.39215687" stroke="rgb(255,0,255)"
stroke-opacity="0.98039216" stroke-width="10.5" stroke-dasharray="31.499998,21.0"
stroke-linecap="square" stroke-linejoin="bevel" points="3571840,6854124
3571972,6854025 3572329,6854054 3572092,6854179 "/>
<polyline fill="none" stroke="rgb(0,255,0)" stroke-opacity="0.98039216" stroke-
width="10.5" stroke-linecap="square" stroke-linejoin="bevel"
points="3571703,6854156 3571946,6853952 3572076,6853933 3572230,6853994
3572297,6853991 3572371,6853978 "/>
<polygon fill="none" stroke="rgb(255,0,0)" stroke-opacity="0.78431374" stroke-
width="18.75" stroke-linecap="square" stroke-linejoin="miter"
points="3571794,6854243 3571793,6854235 3571791,6854228 3571783,6854216
3571771,6854208 3571764,6854206 3571757,6854205 3571749,6854206
3571742,6854208 3571730,6854216 3571722,6854228 3571720,6854235
3571719,6854243 3571720,6854250 3571722,6854257 3571730,6854269
3571742,6854277 3571749,6854279 3571757,6854280 3571764,6854279
3571771,6854277 3571783,6854269 3571791,6854257 3571793,6854250
3571794,6854243 "/>
<g transform="translate(3571832.0,6854205.5) scale(1,-1)">/n<text text-anchor="start"
font-family="Verdana" font-size="75.0" fill="rgb(255,0,0)">Piste 1</text></g>
```