

KARTTAKÄYTTÖLIITTYMÄ

MapUI

LAHDEN AMMATTIKORKEAKOULU
Tietotekniikan koulutusohjelma
Ohjelmistotekniikka
Opinnäytetyö
Kevät 2009
Jarkko Sokeila

Lahden ammattikorkeakoulu
Tietotekniikan koulutusohjelma

SOKEILA, JARKKO:

Karttakäyttöliittymä
MapUI

Ohjelmistotekniikan opinnäytetyö, 41 sivua

Kevät 2009

TIIVISTELMÄ

Opinnäytetyössä tehtiin karttakäyttöliittymä. Karttakäyttöliittymän kautta puhelinoperaattorit voivat katsoa tietoja olemassa olevista tukiasemista. Työ rajattiin aluksi näyttämään kartalta tukiasemat sekä näiden pääsuuntauokset.

Työ toteutettiin suomen Ericssonille Lahden toimipisteessä. Lahdessa työskentelee alle 50 henkilöä; Jorvaksessa, Kirkkonummella työskentelee noin 950 henkilöä, ja koko Ericsson konsernissa noin 78750 henkilöä.

Karttapalveluita voidaan luoda monilla eri työkaluilla. Google tarjoaa ilmaisen ja helpon tavan lisätä karttapalveluja internetsivulle. Google Maps toimii Javascriptin avulla, ja sen ohjelmointiin on tarjolla aloittelijoille hieman sekava ohjelmointirajapinta.

Työ tehtiin Eclipse ohjelmointiympäristössä. Dynaamisen internetsivun luontiin käytettiin Google web toolkit -työkalua. GWT:n avulla internetsivuja voidaan ohjelmoida Java-kielellä.

Työn tuloksena saatiin toimiva karttakäyttöliittymä halutuilla toiminnoilla.

Avainsanat: Ericsson, karttakäyttöliittymä

Lahti University of Applied Sciences
Faculty of Technology

SOKEILA, JARKKO:

Map user interface

Bachelor's Thesis in Software Engineering, 41 pages

Spring 2009

ABSTRACT

The objective of this thesis was to make a map user interface. With the interface telephone operators can find information about base stations. Thesis was limited to showing only base stations and their main orientation. The thesis was made for the Lahti office of Ericsson.

Map services could be created with many different tools. Google offers a free and easy way to implement a map service to a web page. A Google map is made with Javascript. There is an application interface available for using it.

The work was done in the Eclipse integrated development environment. The google web toolkit was used to create the dynamic web page. With it you can code web pages using the Java language.

A functioning map user interface with desired functionality was obtained as a result of the work.

Key words: Ericsson, map user interface

SISÄLLYS

1	JOHDANTO	1
2	TIETOLIIKENNE	3
2.1	GSM ja UMTS matkapuhelinverkot	3
2.1.1	Historia	3
2.1.2	Arkkitehtuuri	4
2.2	ENIQ	6
2.2.1	Esittely	6
2.2.2	Arkkitehtuuri ja komponentit	7
2.3	OSS	8
3	KARTTAPALVELUT	9
3.1	GIS	9
3.2	Google Maps	10
3.3	Live search map	15
3.4	MapXtreme	17
4	WEB PORTAL	20
4.1	Kuvaus	20
4.2	Toiminta	21
5	MAPUI KARTTAKÄYTTÖLIITTYMÄ	24
5.1	Käyttöliittymä	24
5.2	Työkalut	24
5.2.1	Eclipse	24
5.2.2	Google web toolkit	26
5.2.3	Google Maps API	27
5.3	Työ	28
5.3.1	Projektin luonti	28
5.3.2	Käyttöliittymän suunnittelu ja toteutus	29
6	YHTEENVETO	39
	LÄHTEET	41

1 JOHDANTO

Karttakäyttöliittymät ovat lisääntymässä eri ohjelmistoissa. Tavallisimmin karttakäyttöliittymän voi löytää asunto- ja kiinteistökauppojen internetpalveluista. Kartalta voidaan nähdä, missä kiinteistö sijaitsee sekä muita tarvittavia tietoja. Karttakäyttöliittymät voivat helpottaa huomattavasti monia yrityksen asioita, esimerkiksi kuljetusalan yritykset voivat suunnitella eri kuljetuksia, kun he tietävät kuorma-autojen sijainnit. Sijainnit voidaan nähdä kartalta, mikäli kuorma-autoihin on asennettuna gps-laitteisto, joka lähettää sijaintinsa tiedot yrityksen palvelimille.

Mapui karttakäyttöliittymä työ tehtiin Suomen Ericssonille. Suomen yksikkö Lahdessa kehittää verkon optimointityökalua eri puhelinoperaattoreille. Työkalun avulla verkon toimintaa voidaan optimoida eri mittaustulosten perusteella. Operaattorit haluavat nähdä tietoja verkon toiminnasta, ja tätä varten heillä on käytössään verkkoportaaali. Portaalin kautta voidaan nähdä verkosta ja tukiasemista tietoja vain numeroina. Mapui:n avulla voidaan tukiasemien sijainti nähdä suoraan kartalta. Tämän lisäksi tukiasemista voidaan nähdä tarvittavia tietoja, kuten antennin suuntaus. Karttakäyttöliittymä voidaan toteuttaa usean karttapalvelun avulla. Tässä työssä vertaillaan kahta ilmaista sekä yhtä maksullista karttapalvelua, joita ovat Google Maps, Live search map ja MapXtreme.

Suomen Ericssonilla työskentelee noin 950 henkilöä. Valtaosa on töissä Jorvaksessa, Kirkkonummella, ja Lahdessa työskentelee pieni osasto, johon kuuluu alle 50 henkilöä. Lahden yksikkö sijaitsee keskustan läheisyydessä Askonkatu 9:ssä. Yhteensä Ericsson konsernissa on noin 78750 henkilöä, ja pääkonttori sijaitsee Tukholmassa. Ericssonin päätuotteena on puhelinverkkojen rakentaminen ja toimittaminen operaattoreille. Suomessa keskitytään viestintäteknologian tutkimukseen ja tuotekehitykseen. Merkittävä rooli on 3G-matkaviestintäratkaisuilla ja sitä seuraavilla teknologioilla.

Työn tarkoituksena on luoda interaktiivinen karttakäyttöliittymä, jonka avulla voidaan katsoa informaatiota tukiasemista ja luoda raportteja. Työn tutkimusongelma on, miten tehdä dynaaminen web-sovellus, johon on yhdistetty karttapalvelu.

Työ rajataan aluksi näyttämään tukiasemat kartalta sekä näyttämään tukiaseman pääsuuntaus. Lisäksi tukiasemista näytetään lisäinformaatiota silloin, kun hiiren kohdistin on tukiasemaa kuvaavan ikonin päällä. Lisätietoja näkee myös infoikkunasta, joka avautuu ikonia klikatessa.

Alussa kerrotaan GSM- ja UMTS-puhelinverkkojen historiasta sekä toiminnasta. Tällä annetaan kuva siitä, miten puhelinverkot toimivat. Lisäksi kerrotaan tarkemmin ENIQ-optimointityökalusta. ENIQ sisältää kaiken informaation, joka näytetään karttakäyttöliittymän kautta. Luvussa 3 kerrotaan eri karttapalveluiden ominaisuuksista. Neljännessä luvussa kerrotaan web portal -työkalusta, jolla voidaan analysoida informaatiota. Karttakäyttöliittymän kautta web portal -työkalulla voidaan luoda raportteja tukiasemista. Luvussa 5 kerrotaan karttakäyttöliittymän toteutuksesta sekä sen toteutuksessa käytetyistä työkaluista.

2 TIETOLIIKENNE

2.1 GSM ja UMTS matkapuhelinverkot

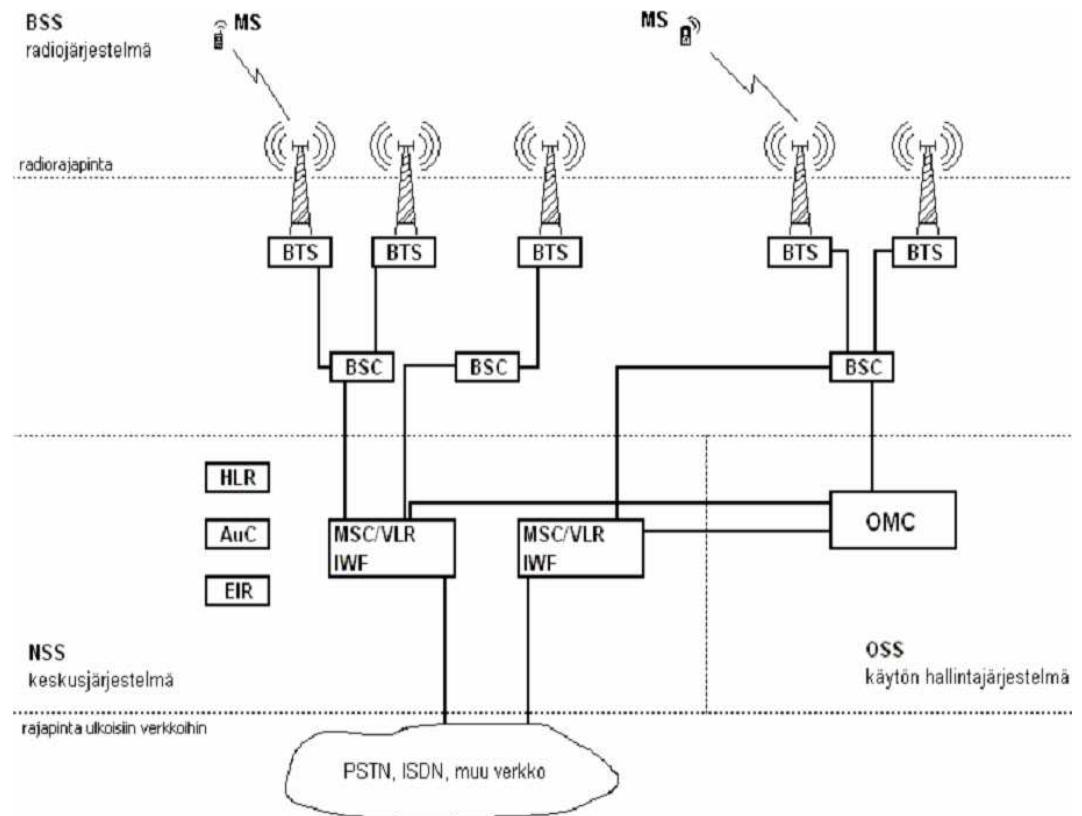
2.1.1 Historia

GSM-matkapuhelinverkko on syntynyt pohjoismaiden ja Hollannin telehallinnon aloitteesta kehittää yhteiseurooppalainen matkapuhelinverkko, joka toimii 900MHz:n taajuudella. GSM:n historian voidaan sanoa alkaneen vuonna 1982, jolloin aloite tehtiin. Euroopan telehallintojen yhteistyöelin CEPT perusti tämän jälkeen GSM (Groupe Special Mobile) työryhmän suunnittelemaan matkapuhelinjärjestelmää ja määrittelemään suosituksia. Työryhmä sai järjestelmän perusominaisuudet määriteltyä vuonna 1987. Tämän jälkeen kolmetoista maata allekirjoitti yhteisymmärryssopimuksen MoU:n (Memorandum of Understanding). Sopimus määrittää mm. verkkojen ja laitteiden toimitusajat ja käyttöönoton sekä muita suunnitelmia. Vuonna 1990 valmistui ensimmäisen vaiheen spesifikaatiot. 1.7.1991 oli määrä ottaa käyttöön spesifikaatioiden mukaiset verkot. Kuitenkin vain muutamat operaattorit saivat toimintansa käyntiin vuoden 1991 aikana viivytyksistä johtuen. Suomalaisista operaattoreista Sonera ja Radiolinja käynnistivät toimintansa, ja vuodesta 1992 alkaen kaupallinen toiminta alkoi kuitenkin kasvaa vähitellen. Matkapuhelinjärjestelmän kaupalliseksi nimeksi tuli GSM (Global system for Mobile communications). Väärinymmärrysten vuoksi GSM työryhmän nimi vaihdettiin SMG-työryhmäksi (Special Mobile Group). GSM on niin sanotusti toisen sukupolven järjestelmä (2G). Edeltävää NMT (Nordisk Mobiltelefon) -järjestelmää kutsutaan ensimmäisen sukupolven matkaviestimiksi. (Wikipedia 2009.)

UMTS (Universal Mobile Telecommunications System) on GSM:n seuraaja. Sen suunnittelu alkoi 1990-luvun lopulla, jolloin kansainvälinen televiestintäliitto (International Telecommunication Union ITU) käynnisti Future Public Land Mobile Telephony System (FPLMTS) -projektin. Projektin tarkoituksena oli

kolmannen sukukupolven järjestelmän ratkaisumallien laatiminen (3G). Japanissa avattiin ensimmäinen 3G verkko vuonna 2001, ja yleiseen käyttöön verkko tuli vuonna 2005.

2.1.2 Arkkitehtuuri



KUVIO 1. GSM-verkon arkkitehtuuri

GSM-verkko voidaan jakaa kolmeen osaan: keskusjärjestelmään NSS (Network and Switching Sub-system), tukiasemajärjestelmään BSS (Base Station Sub-system) ja käytön hallintajärjestelmään OSS (Operations sub-system).

Arkkitehtuuri on kuvattu kuviossa 1.

Järjestelmä toimii siten, että puhelun alkaessa matkapuhelin ottaa yhteyden tukiaseman BTS (Base Transceiver Station) ja tukiasemaohjaimen BSC (Base

Station Controller) kautta matkapuhelinkeskukseen MSC (Mobile Switching Centre). Tämän jälkeen MSC yhdistää puhelun oikeaan puhelimeen joko oikean BSC:n ja BTS:n tai muun verkon kautta.

Lyhyesti kerrottuna BSS:n tehtävänä on yhdistää matkapuhelin (kuvassa MS = Mobile Station) keskusjärjestelmään. Tukiasemajärjestelmässä on useita tukiasemia ja näitä hallitsevia ohjaimia. Yksi ohjain voi kontrolloida useita tukiasemia, mutta yksi tukiasema ei voi kuulua usealle ohjaimelle.

Tukiasema koostuu laitteistotilasta ja tukiasemalaitteista. Se voi sijaita erillisessä rakennuksessa esikaupunki tai maaseutualueilla tai kerrostalon vintillä tai kellarissa kaupunkialueilla. Lisäksi tukiasemassa on tyypillisesti useita (TRX Transceiver) lähetin- ja vastaanottoyksiköitä ja näiden tehonlähteet, kaapeleita, masto ja antenneja.

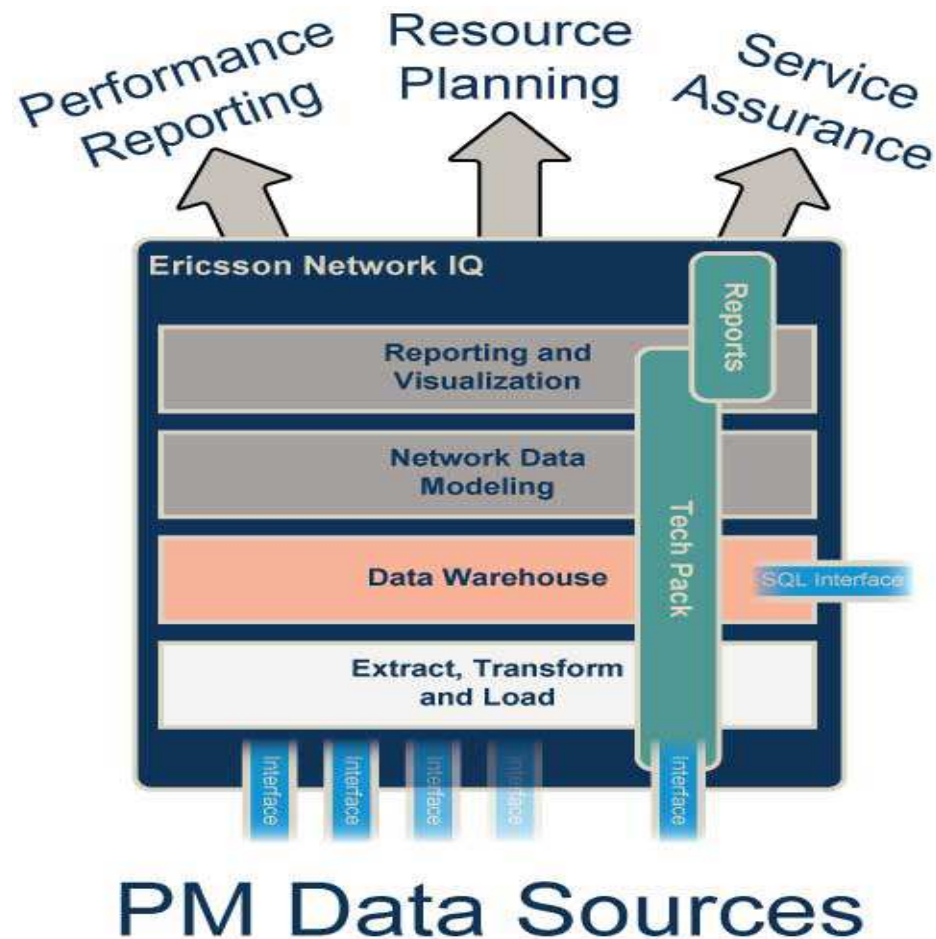
Tukiasemaohjaimen päätehtävänä on oman alueensa radioresurssien hallinta. Tällöin matkapuhelinkeskus voi käyttää kapasiteettinsa sen varsinaiseen tehtävään eli puheluiden läpikytöntään.

Matkapuhelinkeskus sisältää eri rekistereitä ja yhteensopivuustoiminnot (IWT, interworking function). IWT:n avulla voidaan luoda puhelinyhteyksiä GSM-verkon ulkopuolelle kuten tavalliseen puhelinverkkoon. MSC:n tärkein tehtävä on oman alueen puhelujen kytkentä, purku ja ylläpito. Rekisterit keräävät erilaisia tietoja kuten tilaaja- ja laskutietoja. Verkon toiminnan kannalta pakollisiin rekistereihin kuuluu kotirekisteri HLR (Home location Register) ja vierasrekisteri VLR (Visitor Location Register).

2.2 ENIQ

2.2.1 Esittely

ENIQ (Ericsson Network IQ) on Ericssonin kehittämä puhelinverkon optimointityökalu, joka on tarkoitettu puhelinoperaattoreille. Se toimii ympäristössä, joka koostuu erilaisista verkkoteknologioista ja tietolähteistä. Optimointi sisältää verkkotietojen keräämisestä useista verkkoelementeistä, tiedon säilyttämisestä ja prosessoinnista sekä raporttien tarjoamisesta loppukäyttäjälle.



KUVIO 2. ENIQ arkkitehtuuri

ENIQ-järjestelmä koostuu itse järjestelmälustasta sekä teknologiapaketeista (Technology Package tai TP, kuvio 2). Teknologiapaketit on suunniteltu keräämään suorituskykytietoja puhelinverkosta. Ne tarjoavat loppukäyttäjille helpon ja joustavan pääsyn suorituskyky- ja topologiatietoihin. ENIQ:n arkkitehtuuri on suunniteltu siten, että tulevaisuudessa uusi tieto on loppukäyttäjän saatavilla heti, kun se vain on saatavilla verkosta. ENIQ:n ohjelmistoarkkitehtuuri koostuu standardeista patentoimattomista komponenteista.

2.2.2 Arkkitehtuuri ja komponentit

Tietovarasto, jossa ENIQ säilyttää kaikkea tietoa, on rakennettu Sybase IQ - tiedonhallintajärjestelmän päälle. Sybase IQ on tehokas tietovarastoratkaisu. Se pystyy käsittelemään suurta määrää informaatiota nopeasti. Nopeus on saavutettu sillä, että kaikki tieto tietokannassa on aina indeksoitu ja pakattu mahdollisimman pieneen tilaan (Sybase 2009). Nopeutta tarvitaan, koska tietokannassa on satoja miljoonia rivejä.

Suuren tietomäärän hallinta on usein haastavaa. Tietojen poisto voi olla hidasta ja ajan kuluessa tieto sirpaloituu. ENIQ:ssä on tietovaraston hallintakomponentti (DWH Management), joka käsittelee tietoja niin, että edellämainituista haasteista johtuvat ongelmat vähenevät.

ETL (Extract, Transform and Load) -prosessi vastaa tiedon lataamisesta, muuntamisesta ja tallentamisesta tietokantaan. Kaikki tieto tulee ENIQ-järjestelmään OSS-serveereiltä. Tämän jälkeen tieto puretaan adapterien avulla ENIQin ymmärtämään muotoon. Tieto voi tulla useassa eri tiedostomuodossa, joten ENIQ:ssä on useita eri adaptoreita eri tiedostomuodoille. Tiedon purun jälkeen tieto ladataan tietokantaan (kuvio 2).

ENIQ tarjoaa graafisen valvonta käyttöliittymän (Administration UI), jota käytetään mm. järjestelmän tilan tarkkailuun, tiedon virtauksen ja latauksien tarkkailuun sekä tiedon verifioimiseen. Käyttöliittymää voi käyttää

internetselaimella. Operaattoreiden ensisijainen käyttöliittymä on ENIQ Web Portal, jota käsitellään luvussa neljä.

2.3 OSS

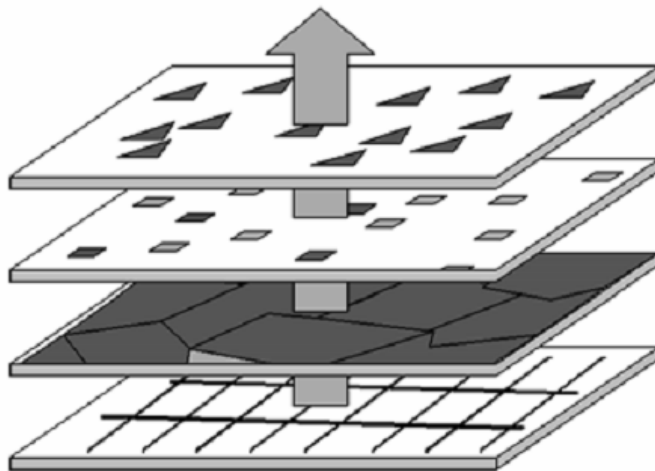
Käytönhallintajärjestelmä OSS on yhteydessä tukiasemajärjestelmän ja keskusjärjestelmän kanssa. Sen tärkeimpiä tehtäviä ovat verkon käyttö ja kunnossapito, matkapuhelimien hallinta ja tilaajatietojen hallinta, esimerkiksi laskutus. OSS:n toteuttamista ei ole tarkasti määritelty GSM:n spesifikaatioissa. Toteutus on jätetty operaattoreiden tehtäväksi. BSS:n ja NSS:n välillä oleva rajapinta on kuitenkin tarkkaan spesifioitu. OSS koostuu yhdestä tai useammasta käyttö- ja kunnossapitokeskuksesta (OMC, Operations and Maintenance Centre). OMC:n avulla verkkoelementeistä kerätään tietoja, siirretään tiedostoja sekä analysoidaan ja tallennetaan mittaustuloksia. OMC:n tehtävänä on mittaustuloksia tarkkailemalla operoida verkkoa. OMC:n kautta tapahtuu myös verkon muutosten hallinta. Erilaisten parametrien muutokset, ohjelmistojen päivitykset ja verkon laajennokset voidaan hallita ”etätyöskentelynä” OMC:n kautta. Tämä tekee esimerkiksi verkon laajennokset hyvin kustannustehokkaiksi.

OMC:n kautta kerätään verkon toiminnasta ja käyttöasteesta kertovia mittaustietoja. Nämä ovat operaattoreille erittäin tärkeitä. Mittaustietojen avulla voidaan seurata GSM-verkon kuormitusta ja purkukodeja. Purkukoodien avulla voidaan selvittää ongelmatilanteiden todellinen syy. Esimerkiksi puhelinyhteyden hallitsemattoman katkeamisen syy voidaan näin selvittää ja korjata. Yhteyden katkeaminen voi johtua mm. laitteistoviasta tai radioverkkosuunnittelussa tapahtuneesta virheestä. OSS lähettää näitä mittaustietoja ENIQ:lle käsiteltäväksi ja säilöttäväksi.

3 KARTTAPALVELUT

3.1 GIS

Paikkatietojärjestelmällä (Geographical Information System GIS) tarkoitetaan järjestelmää, jonka avulla paikkatietoa voidaan esittää, analysoida, hallita ja tallentaa. Paikkatietojärjestelmä ei ole yksi kokonaisuus, vaan se koostuu useasta eri osa-alueesta. Järjestelmä koostuu laitteistosta, ohjelmistoista, paikkatiedot sisältävästä aineistosta ja käyttäjistä. Aineistoa voidaan säilöä esimerkiksi tietokannoissa tai tiedostoissa. Paikkatiedot näytetään käyttöliittymän kautta, joka yleensä on graafinen, mutta käyttöliittymä voi olla myös muunkinlainen. Yleensä käyttäjälle näytetään järjestelmän tuottama kartta. Paikkatietojärjestelmän tuottama kartta koostuu kerroksista, joita piirretään päällekkäin. Alimpana on yleensä ilma- tai satelliittikuva, jonka päälle voidaan piirtää esimerkiksi tiekartasto. Muilla kerroksilla voi olla esimerkiksi kuntoilureittejä tai rakennukset ja metsät. Kerrokset ovat järjestelmän tietokannassa vektori- tai rasterimuodossa. Kerrokset ovat myös läpinäkyviä, jotta useita eri aineistoja voidaan esittää kartalla (kuvio 3).



KUVIO 3. Kerrosperiaate

Maailmassa on useita eri koordinaattijärjestelmiä. Näiden järjestelmien erona on lähinnä tarkkuus. Eri koordinaattijärjestelmiä ovat mm. WGS84, ITRS, ETRS89, EUREF-FIN, NAD83. WGS84-järjestelmää käytetään web-karttapalveluissa ja GPS satelliittipaikannuslaitteissa. Sen toteuttamiseen on käytetty kahtakymmentä kiintopistettä, joten sen tarkkuus ei riitä suurta tarkkuutta vaativiin tehtäviin. WGS84-järjestelmän koordinaatit voidaan esittää leveys- ja pituusasteina. Esimerkiksi Lahden koordinaatit ovat 60.981° N 25.655° E. Järjestelmä on kolmiulotteinen, joten sen avulla on mahdollista ilmoittaa myös korkeus.

Koordinaattijärjestelmiin liittyy oleellisesti termi geokoodaus. Termillä tarkoitetaan osoitteen muuntaa maantieteellisiksi koordinaateiksi, ja käänteisgeokoodaus tarkoittaa koordinaattien muuntamista osoitteeksi tai paikannimeksi.

Aiemmin koordinaattijärjestelmät ovat olleet alueellisia, eri maiden maanmittausorganisaatioiden ylläpitämiä. Maailmanlaajuisten koordinaattijärjestelmien luomisen mahdollisti vasta avaruudesta käsin tehdyt mittaukset (Wikipedia 2009).

3.2 Google Maps

Google Maps on ilmainen karttapalvelu, joka tarjoaa dynaamisia ja interaktiivisia karttoja (kuvio 4). Palvelulla on monia ominaisuuksia, ja kartalta voidaan etsiä haluttuja kohteita tai luoda reittitietoja kohteesta toiseen. Karttakuvia voidaan katsella joko tavallisena kartta-, satelliitti-, maasto-, katu- tai liikennekarttanäkymänä.



KUVIO 4. Google Maps

Lintuperspektiivi tarjoaa käyttäjälle joko satelliitti- tai ilmakuvia. Kaikkialta maailmasta ei löydy satelliittikuvia, koska kuvien ottaminen on kallista. Tästä syystä ei ole kannattavaa ottaa kuvia vähän asutetuista maa-alueista. Ilmakuvat ovat halvempia ja helpompia ottaa tällaisilta alueilta. Kuviossa 5 on ilmakuva Helsingin keskustasta.



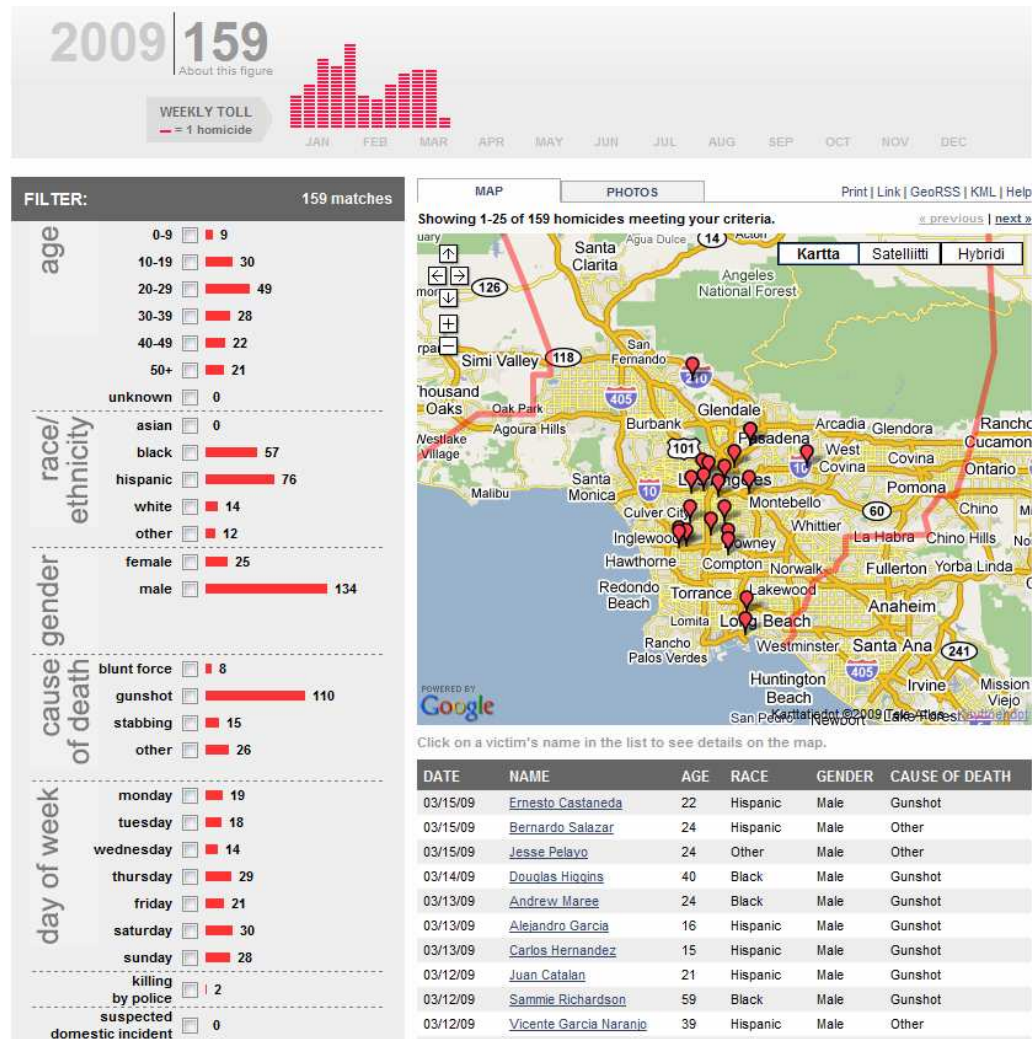
KUVIO 5. Ilmakuva Helsingistä

Suurimmissa kaupungeissa on mahdollisuus katsoa karttaa katunäkymän avulla. Tämän ominaisuuden avulla voidaan tutustua ympäristöön kulkemalla virtuaalisesti kaduilla ja liikkuminen tapahtuu painamalla katukuvissa näkyviä nuolia. Tiet ja kadut, joilla tätä ominaisuutta voidaan käyttää, on merkitty sinisellä reunuksella. Kuviossa 6 on katunäkymä Googlen pääkonttorin edustalta.



KUVIO 6. Katunäkymä

Google tarjoaa ilmaisen ohjelmointirajapinnan, jolla voidaan luoda omia karttoja ja lisätä ne omille internetsivuille. Rajapinta on tehty JavaScript-ohjelmointikielellä. Google Maps-karttapalvelun avulla voidaan luoda monia erilaisia sovelluksia. Suomalaisista sovelluksista mainittakoon Helsingin kaupungin liikennelaitoksen karttapalvelu, josta voidaan reaaliaikaisesti seurata raitiovaunujen ja linjautojen sijaintia kartalta. Toinen suomalainen palvelu on tilannehuone, josta taas näkyvät palo- ja ensivastehälytykset viimeiseltä vuorokaudelta. Ulkomaalaisia sovelluksia ovat muun muassa Los Angelesin henkirikoksien seurantapalvelu (kuvio 7) ja Mc Donaldsin Big Mac hampurilaisten hintaseuranta globaalisti.



KUVIO 7. Los Angeles henkirikoksien seurantapalvelu

Kartoista voidaan tehdä myös staattisia, mikä tarkoittaa sitä, että karttaa ei voi liikutella. Tällöin voidaan esimerkiksi yrityksen sivuille lisätä kuvan avulla oma sijainti ilman erillisiä Google Maps-toimintoja.

Google on tehnyt myös työpöytäsovelluksen nimeltään Google Earth (kuvio 8), jonka avulla voidaan katsoa karttatietoja kolmiulotteisella maapallolla. Tätä voidaan verrata esimerkiksi pöydällä olevaan karttاپالloon. Google Earth tarjoaa monipuolisemmat ominaisuudet kuin Google Maps. Internetselaimille on myös olemassa laajennokset, joilla kolmiulotteista maapalloa voidaan katsoa myös selaimessa.



KUVIO 8. Google Earth

Karttapalvelun avulla voidaan suorittaa erilaisia hakuja kartalta. Paikkatietoja voidaan etsiä postinumeron, kadun nimen, kaupungin nimen, lähiympäristön ja koordinaattien perusteella. Google maps karttapalvelussa käytettävät karttatiedot tulevat seuraavilta yrityksiltä: TeleAtlas, DigitalGlobe ja MDA Federal.

Karttatiedot ovat yhdestä kolmeen vuotta vanhoja.

Googella on erinomainen yhteisö, josta voi kysyä neuvoja erilaisissa ongelmatilanteissa. Lisäksi internetistä löytyy useita sivustoja, jossa on esimerkkejä ja oppaita Google Maps:n käytöstä. Google Maps ohjelmointirajapinta (API) saattaa olla vasta-alkajille hieman sekava. Kuviossa 9 on esimerkki javascript-koodista, jossa luodaan kartta sekä 10 merkkiä satunnaiseen paikkaan. Rivillä 3 luodaan kartta html elementin paikalle, jonka id on ”map_canvas”. Rivillä 4 määritellään kartan keskipiste. Riveillä 7-11 määritellään kartan näkyvän alueen koko. Rivillä 12 alkavassa for silmukassa luodaan merkit satunnaiseen paikkaan. Rivillä 13 määritellään satunnainen piste näkyvältä kartta-alueelta. Rivillä 15 karttaan lisätään uusi merkki edellä määritettyyn pisteeseen.

```

1 function initialize() {
2   if (GBrowserIsCompatible()) {
3     var map = new GMap2(document.getElementById("map_canvas"));
4     map.setCenter(new GLatLng(37.4419, -122.1419), 13);
5
6     // Add 10 markers to the map at random locations
7     var bounds = map.getBounds();
8     var southWest = bounds.getSouthWest();
9     var northEast = bounds.getNorthEast();
10    var lngSpan = northEast.lng() - southWest.lng();
11    var latSpan = northEast.lat() - southWest.lat();
12    for (var i = 0; i < 10; i++) {
13      var latLng = new GLatLng(southWest.lat() + latSpan * Math.random(),
14                              southWest.lng() + lngSpan * Math.random());
15      map.addOverlay(new GMarker(latLng));
16    }
17  }
18 }

```

KUVIO 9. Google Maps javascript -koodia

3.3 Live search map

Live search map on Microsoftin vastine Google Map s-karttapalvelulle (kuvio 10). Käytettävyydeltään Live search map on yhtä helppo kuin Google Maps. Kartan liikuttaminen tapahtuu täsmälleen samoin tavoin kuin Googlen kartalla.

Karttanäkymiä on Live search map:ssa vakiona useampia kuin Google Maps:ssa. Yksi mielenkiintoinen näkymä on linnun silmä, jossa samaa aluetta voidaan katsella neljästä eri kuvakulmasta. Vakiona on myös 3D-näkymä, jolla voidaan katsella Microsoft Virtual Earth virtuaalista karttapalloa. Vaikka tämä on vakiona valittavissa, niin se tarvitsee kuitenkin oman selainlaajennokset toimiakseen. Paikkahakuja kartalta voidaan tehdä postinumeron, kadun nimen ja kaupungin nimen perusteella.



KUVIO 10. Live search map

Microsoftin karttapalvelussa käytettävät karttatiedot tulevat seuraavilta yrityksiltä: NAVTEQ, Intermap, Pictometry ja NASA. Live search map:ssa käytettävät karttatiedot ovat noin kahdesta kolmeen vuotta vanhoja.

Vaikka Microsoftin karttapalvelu on ilmainen, on sillä myös huonoja puolia: Live search map:illa on huonona puolena ohjelmointirajapinta. Microsoft tarjoaa oman rajapintansa MSDN:n (Microsoft Developer Network) kautta. MSDN sivusto on erittäin sekava, erityisesti vasta-aloitteleville ohjelmoijille. Sivustolta löytyy ohjelmointirajapinnat useille eri Microsoftin tuotteille, ja pelkästään tämän vuoksi oikean rajapinnan löytäminen on vaikeaa. Kokeneemmillakin ohjelmoijilla saattaa olla vaikeaa käyttää haluttua rajapintaa. Hyvänä puolena sivustossa on se, että monissa metodien kuvauksissa on kuitenkin esimerkkikoodi, joka kertoo, miten luokkaa ja metodeita käytetään (kuvio 11). Esimerkkikoodi ei kuitenkaan aina auta, mikäli ei täysin hallitse kyseistä kirjastoa. Tämä saattaa olla myös ainoa esimerkki metodin toiminnasta, sillä erilaisia oppaita Microsoftin rajapintojen toiminnasta on harvassa. Lisäapua ohjelmointiin voi löytyä Microsoftin kirjoista. Microsoft on julkaissut monia kirjoja eri ohjelmointikielille sekä -tekniikoille.

Live search map:n ohjelmointiin tosin ei tällä hetkellä ole olemassa painettua opasta.

Kuviossa 11 luodaan kartta sekä yksi merkki, ja rivillä 3 luodaan kartta html elementtiin ”myMap”. Kartta ladataan näytölle LoadMap metodin avulla rivillä 4, ja AddPin funktiossa luodaan yksi merkki keskelle karttaa. GetCenter metodilla saadaan kartan keskipiste selvitettyä. Rivillä 12 keskipiste tallennetaan muuttujaan, ja rivillä 13 luodaan merkki pisteen määrittelemään paikkaan.

```

1 function GetMap()
2 {
3     map = new VEMap('myMap');
4     map.LoadMap(LA, 14, VEMapStyle.Road, false, VEMapMode.Mode2D, true, 1);
5
6     AddPin();
7 }
8
9 function AddPin()
10 {
11     // Add a new pushpin to the center of the map.
12     pinPoint = map.GetCenter();
13     map.AddPushpin(pinPoint);
14 }

```

KUVIO 11. Live search map esimerkikoodi

Microsoftilla ei ole lähellekään yhtä suurta yhteisöä verrattuna Googleen, josta voi saada neuvoa ongelmatilanteissa. Huonona puolena on myös se, että Google Maps tukee virallisesti enemmän selaimia kuin Live search map. Esimerkiksi Opera-internetselain ei ole lainkaan Live search map:n tuettujen selainten listalla.

3.4 MapXtreme

MapXtreme on vertailtavista karttapalveluista ainoa maksullinen karttapalveluita tarjoava vaihtoehto. Sen on luonut MapInfo-yritys, joka perustettiin vuonna 1986. MapXtremestä on olemassa kaksi versiota eri ohjelmointikielillä. Karttasovelluksia voidaan kirjoittaa, joko .NET- tai Java- kielillä. Toisin kuin

Google Maps tai Live search map karttapalveluilla MapXtremen avulla voidaan luoda myös työpöytäsovelluksia. Java-kielinen versio mahdollistaa käyttöjärjestelmästä riippumattomien sovellusten tekemisen.

Koska MapXtreme sovelluksia voidaan tehdä puhtaasti Java- tai .NET- kielillä, on sovellusten suorituskyky huomattavasti parempi kuin Javascript-sovellusten. Esimerkiksi säikeistämisen avulla voidaan tehdä asioita, joita Javascript-kielillä ei voida tehdä lainkaan.

Käytettävyydeltään karttasovelluksista voidaan tehdä yhtä helppokäyttöisiä kuin Google Maps tai Live search map-sovelluksista. Muilta ominaisuuksiltaan MapXtreme on monipuolisempi. Käyttäjät voivat esimerkiksi luoda omia karttojaan, ja karttoja voidaan myös muokata monipuolisemmin. Esimerkiksi kirkkautta ja kontrastia voidaan säätää, ja kuviossa 12 on esimerkki MapXtreme kartasta.



KUVIO 12. MapXtreme

Java kielisen version ohjelmointirajapinta on erittäin selkeä. API on tehty Javadoc-työkalun avulla. Koska MapXtreme on maksullinen, ei internetistä löydy esimerkkejä tai ohjeita sovelluksien tekemiseen. Kun ostetaan tuote, mukana tulee ohjelmiston asennus cd-levy sekä kehittäjäopas. Oppaassa on joitakin koodiesimerkkejä, mutta kaiken kaikkiaan ohjeistusta löytyy melko vähän.

Kuviossa 13 on esimerkki MapXtremen Java koodista. Rivillä 3 luodaan uusi karttaobjekti. Rivillä 6 luodaan uusi näyttöpiste hiiren x- ja y-koordinaateista. Näyttöpiste muutetaan kartan ymmärtämäksi pisteeksi rivillä 9. Rivillä 12 kartta keskitetään tähän pisteeseen.

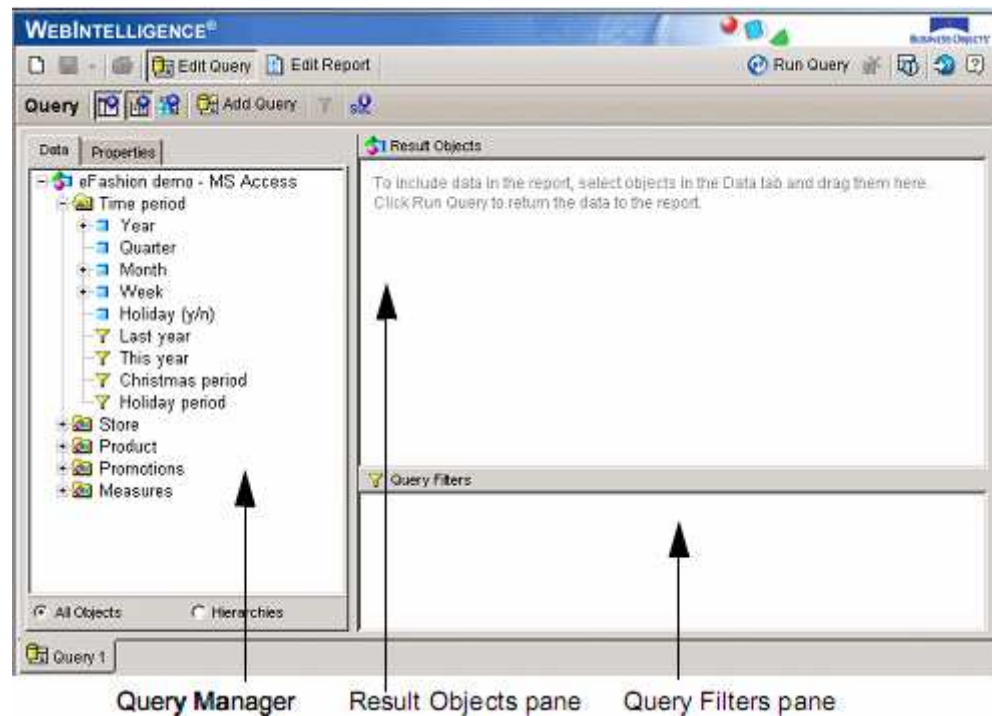
```
2// Initialize MapJ object
3myMap = new MapJ();
4
5// Create the screen point
6DoublePoint screenpoint = new DoublePoint(event.getX(), event.getY());
7
8// Create the real world point
9DoublePoint worldpoint = myMap.transformScreenToNumeric(screenpoint);
10
11// Set the center of the map
12myMap.setCenter(worldpoint);
```

KUVIO 13. MapXtreme-esimerkkikoodi

4 WEB PORTAL

4.1 Kuvaus

Web portal pohjautuu BusinessObjects Web intelligence XI-työkaluun. Työkalun avulla yritys voi analysoida keräämäänsä informaatiota. Web intelligence tarjoaa helppokäyttöisen käyttöliittymän, jolla voidaan tehdä tietokannasta hakuja jokapäiväisillä, työssäkäytettävillä termeillä (kuvio 14). Lisäksi hakujen perusteella voidaan luoda raportteja.



KUVIO 14. Web Intelligence Rich Client

Yrityksen dataan päästään käsiksi semanttisen kerroksen avulla (semantic layer). Kerrosta kutsutaan perusjoukoiksi (BusinessObjects universes). Semanttinen kerros on näkymä yrityksen dataan, ja se koostuu yhdestä tai useammasta perusjoukosta. Perusjoukot perustuvat yrityksen sisällä käytettäviin termeihin, joten käyttäjien ei tarvitse tietää tietokantojen sisäistä toimintaa. Web intelligence-

työkalu käyttää hyväkseen näitä perusjoukkoja, kun käyttäjä tekee hakuja tietokannasta. Jotta ymmärtää, miten semanttinen kerros toimii, täytyy käyttäjän tietää seuraavat termit.

- objekti (object)
- luokka (class)
- ennalta määrätty kyselysuodin (predefined query filter)
- perusjoukko (universe)

Objektit ovat perusjoukon elementtejä, jotka yhdistävät tietyn datajoukon relaatiotietokantaan. Jokainen objekti perusjoukossa on määritelty jokapäiväisellä työssäkäytettävällä termillä. Esimerkiksi kuviossa 14 objekteja ovat vuosi, neljännesvuosi, kuukausi jne.

Luokat ovat loogisia ryhmiä objekteille. Kuviossa 14 edellämainitut objektit kuuluvat luokkaan time period. Muita luokkia kuviossa ovat store, product jne. Luokat voivat sisältää myös muita luokkia.

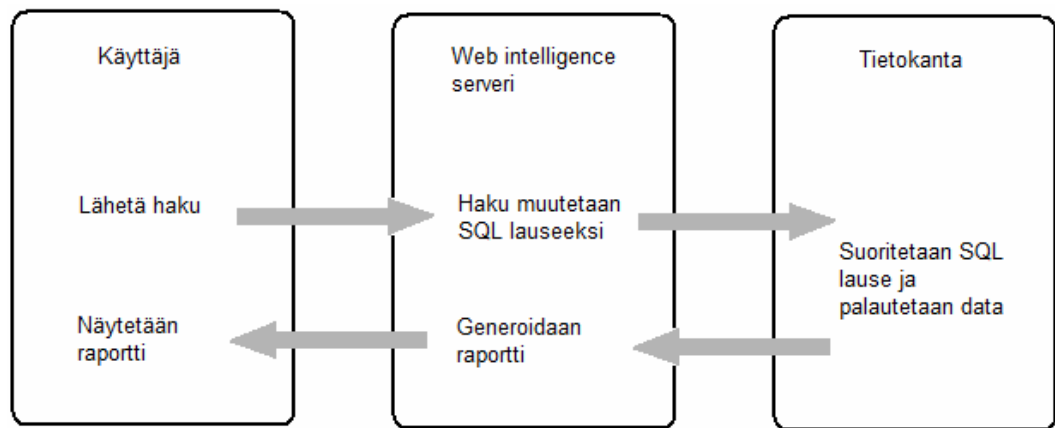
Ennalta määritellyt kyselysuotimet rajoittavat, mitä informaatiota palautetaan, kun käyttäjä tekee hakuja tietokannasta. Kuviossa 14 suotimia ovat objektit last year, this year jne.

Perusjoukot sisältävät samaan aihealueeseen kuuluvia objekteja. Erilaisia perusjoukkoja voivat olla esimerkiksi myynti, henkilöstö ja talous. Kuvioista 14 voidaan nähdä yksi perusjoukko nimeltään eFashion. Perusjoukot luo suunnittelija, joka tuntee yrityksen tietokannan. Kun käyttäjä tekee kyselyjä portaalin kautta, valitaan ensin perusjoukko, joka rajaa haut haluttuun informaation.

4.2 Toiminta

Web intelligence työkalu toimii siten, että käyttäjä luo ensin dokumentin ja valitsee tietolähteen, joka sisältää halutun informaation. Tietolähde on esimerkiksi

Perusjoukko (universe). Tämän jälkeen käyttöliittymän kyselypaneelin avulla tehdään kysely valitsemalla halutut objektit, jotka dokumentissa halutaan nähdä. Lisäksi valitaan tarvittaessa kyselysuotimet rajaamaan hakua. Kun kysely on valmis, lähetetään se Web intelligence -palvelimelle. Palvelin muuttaa kyselyn tietokannan ymmärtämäksi Sql-lauseeksi. Sql lause lähetetään tietokantaan, joka suorittaa lauseen ja palauttaa tietokannasta halutut rivit. Web intelligence serveri generoi palautetusta informaatiosta raportin ja palauttaa sen käyttäjälle. Dokumentti voi sisältää useita eri raportteja, ja raportit voidaan tehdä useilla eri kyselyillä. Toiminta on kuvattu kuviossa 15.



KUVIO 15. Web intelligence-toiminta

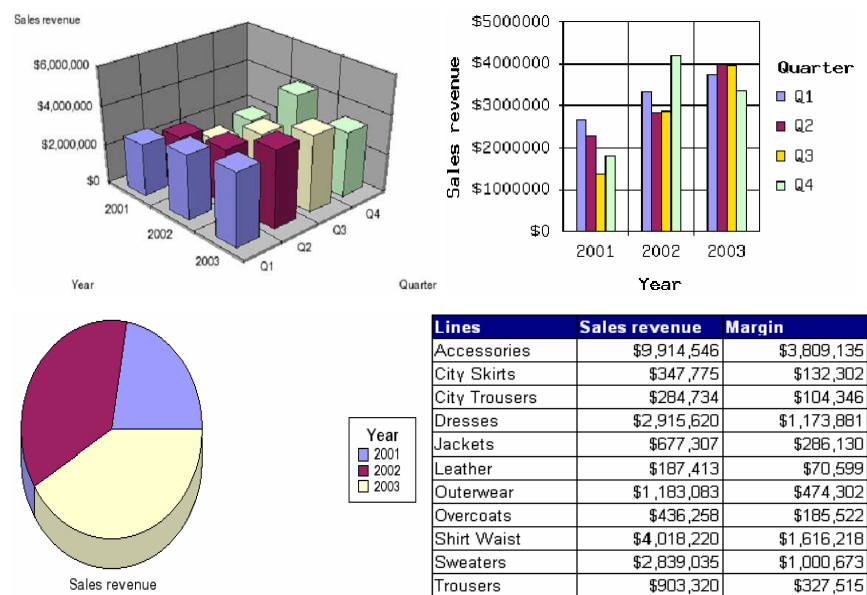
Kyselyitä voidaan tehdä neljällä eri käyttöliittymällä. Kolme niistä voidaan käyttää internetselaimella, ja neljäs käyttöliittymä on asennettava työpöytäsovellus.

Java Report Panel on suunniteltu niille, jotka tarvitsevat joustavuutta monimutkaisten hakujen tekemiseen, raporttien ulkoasun suunnitteluun sekä kaavojen ja muuttujien määrittelyyn. Käyttöliittymä toimii vedä ja pudota periaatteella. Tämän avulla voidaan luoda dokumentteja, jotka sisältävät useita raportteja, taulukoita sekä diagrammeja. Java Report Panel on Java applet sovellus. Se toimii selaimen avulla eikä tarvitse erillistä asennusta, mutta tietokoneessa täytyy olla Java-ympäristö asennettuna.

Web Intelligence Rich Client on Windows pohjainen sovellus, joka tarjoaa samat ominaisuudet kuin Java Report Panel. Sovelluksen avulla voidaan työskennellä ilman verkkoyhteyttä. Dokumentteja voidaan luoda ja tallentaa tietokoneen levyille. Dokumenteissa voidaan käyttää lisäksi tietokoneelta löytyviä tiedostoja.

Interactive HTML käyttöliittymä on suunniteltu puhtaasti html ympäristössä toimivaksi. Se tarjoaa suurimman osan ominaisuuksista, jotka edellä mainitut käyttöliittymät tarjoavat. Neljäs käyttöliittymä on Web Accessibility HTML Panel, joka on Interactive HTML käyttöliittymän kaltainen, mutta yksikertaistettu. Se on tarkoitettu kehittäjille, jotka haluavat ottaa käyttöön omien toiveiden mukaisesti muokatun version Web Intelligence ohjelmasta.

Web intelligence -työkalun dokumentointiominaisuuksien avulla voidaan luoda ammattimaisia raportteja. Kun informaatio on haettu tietokannasta, voidaan se näyttää useassa eri muodossa. Vaihtoehtoina ovat erilaiset taulukot (pystysuorat, vaakasuorat, lomake) sekä diagrammit (pylväs, viiva, piirakka). Kuviossa 16 on esitetty muutamia esimerkkejä.



KUVIO 16. Erityyppisiä kaavioita

5 MAPUI KARTTAKÄYTTÖLIITTYMÄ

5.1 Käyttöliittymä

Mapui työ toteutettiin Google Maps karttapalvelun avulla. Google Maps tarjoaa helpon tavan integroida kartta internet sivulle. Lisäksi käyttämällä Google web toolkit työkalua, voidaan tehdä helposti dynaamisia web-sivuja ja Ajax-sovelluksia. Tämä tarkoittaa sitä, että sivua ei tarvitse ladata aina uudestaan, kun sivulla tapahtuu muutoksia tai tietoja haetaan palvelimelta. Google Maps valittiin työhön ilmaisuuden sekä helppojen ja monipuolisten työkalujen takia.

Työssä käytetään seuraavia työkaluja, Eclipse, Google web toolkit ja Google Maps api. Eclipse on ohjelmointiympäristö, jolla ohjelmointi pääasiassa suoritetaan. Google web toolkit on java kehys, jolla kehitään Ajax-sovelluksia. Google Maps api on Google web toolkit laajennos, jolla voidaan integroida Google Maps kartat internetsovelluksiin.

5.2 Työkalut

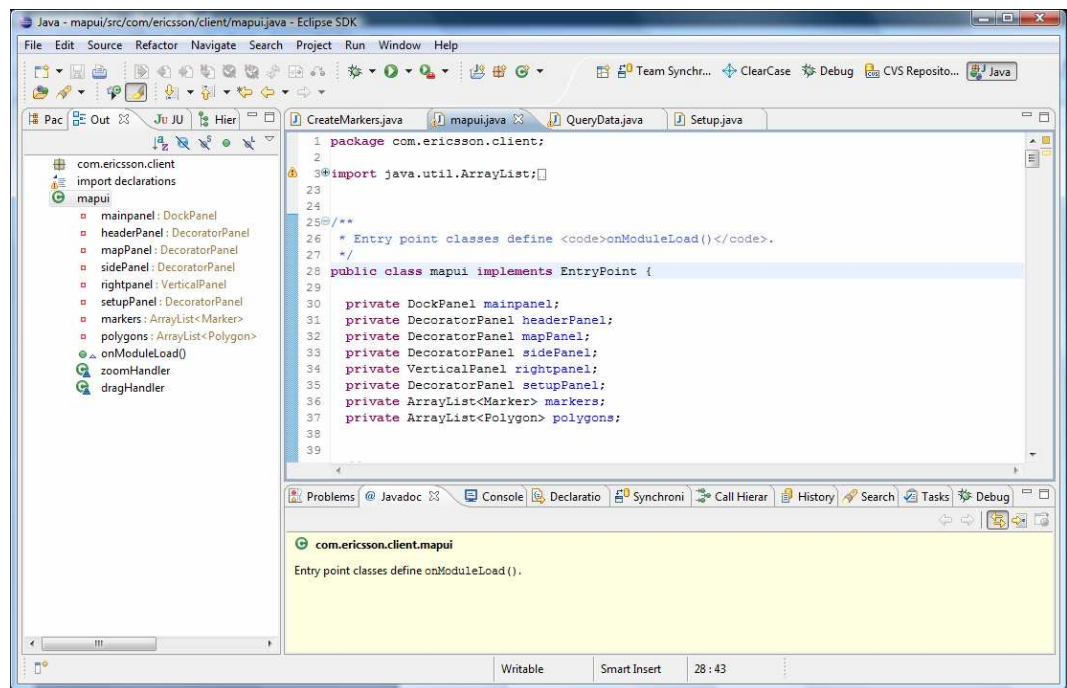
5.2.1 Eclipse

Eclipse on ohjelmointiympäristö, joka on kirjoitettu Javalla (kuvio 17). Alun perin Eclipse oli tarkoitettu Java kielen ohjelmointiin, mutta nykyään sen avulla voidaan kirjoittaa myös muitakin kieliä, kuten C, C++ ja PHP. Eclipsessä on liitännäis (plug-in) tuki. Tämä mahdollistaa monien toiminnallisuuksien integroimisen käyttöliittymään. Liitännäisten avulla voidaan tehdä esimerkiksi UML-mallinnusta.

Ohjelmointiympäristön mukana tulee tuki CVS-versionhallinnalle. Versionhallinta on tekniikka, jonka avulla voi seurata dokumentteihin tehtyjä muutoksia.

Dokumentit voivat olla esimerkiksi ohjelmien lähdekoodia, ohjekirjoja tai CAD-piirustuksia. Mikäli jostakin syystä ei haluta käyttää CVS-versionhallintaa, voi sen sijaan käyttää ilmaista Subversionia tai vaikka kaupallista Clearcasea.

Eclipsen on kehittänyt alun perin IBM, ja se julkaistiin avoimena lähdekoodina vuonna 2001. Vuonna 2004 perustettiin voittoa tavoittelematon Eclipse säätiö. Säätiön tarkoitus on kehittää Eclipseä, ja säätiöön kuuluu myös suomalainen yritys Nokia. Nokia on myös kehittänyt oman ohjelmointiympäristön, joka pohjautuu Eclipseen. Ohjelmointiympäristön nimi on Carbide, joka on tarkoitettu puhelimissa toimivien ohjelmien kehittämiseen. Alun perin carbide oli maksullinen, mutta versiosta 2.0 lähtien se on ollut ilmainen. Eclipsestä on siis olemassa kaupallisiakin versioita. Ehkä tunnetuin näistä on MyEclipse, joka on tarkoitettu JavaEE-sovellusten kehittämiseen.



KUVIO 17. Eclipse kehitysympäristö

5.2.2 Google web toolkit

Google web toolkit (GWT) on Googlen suunnittelema avoimen lähdekoodin Java-kehys. GWT mahdollistaa Ajax-sovellusten kirjoittamisen Javan avulla. Kun sovellus on kirjoitettu, se käännetään kääntäjän avulla JavaScript tiedostoiksi. Tiedostot voidaan haluttaessa salata tai optimoida. GWT on tarkoitettu ainoastaan selaimessa toimivien asiakassovellusten kirjoittamiseen. Serverin puoleisen sovelluksen voi kirjoittaa esimerkiksi PHP:n (PHP: Hypertext Preprocessor) tai Microsoftin kehittämän ASP-tekniikan (Active Server Pages) avulla.

GWT sisältää komentorivityökalun, jonka avulla voidaan luoda suoraan Eclipsen projekti. Tämän jälkeen projekti tuodaan sisään Eclipseen ja aloitetaan sovelluksen kirjoittaminen. On olemassa muitakin avoimen lähdekoodin projekteja, joilla voidaan luoda GWT-projekteja muihin ohjelmistoympäristöihin, kuten NetBeans ja JDeveloper.

GWT-ohjelmia voidaan ajaa kahdella tavalla. Ensimmäinen tapa on Hosted mode, joka tarkoittaa sitä, että koodia ajetaan Javan virtuaalikoneessa tavukoodina. Tätä käytetään tyypillisesti kehitystyössä, kun koodia testataan ja siitä etsitään virheitä. Toinen tapa on Web mode, jota käytetään ohjelman käyttöönnotossa. Tällöin Java koodi on käännetty JavaScript koodiksi, ja ohjelmaa suoritetaan internet selaimessa.

GWT sisältää useita erilaisia käyttöliittymäkomponentteja, esimerkiksi painonappi, tekstikenttä ja valintanappi. Komponenteille voidaan asettaa tapahtumakuuntelijoita, jotka sitten suorittavat haluttuja toimintoja, kun komponenttiin kohdituu jokin tapahtuma, kuten hiiren näppäimen painallus. Kolmannen osapuolen kirjastoilla voidaan käyttöliittymäkomponenttien määrää lisätä.

Muita tärkeimpiä ominaisuuksia ovat mm. JUnit integrointi ja selaimen historiatietojen hallinta. JUnit on testausta varten luotu testauskehys. Kehyksen avulla voidaan luoda testejä, joilla testataan koodin toimivuutta. GWT hoitaa

myös selaimen historiatietojen hallinnan. Kehittäjän ei tarvitse itse huolehtia sovelluksen historialogiikasta, kun selaimella selataan sivuja eteen- tai taaksepäin. Kehittäjän ei tarvitse myöskään huolehtia koodin toimivuudesta eri selaimissa. GWT luo JavaScript koodia, joka toimii samalla tavalla jokaisella tuetulla selaimella. Tuettuja selaimia ovat Firefox (versiot 1.0, 1.5, 2.0, 3.0), Internet Explorer 6 ja 7, Safari 2 ja 3 sekä Opera 9.0. Tieto tuetuista selaimista on vuoden 2008 kesäkuun kohdalta.

5.2.3 Google Maps API

Google Maps API (Application Programming Interface) tarjoaa internetsivuja suunnitteleville helpon tavan integroida Google Maps -kartan omille sivuilleen. API:sta on kaksi versiota. Yleisemmin käytetty versio tarjoaa suorat JavaScript-funktiot, joilla voidaan esittää kartta internetsivulla. Toinen versio on tarkoitettu käytettäväksi GWT:n kanssa. Se tarjoaa Java kirjaston, jonka avulla voidaan Java kielellä kirjoittaa Google Maps kartan sisältäviä sovelluksia. API:n käyttö edellyttää API-avaimen luomista. Avain luodaan jokaiselle domainille eli verkkotunnukselle erikseen.

API:n avulla voidaan siis näyttää kartta internetsivulla, mutta sillä voidaan tehdä paljon enemmänkin. Kartalla voidaan näyttää merkkejä tai kartalle voidaan piirtää viivoja tai kuvioita. Merkit voivat olla joko Google Maps:n omia merkkejä tai niille voidaan määrittää omat ikonit. Merkkeihin on usein liitetty hiiren näppäimen painamisen kuuntelija, joka sitten avaa info-ikkunan, jossa voi olla esimerkiksi tekstiä tai linkkejä.

5.3 Työ

5.3.1 Projektin luonti

Työ aloitettiin hakemalla Google web toolkit Java-kehys sekä sen kanssa käytettävä Google Maps-kirjasto. Pakettien purkamisen jälkeen aloitettiin projektin luominen. ProjectCreator komentorivityökalulla luodaan Eclipsen projekti. Itse sovellus luodaan applicationCreator-työkalulla (kuvio 18).

```
C:\gwt-windows-1.5.3>projectCreator.cmd -eclipse mapui -out mapui -addtoClassPath C:\gwt-maps-1.0.2\gwt-maps.jar_
```

```
C:\gwt-windows-1.5.3>applicationCreator.cmd -eclipse mapui -out mapui -addtoClassPath C:\gwt-maps-1.0.2\gwt-maps.jar -addModule com.google.gwt.maps.GoogleMaps com.ericsson.client.mapui
```

KUVIO 18. Projektin luominen

ProjectCreatorin parametrilla – eclipse luodaan Eclipsen projektissa tarvittavat .classpath sekä .project tiedostot. Parametri – out kertoo, mihin hakemistoon sovelluksen tiedostot luodaan. Projektiin voidaan liittää GWT:n tukemia kirjastoja – addToClassPath parametrin avulla. Tässä työssä tarvittava Google Maps kirjasto liitetään projektiin kertomalla gwt-maps.jar tiedoston sijainti. Kun applicationCreator on ajettu, löytyy mapui hakemistosta kaikki tarvittavat tiedostot. Tämän jälkeen luodut tiedostot voidaan tuoda Eclipsen Java projektiksi import-toiminnon avulla.

Jotta GWT voi käyttää Google Maps kirjastoa, täytyy moduuli xml tiedostoon kirjoittaa rivi ” <inherits name="com.google.gwt.maps.GoogleMaps" />” (kuvio 19 rivi 16). Projektiin lisätään myös HTTP- ja XML-moduulit, jotta voidaan käyttää GWT:n verkko-ominaisuuksia sekä xml-parseria. Tiedostossa määritellään myös css-tyylitiedoston nimi, Google Maps:ssa tarvittava avain, teema sekä Java

luokka, joka toteuttaa EntryPoint rajapinnan. Sovellus käynnistetään luokan toteuttamasta onModuleLoad-metodista.

```

1<module>
2
3  <!-- Inherit the core Web Toolkit stuff.          -->
4  <inherits name='com.google.gwt.user.User' />
5  <inherits name="com.google.gwt.http.HTTP" />
6  <inherits name='com.google.gwt.xml.XML' />
7
8  <!-- Inherit the default GWT style sheet.  You can change  -->
9  <!-- the theme of your GWT application by uncommenting  -->
10 <!-- any one of the following lines.              -->
11 <!-- inherits name='com.google.gwt.user.theme.standard.Standard' /-->
12 <inherits name='com.google.gwt.user.theme.chrome.Chrome' />
13 <!-- inherits name='com.google.gwt.user.theme.dark.Dark' /  -->
14
15 <!-- Other module inherits                        -->
16 <inherits name="com.google.gwt.maps.GoogleMaps" />
17
18
19 <!-- Specify the app entry point class.          -->
20 <entry-point class='com.ericsson.client.mapui' />
21
22 <!-- Specify the application specific style sheet.  -->
23 <stylesheet src='mapui.css' />
24 <script src="http://maps.google.com/maps?gwt=1&file=api&v=2" />
25
26</module>

```

KUVIO 19. mapui.gwt.xml tiedosto

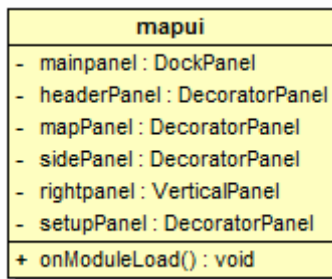
Karttakäyttöliittymä toteutettiin sekä asiakas että palvelinsovelluksena. Itse mapui-sovellusta ajetaan asiakassovelluksena internet selaimella. Käyttöliittymä sovelluksella näytetään tiedot dynaamisesti käyttäjälle. Tietojen haku tapahtuu Ajax-tekniikan avulla palvelimelta. Palvelinsovellus koostuu Java servlet -tiedostosta, joka hoitaa tietokantayhteyden ja informaation haun kannasta.

5.3.2 Käyttöliittymän suunnittelu ja toteutus

Käyttöliittymän pitäisi toimia siten, että kartalta voidaan nähdä tukiasemien sijainti sekä niiden pääsuuntaus. Lisäksi tukiasemista voidaan nähdä muita tarvittavia tietoja liikuttamalla hiiren kursori tukiaseman ikonin päälle. Nämä tiedot näytetään dynaamisesti pienessä ikkunassa kartan vieressä.

Lisäinformaatiota voidaan nähdä myös infoikkunassa, joka avautuu tukiaseman ikonia klikatessa. Käyttöliittymässä on lisäksi asetuspaneeli. Paneelin avulla voidaan piilottaa tai näyttää ikonit kartalla.

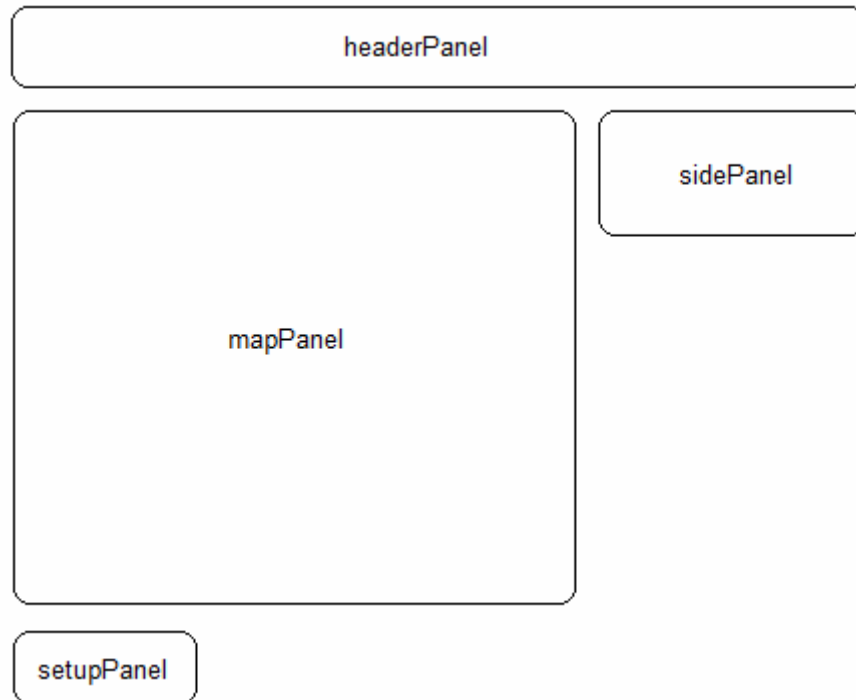
Käyttöliittymän toteutus aloitettiin muokkaamalla mapui luokkaa, jonka applicationCreator työkalu loi automaattisesti. Luokkakaavio on esitetty kuviossa 20. Luokalla on yksi julkinen metodi onModuleLoad, joka suoritetaan ensimmäisenä sovelluksen käynnistyessä.



KUVIO 20. Mapui-luokkakaavio

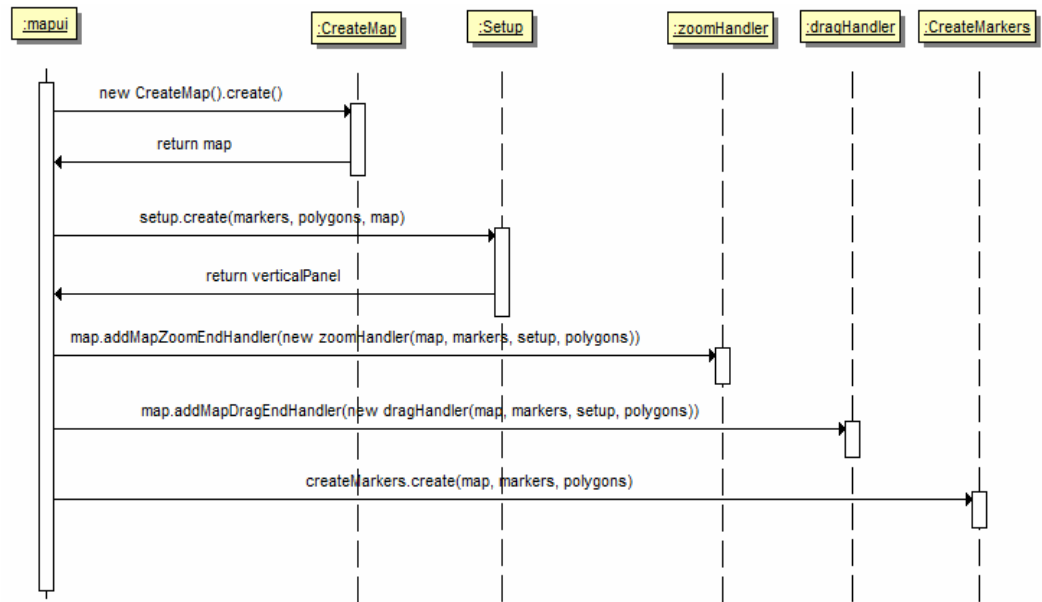
Luokan onModuleLoad metodissa luodaan DockPanel tyyppinen olio mainPanel, johon lisätään kaikki käyttöliittymä komponentit. Käyttöliittymä koostuu neljästä paneelista (kuvio 21). Ylin paneeli on otsikkopaneeli, jonka alapuolella on karttapaneeleja sekä sivupaneeli. Alimpana on asetus paneeli. Paneelit luodaan metodin alussa, ja niiden sisältö luodaan erillisissä luokissa. Paneelit lisätään mainPanel olioon, minkä jälkeen itse mainPanelin sisältö lisätään html sivulle. Tämän jälkeen luodaan merkit, jotka kuvaavat tukiasemien sijaintia sekä suuntausta. Mapui luokka sisältää myös tapahtumankäsittelijä luokat kartta objektille. Tapahtumankäsittelijöiden avulla hoidetaan merkkien sijoitus kartalle. Tukiasemien ja niiden suuntaukset näytetään kartalla vasta, kun kartan suurennustaso on tarpeeksi korkea. Google maps:ssa on 19 suurennustasoa ja ZoomHandler tapahtumankäsittelijässä selvitetään sen hetkinen suurennustaso. Mikäli suurennustaso on suurempi kuin 7, näytetään tukiasemien ikonit kartalla. Kun suurennustaso on suurempi kuin 12, näytetään lisäksi tukiaseman

pääsuuntaus. Tämä tehdään selaimen suorituskyvyn vuoksi. Tukiaseimia on useita tuhansia, ja näitä varten täytyy luoda kaksinkertainen määrä karttamerkkejä, eikä kaikkia voida näyttää kartalla yhtä aikaa.



KUVIO 21. Käyttöliittymän paneelit

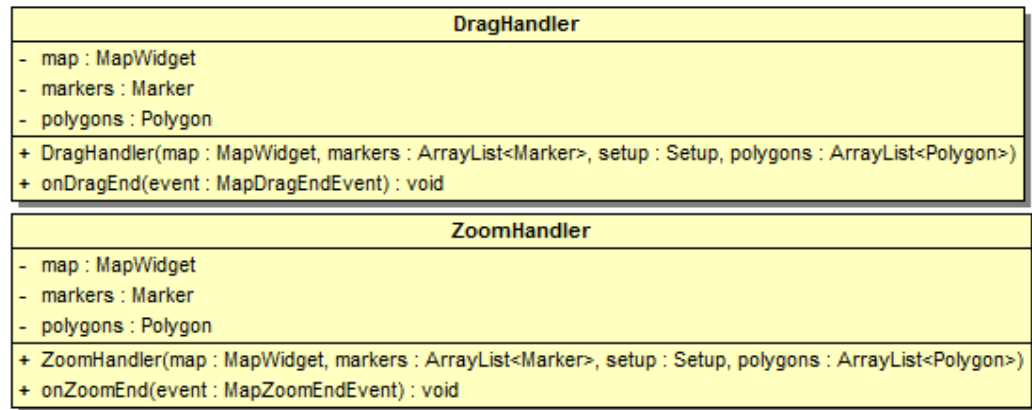
Mapui luokan toimintaa on kuvattu kuviossa 22. Luokan alussa luodaan kartta CreateMap luokan avulla. Luokka palauttaa kartta objektin. Tämän jälkeen luodaan setup paneelin sisältö Setup-luokassa. CreateMap-luokan create-metodin paluttamalle map-objektille lisätään kaksi tapahtumankuuntelijaa. Tapahtumankuuntelijat tekevät toimenpiteitä, kun karttaa on siirretty tai suurennustasoa muutettu. Viimeisenä luodaan merkit kartalle. CreateMarkers luokan create metodille annetaan parametrina karttaobjekti, johon merkit piirretään.



KUVIO 22. Sekvenssikaavio

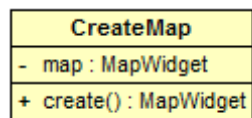
Tapahtumankäsittelijöiden avulla merkit piirretään kartalle käyttäjän toimintojen perusteella. ZoomHandler- sekä DragHandler-luokat sisältävät täsmälleen samat toiminnot. Kartan siirtämisen tai zoomauksen jälkeen kartalta poistetaan kaikki merkit, minkä jälkeen kaikki merkit iteroidaan yksi kerrallaan. Merkin koordinaatteja verrataan näkyvään kartta-alueeseen. Mikäli koordinaatit osuvat näkyvälle kartta-alueelle, merkki lisätään karttaan. Koska kartalle piirretään aina vain näkyvän alueen merkit, säilytetään selaimen suorituskyky.

Luokkakaavio on esitetty kuviossa 23. Luokissa on attribuuttina kartta objekti sekä hajautustaulut, jotka sisältävät kartalla näkyvät objektit. Attribuutit ovat näkyvyysalueeltaan yksityisiä. Julkisia metodeja ovat rakentaja metodit sekä metodit, jotka suoritetaan tapahtuman jälkeen.



KUVIO 23. Tapahtumankäsittelijöiden luokkakaaviot

CreateMap luokan avulla luodaan kartta objekti. Luokan luokkakaavio on esitetty kuviossa 24 ja koodi kuviossa 25. Näkyvä kartta-objekti on GWT-kirjastossa mapWidget-luokka. Create-metodissa luodaan uusi kartta rivillä 17 (kuvio 25). Luonnin jälkeen kartalle määritellään erilaisia astuksia, ja ensimmäisenä määritellään kartan koko. Kartta voidaan keskittää haluttuun koordinaattiin rivin 19 mukaisesti. Lisäksi kartta zoomataan tasolle 2, jolloin kartalla näkyvät kaikki maat. Kartalle lisätään lisäksi kontrollipainikkeet sekä kartan mittakaava. Rivillä 22 mahdollistetaan kartaan zoomaus hiiren rullan avulla. Hiiren rullan käyttö parantaa kartan käytettävyyttä huomattavasti.



KUVIO 24. CreateMap-luokkakaavio

```

11 public class CreateMap {
12
13     private MapWidget map;
14
15     public MapWidget create() {
16
17         map = new MapWidget();
18         map.setSize("1000px", "700px");
19         map.setCenter(LatLng.newInstance(64.509, 26.434), 2);
20         map.addControl(new LargeMapControl());
21         map.addControl(new ScaleControl());
22         map.setScrollWheelZoomEnabled(true);
23
24         return map;
25     }
26 }

```

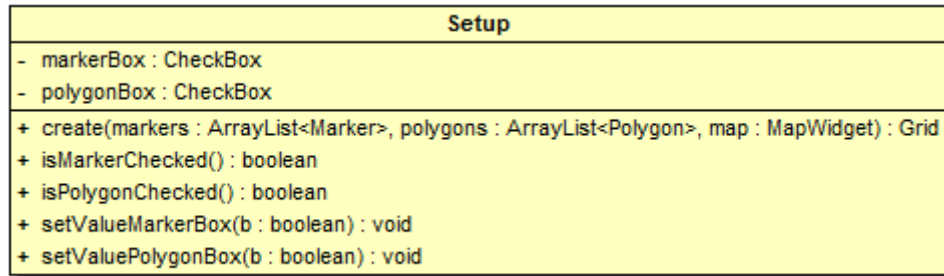
KUVIO 25. CreateMap-koodi

Setup-luokan avulla voidaan kontrolloida kartalla näkyviä objekteja (kuvio 26). Tukiasemat sekä tukiasemien suuntaukset voidaan poistaa näkyviltä, jotta karttaa olisi helpompi lukea. Mapui luokassa luodaan setup-olio, jonka jälkeen kutsutaan olion create-metodia. Metodi luo taulukon, joka sisältää kaikki käytettävät asetukset ja palauttaa sen mapui luokkaan. Mapui luokka sijoittaa palautetun taulukon asetuspaneeliin.

Create-metodissa luodaan kaksi valintaruutua. Toisen valintaruudun avulla voidaan hallita tukiasemien ja toisella tukiasemien suuntausten näkymistä kartalla. Molemmille valintaruuduille luodaan omat tapahtumankäsittelijät. Tapahtumankäsittelijöissä iteroidaan hajautustaulut, jotka sisältävät tukiasemamerkit sekä suuntausmonikulmiot, läpi yksi kerrallaan. Valintaruudun tilasta riippuen tukiasemia kuvaavat merkit tai suuntaukset poistetaan kartalta tai näytetään kartalla.

Valintaruudut sekä niille annetut kuvaukset sijoitetaan taulukkoon, joka palautetaan metodin lopussa. Taulukkoon lisätään myös kysymysmerkki rivin loppuun, jota klikkaamalla voidaan avata ohjeita sisältävä ponnahdusikkuna.

Setup-luokassa on lisäksi kaksi metodia: `isMarkerChecked` sekä `isPolygonChecked`. Metodien avulla voidaan muualla projektissa kysyä valintaruutujen tiloja.



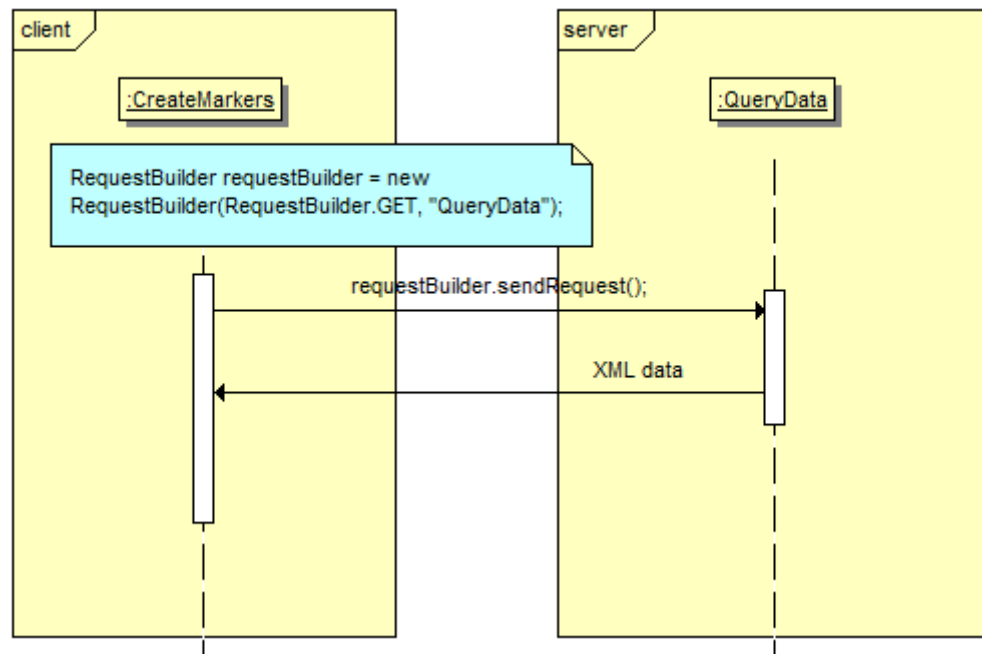
KUVIO 26. Setup luokkakaavio

CreateMarkers-luokassa luodaan tukiasema merkit ja niiden suuntauksia kuvaavat monikulmiot. Create metodissa luodaan RequestBuilder olio, jonka avulla otetaan yhteys serverillä sijaitsevaan servlet:iin (kuvio 27). RequestBuilder luokka käyttää Ajax-tekniikkaa lähettäessään HTTP-pyyntöön serverille. Luokan muodostin metodissa määritellään käytettävä http metodi, get tai post sekä url-osoite, johon pyyntö lähetetään. Tässä työssä lähetetään http get pyyntö QueryData servletille. Servletin avulla otetaan yhteys Sybase-tietokantaan. Tietokannasta haetaan tukiasemista tarvittavat informaatiot. Haun jälkeen informaatio lähetetään mapui sovelluksen CreateMarkers olioon XML-muodossa.

Kartalla olevat merkit luodaan XML-datan perusteella. Jokaiselle merkille on omat marker elementtinsä XML-muodossa. Marker-elementti sisältää muita elementtejä, jotka sisältävät tukiaseman informaation. Merkit luodaan siten, että ensin selvitetään marker-elementtien lukumäärä. Tämän jälkeen silmukassa parsitaan jokaisen marker-elementin alielementtien informaatio. Tämän informaation perusteella luodaan uusi merkki ja laitetaan se hashmap tietorakenteeseen. Lisäksi merkille luodaan kolme tapahtumankäsittelijää, hiiren klikkauksen kuuntelija sekä hiiren päälle- ja poisvienti kuuntelijat. Kun merkkiä klikkaa hiiren vasemmalla painikkeella, karttaan piirretään viiva toiseen

tukiasemaan. Tällä kuvataan tukiaseman naapurisolua. Kun hiiren kursori viedään tukiaseman merkin päälle, sivupaneelissa näytetään tukiasemasta kertovaa informaatiota. Sivupaneeli poistetaan näkyviltä siinä vaiheessa, kun kursori poistuu merkin päältä.

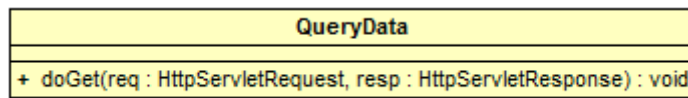
Tukiaseman merkin lisäksi luodaan uusi monikulmio kuvaamaan tukiaseman suuntausta. Monikulmio luodaan erillisessä metodissa, jolle annetaan parametriksi tukiaseman koordinaatit, kulma sekä suuntaus. Monikulmio muodostetaan vähintään kolmen pisteen avulla. Pisteiksi annetaan tukiaseman koordinaatit sekä laskemalla kulman avulla kaksi pistettä halutun säteen verran tukiaseman koordinaateista. Pisteiden määrä riippuu kulmasta. Pisteiden määrä on minimoitu, jotta selaimen suorituskyky ei heikkene pisteiden laskemisen johdosta. Kun pisteet on laskettu, luodaan uusi monikulmio ja laitetaan se omaan hashmap-tietorakenteeseensa. Jotta toisiinsa liittyvät monikulmiot ja merkit voidaan helposti löytää tietorakenteista, annetaan niille samat avaimet tietorakenteessa.



KUVIO 27. Asiakaspalvelinkommunikointi

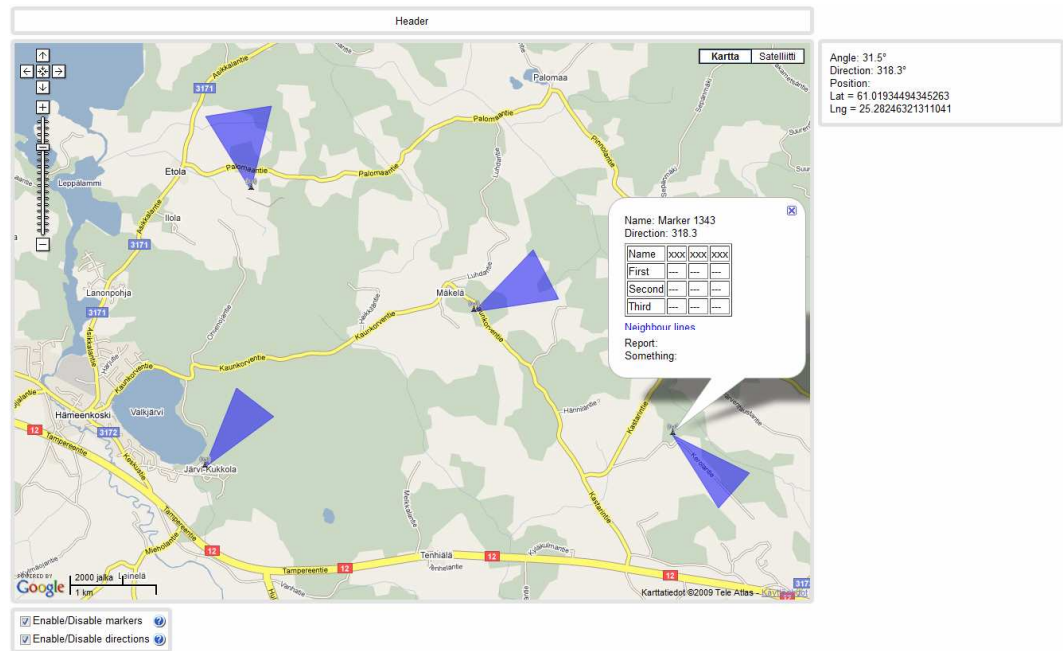
Tiedon prosessointi palvelimelta hoidetaan servletin avulla. Servlet on dynaamisten internetsivujen kehittämiseen tarkoitettu rajapinta. Servletille välitetään http-pyyntö, ja pyyntöön voidaan vastata esimerkiksi doGet-metodissa.

QueryData servlet hakee tiedot tietokannasta. Luokkakaavio on esitetty kuviossa 28. Kun tietokannasta on haettu tukiasemia koskeva informaatio, muodostetaan niistä xml-muotoinen dokumentti, joka palautetaan mapui-sovellukseen. Dokumentti sisältää juuri-elementin sekä marker-elementit ja niiden alielementit. Marker elementti koostuu nimi-, koordinaatti-, suunta- sekä naapurielementeistä. Näille annetaan attribuuttina arvoja, jotka sitten CreateMarker-luokassa parsitaan. Koordinaatti elementin attribuuttina ovat pituus- ja leveysasteet (lat, lng). Näiden arvoina on merkin koordinaatit. Suuntaus elementin attribuutteina ovat kulma sekä suunta. Näillä kuvataan tukiaseman pääsuuntaus sekä suuntauksen kulma. Naapuri-elementillä kuvataan tukiaseman naapuri solu.



KUVIO 28. QueryData servlet

Kuviossa 29 on kuvankaappaus mapui sovelluksesta. Kuviossa nähdään neljä tukiasemaa sekä niiden pääsuuntaukset. Lisäksi yhdestä tukiasemasta nähdään lisäinformaatiota oikealla olevassa sivupaneelissa sekä infoikkunassa.



KUVIO 29. MapUI karttakäyttöliittymä

6 YHTEENVETO

Tämän työn tarkoituksena oli tehdä interaktiivinen karttakäyttöliittymä. Käyttöliittymän kautta voidaan katsella tietoja tukiasemista ja luoda raportteja. Työ rajattiin aluksi näyttämään tukiasemista pääsuuntaus ja hieman lisäinformaatiota, kun tukiaseman ikonia klikataan. Myöhemmässä vaiheessa karttakäyttöliittymään voidaan kuitenkin lisätä uusia toimintoja. Tukiasemista saadaan kuitenkin niin paljon informaatiota, että kaiken näyttäminen vaatisi paljon työtä ja aikaa.

Karttasovellusten tekemiseen on useita eri työkaluja. Tässä työssä vertailtiin kahta ilmaista sekä yhtä maksullista karttapalvelua: Google Mapsia, Live search mapia sekä MapXtremeä. Kaksi ensimmäistä on internetiselaimessa toimivia Javascript-karttapalveluja. MapXtreme on joko Java- tai .NET-ohjelmointikielellä kirjoitettava karttapalvelu. Tähän työhön valittiin Google Maps sen ilmaisuuden sekä helppojen työkalujen perusteella.

Työn toteutuksessa käytettiin Eclipse-ohjelmointiympäristöä sekä Google web - toolkit työkalua. GWT-työkalun avulla voidaan luoda dynaamisia internetsovelluksia Java-kielellä. Siihen voidaan integroida Google Maps-kirjasto, jolloin sovellukseen voidaan lisätä myös karttapalvelu.

Työ toteutettiin asiakas ja palvelin sovelluksena. Mapui-sovellusta ajetaan internetiselaimella, ja tiedot tukiasemista näytetään käyttäjälle dynaamisesti. Palvelinsovellus hakee tiedot tietokannasta ja lähettää ne asiakassovellukselle xml-muodossa. Tiedon haku tapahtuu Ajax-tekniikan avulla.

Työn lopputulos on onnistunut. Karttakäyttöliittymään saatiin työn rajauksessa määritetyt toiminnot. Toimintoja on kuitenkin vielä niin vähän, että työ ei ole todellisessa käytössä. Jatkokehityksenä käyttöliittymään voidaan lisätä uusia toimintoja, kuten tukiaseman käyttöastetta kuvaavat värikoodit esimerkiksi

suuntausta näyttävään monikulmioon. Lisäksi voidaan tutkia, miten käyttöliittymää voidaan optimoida tehokkaammin toimivaksi. Esimerkiksi muuttamalla serveriltä lähetettävä XML-muodossa oleva informaatio JSON-muodossa. JSON saattaa vähentää siirrettävän informaation määrää, jolloin sovellus toimii nopeammin.

LÄHTEET

Penttinen, J. 2001. GSM-tekniikka. Helsinki: WSOY.

Wikipedia. 2009. GSM, [viitattu 12.02.2009]. Saatavissa:
<http://fi.wikipedia.org/wiki/Gsm>.

Wikipedia. 2009. Google Maps, [viitattu 15.02.2009]. Saatavissa:
http://en.wikipedia.org/wiki/Google_Maps.

Wikipedia. 2009. Live Search Maps, [viitattu 16.02.2009]. Saatavissa:
http://en.wikipedia.org/wiki/Live_Search_Maps.

Wikipedia. 2009. Koordinaattijärjestelmä, [viitattu 20.03.2009]. Saatavissa:
<http://fi.wikipedia.org/wiki/Koordinaattij%C3%A4rjestelm%C3%A4>.

MapInfo. 2009. MapXtreme Java Edition, developer guide, [viitattu 10.03.2009].
Saatavissa:
http://reference.mapinfo.com/software/mapxtreme_java/english/4_8_2/index.html.

Sybase. 2009. Sybase IQ tuotekuvaus, [viitattu 19.3.2009]. Saatavissa:
<http://www.sybase.fi/products/databasemanagement>.