UNIVERZA V LJUBLJANI

FAKULTETA ZA RAČUNALNIŠTVO IN INFORMATIKO

Boštjan Kastelic

Implementacija GIS z Oraclovimi tehnologijami

DIPLOMSKO DELO

NA VISOKOŠOLSKEM STROKOVNEM ŠTUDIJU

Mentor: doc. dr. Janez Demšar

Ljubljana, 2010

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00456/2009
Datum: 01.09.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **BOŠTJAN KASTELIC**

Naslov: **IMPLEMENTACIJA GIS Z ORACLOVIMI TEHNOLOGIJAMI**
IMPLEMENTATION OF A GIS BASED ON ORACLE TECHNOLOGY

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Geografski informacijski sistemi (GIS) so ena od pomembnih rab modernih računalniških tehnologij v praksi. Predstavite, kako so videti takšni sistemi v splošnem in kakšno tehnologijo v ta namen ponuja podjetje Oracle. Uporabo tehnologije preskusite na praktičnem primeru nadgradnje GISa za prikaz grafične enota rabe kmetijskih zemljišč (GERK).

Mentor:

doc. dr. Janez Demšar

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Franc Solina

Univerza
v Ljubljani

Fakulteta za računalništvo
in informatiko

Tržaška 25
1000 Ljubljana, Slovenija
telefon: 01 476 84 11
faks: 01 426 46 47
www.fri.uni-lj.si
e-mail: dekanat@fri.uni-lj.si

Št. naloge: 00456/2009
Datum: 01.09.2009

Univerza v Ljubljani, Fakulteta za računalništvo in informatiko izdaja naslednjo nalogo:

Kandidat: **BOŠTJAN KASTELIC**

Naslov: **IMPLEMENTACIJA GIS Z ORACLOVIMI TEHNOLOGIJAMI**
IMPLEMENTATION OF A GIS BASED ON ORACLE TECHNOLOGY

Vrsta naloge: Diplomsko delo visokošolskega strokovnega študija

Tematika naloge:

Geografski informacijski sistemi (GIS) so ena od pomembnih rab modernih računalniških tehnologij v praksi. Predstavite, kako so videti takšni sistemi v splošnem in kakšno tehnologijo v ta namen ponuja podjetje Oracle. Uporabo tehnologije preskusite na praktičnem primeru nadgradnje GISa za prikaz grafične enota rabe kmetijskih zemljišč (GERK).

Mentor:

doc. dr. Janez Demšar

Fakulteta za računalništvo in informatiko • UNIVERZA V LJUBLJANI • 2

Dekan:

prof. dr. Franc Solina

I Z J A V A O A V T O R S T V U

diplomskega dela

Spodaj podpisani Boštjan Kastelic,

z vpisno številko 63050419,

sem avtor diplomskega dela z naslovom:

Implementacija GIS z Oraclovimi tehnologijami

S svojim podpisom zagotavljam, da:

- sem diplomsko delo izdelal samostojno pod mentorstvom doc. dr. Janeza Demšarja;
- so elektronska oblika diplomskega dela, naslov (slov., angl.), povzetek (slov., angl.) ter ključne besede (slov., angl.) identični s tiskano obliko diplomskega dela;
- soglašam z javno objavo elektronske oblike diplomskega dela v zbirki »Dela FRI«.

V Ljubljani, dne 4.3.2010 Podpis avtorja:

Zahvala

Ob zaključku študija se iskreno zahvaljujem podjetju Ixtlan Team, kjer sem pridobil potrebno znanje. Hvala predvsem mentorju v podjetju Jocu Mlakarju.

Za strokovno vodenje in pomoč pri nastajanju diplomskega dela se zahvaljujem mentorju doc. dr. Janezu Demšarju ter profesorjem na Fakulteti za računalništvo in informatiko.

Za vse vzpodbude, pomoč in podporo se zahvaljujem mojim staršem, sestri, partnerici, prijateljem in vsem, ki so mi pomagali na poti do uspeha.

Kazalo

Seznam uporabljenih kratic in pojmov

ADF(Application Development Framework)
Oraclova tehnologija za razvoj aplikacij.

AKTRP ***(Agencija Republike Slovenije za kmetijske trge in razvoj podeželja)***
Agencija, za katero razvijamo GIS.

API (application programming interface)
Pojem predstavlja aplikacijski vmesnik.

CLOB (character large object)
Polje na bazi, namenjeno shranjevanju velikih podatkov.

EAR (Enterprise Archive)
Javina arhivska datoteka, namenjena pakiranju enega ali več modulov.

ETRS89 (European Terrestrial Reference System 1989)
Evropski kopenski referenčni koordinatni sistem, ki ga od leta 2008 uporablja tudi Slovenija.

FOI (Feature of Interest)
FOI objekti so objekti, s katerimi upravlja strežnik FOI. Aplikacijo MapViewer nadgradijo z dinamiko.

GERK (grafična enota rabe kmetijskih gospodarstev)
Urejenost GERK-ov z vidika posamezne kmetije pomeni, da ima nosilec kmetijskega gospodarstva vpisane oz. pravilno vrisane GERK-e za vse površine v uporabi in pravilno zajete vse podatke, kar je predpogoj, da lahko nosilec vloži vlogo za pridobitev subvencij s področja kmetijstva.

GIF (Graphics Interchange Format)
Rastrski slikovni format, namenjen enostavnejšim motivom in animacijam.

J2EE vsebovalnik (Java 2 Platform Enterprise Edition Container)
Vmesnik med komponento in spodnjim nivojem funkcionalnosti platforme, ki podpirajo komponento. Na vsebovalnik J2EE nameščamo komponente aplikacij J2EE.

JAR (Java ARchive)
Javina arhivska datoteka namenjena arhiviranju distribucij javinih aplikacij ali knjižnic.

JDBC (Java Database Connectivity)
Vmesnik, ki se povezuje na bazo in omogoča izvajanje vprašanj SQL.

JPEG (Joint Photographic Experts Group)
Rastrski slikovni format, primeren tudi prikazovanju fotografij.

JSP Tag Library (Java Server Pages)
Elementi, ki lahko kreirajo, izvajajo in dostopajo do objektov programskega jezika.

Kompresija, dekompresija (compression, decompression)
Stiskanje podatkov, razširjanje podatkov.

MBR (minimum bounding rectangle)
Minimalen pravokotnik, ki zaobjame celotno geometrijo.

Navigacija (navigation)
Sprehajanje po zemljevidu.

Oracle shema (Oracle schema)
Baza Oracle je organizirana po shemah. Shema se kreira skupaj z uporabnikom. Sestavljajo jo bazni objekti (tabele, pogledi, indeksi, procedure, sinonimi ...).

PL/SQL (Procedural Language/Structured Query Language)
Oraclova proceduralna razširitev SQL-a.

PNG (Portable Network Graphics)
Novi rastrski slikovni format, ki nadomešča predvsem pomanjkljivosti formata GIF (omejitev 256 barv ...).

Renderiranje (rendering)
Izrisovanje visokonivojskega opisa v grafično obliko.

Repozitorij (repository)
Okolje, kjer je po določenih kriterijih zbrano, urejeno in shranjeno elektronsko gradivo, npr. del načrta informacijskega sistema, učna gradiva.

STE (Subv Topology Editor)
Starejša aplikacija GIS, ki jo AKTRP uporablja za prikaz in manipulacijo s prostorskimi podatki.

WFS (Web Feature Service)
Je standard, ki omogoča direktno manipulacijo z objekti prostorskih podatkov na strežniku.

WMS (Web Map Service)
Je standard za prenos, poizvedbo in prikaz prostorskih podatkov med strežnikom in klientom ter strežnikom in strežnikom. WMS podpirajo MapServer, ArcIMS in drugi programski paketi.

XML parser (Extensible Markup Language parser)
Razčlenjevalnik kode XML.

Zoom
Velikost prikaza.

Povzetek

Cilj diplomskega dela je vzpostaviti geografski informacijski sistem (GIS), kjer bo možno prikazati že obstoječe geografske podatke. Agencija Republike Slovenije za kmetijske trge in razvoj podeželja (AKTRP) uporablja starejšo aplikacijo, ki temelji na Javini platformi. Zaradi nekaterih potreb po nadgradnji GIS-a, ki jih je skoraj nemogoče uvesti v obstoječi sistem, smo se odločili za razvoj novega z Oraclovim produktom, imenovanim Oracle MapViewer. Navdušila nas je predvsem hitrost delovanja predstavitvenih primerov, ki se nahajajo na uradni strani produkta, in številna orodja, s katerimi si pri gradnji GIS-a lahko pomagamo. Odločitvenega pomena so bili tudi geografski podatki, ki so že shranjeni v obliki Oracle Spatial, ter možnost namestitve produkta Oracle MapViewer na obstoječi strežnik Oracle Application Server.

Novi GIS, poimenovan AgriMap, je zadovoljil vse naše zahteve. Meritev nam ni bilo potrebno izvajati, saj je bila performanca v nasprotju z obstoječim sistemom, že na prvi pogled boljša pri AgriMapu. Spoznali smo tudi, da so Oraclove tehnologije, kot sta na primer Oracle MapViewer in Oracle Maps, dovolj zmogljive za implementacijo funkcionalnosti obstoječega GIS-a in za uresničitev ter razvoj vseh novih zahtev.

Ključne besede:

GIS, AgriMap, Oracle Spatial, Oracle MapViewer, Oracle Maps

Abstract

The purpose of this diploma thesis is to establish geographic information system (GIS), which would enable the presentation of already existing geographic data. Agency of the Republic of Slovenia for Agricultural Markets and Rural Development uses older application, based on Java platform. Due to the growing needs for the improvement of the GIS, which are almost impossible to introduce into the existing one, we decided for the development of a new system with Oracle product called Oracle MapViewer. We were enthusiastic about the speed of the performance examples, located on the official site of the product, and numerous tools, which can help us with building the GIS. The geographic data, saved in the form of Oracle Spatial, and the possibility to place Oracle Map Viewer on the existing Oracle Application Server were the most significant factors for our decision.

The new GIS, called AgriMap, met all our requirements. We did not have to carry out the measurements, since the performance with AgriMap looked better already at first sight. We found out that Oracle technologies, for example Oracle MapViewer and Oracle Maps, are efficient enough for the implementation of the existent GIS and for the realization and progress of the emerging requirements.

Key words:

GIS, AgriMap, Oracle MapViewer, Oracle Spatial, Oracle Maps

1 Uvod

1.1 Tema diplomskega dela

Geografski informacijski sistem (v nadaljevanju GIS) je izraz, ki vključuje celotno strojno, programsko in podatkovno opremo, ki nam omogočajo manipulacijo, analiziranje in prikazovanje geografskih podatkov. (1)

Ixtlan Team je podjetje, kjer smo razvijali diplomsko delo. Ukvarja se z načrtovanjem informacijskih sistemov in razvojem programske opreme po naročilu. Naročniki sistemov so predvsem ustanove državne in javne uprave (ministrstva, agencije ...). Zelo pomemben naročnik je Agencija Republike Slovenije za kmetijske trge in razvoj podeželja (v nadaljevanju AKTRP) za katerega razvijamo aplikacije za zajem vlog za subvencije in grafično aplikacijo Subv Toplogy Editor (v nadaljevanju STE).

STE temelji na javini platformi. Aplikacija omogoča prikaz geografskih podatkov določenega kmetijskega gospodarstva ali prikaz grafične enote rabe kmetijskih gospodarstev (v nadaljevanju GERK). Na spletni aplikaciji za zajem vlog za subvencije izberemo določen GERK, ki odpre namizno grafično aplikacijo STE (Slika 19), ki izriše vse njegove podatke (poljine, rabe in drugo).

Na AKTRP-ju so podali zahtevo, da bi lahko preko zemljevida pridobivali podatke o GERK-ih iz sosednjih kmetijskih gospodarstev. Radi bi omogočili tako imenovani sprehod po zemljevidu (navigacija), kjer bi lahko uporabnik s klikom nanj pridobival podatke za izbrano območje. Ker bi bila posodobitev obstoječega sistema prezahtevna in nesmiselna predvsem s performančnega vidika, smo se odločili, da razvijemo nov sistem, ki bo deloval podobno kot najsodobnejši zemljevidi, na primer Google Maps. Za najprimernejše orodje pri izdelavi takega sistema smo izbrali Oracle MapViewer.

Cilj diplomske naloge je vzpostaviti hiter in zmogljiv GIS, ki bo prikazoval geografske podatke v obliki zemljevida. Zemljevid mora podpirati navigacijo, približevanje, iskanje ter izpis podatkov ob kliku na določeno območje.

1.2 Opis problemskega področja

Obstoječa aplikacija deluje precej počasi, saj je potreben njen zagon preko javinega navideznega stroja (angl. Java Virtual Machine – JVM). Povezuje se še na strežnik ArcIMS, kjer pridobi letalsko sliko, na katero nariše poligone, ki jih mora pred tem še normalizirati. Aplikacija zna prikazati GERK-e samo za določeno kmetijo.

Hiter razvoj internetnih tehnologij je pripomogel tudi k razvoju GIS-ov. Spletni GIS-i imajo mnoge prednosti. Omogočajo hitrejši razvoj in lažjo integracijo kot samostojne aplikacije. Veliko prednost pa občutijo tudi uporabniki, saj jim ni potrebno nameščati odjemalske programske opreme. V vlogi odjemalca nastopajo kar brskalniki, ki pa jih uporabljamo praktično vsi.

2 GIS

GIS vključuje celotno strojno, programsko in podatkovno opremo. Omogoča nam manipulacijo, analiziranje in prikazovanje geografskih podatkov. Poimenovali bi ga lahko kot »pameten zemljevid«, kjer lahko uporabniki iz ogromnega števila podatkov lažje ter učinkoviteje pridobijo informacije. (1)

2.1 Zgodovina razvoja GIS

Nastanek GIS-a sega že v prazgodovino. Pred 15.500 leti so lovci v jamah risali poti živali, s čimer so postali učinkovitejši pri lovu.

Leta 1854 je v Londonu izbruhnila kolera. John Snow je narisal točke posameznih primerov, s katerih je kasneje lahko razbral vir bolezni (vodnjak Broad Street). Vodnjak so prenehali uporabljati in izbruh se je končal.

Leta 1962 je nastal prvi pravi računalniški GIS imenovan CGIS (kanadski geografski sistem). Razvil ga je Roger Tomlinson, uporabljal pa se je za shranjevanje, analiziranje in manipuliranje s parcelnimi podatki.

Leta 1989 sta nastala prva komercialna produkta ESRI (angl. Environmental Systems Research Institute) in CARIS (angl. Computer Aided Resource Information System). V tem letu je bil utirjen tudi prvi od 24-ih satelitov, ki jih trenutno uporablja sistem globalnega določanja položaja (angl. Global Positioning System – GPS).

V poznih 1990-ih je z razvojem omrežij narasla potreba po splošni dostopnosti do GIS-ov. Nastali so standardi, ki omogočajo dostop do njih preko brskalnikov (Google Maps, Bing Maps …). (2)

2.2 Današnji GIS-i

Najsodobnejši GIS-i omogočajo številne operacije, ki še dodatno obogatijo uporabniško izkušnjo. Obnašajo se kot nekakšni portali, kjer lahko uporabniki določajo:

- načine prikaza podatkov,
- sloje,
- barve,
- velikost,
- velikost prikaza in drugo.

Pomemben mejnik pri razvoju GIS-ov predstavljajo globalni navigacijski satelitski sistemi (v nadaljevanju GNSS). GNSS je skupno ime za sisteme, ki s pomočjo satelitov določajo koordinatne točke kjerkoli na Zemlji. Poznamo več takšnih sistemov, najbolj znani med njimi so:

- ameriški GPS,
- ruski GLONASS,
- evropski Galileo in
- kitajski Beidou. (3)

GIS iz GNSS pridobiva koordinatne, ki jih nato pretvori v svoj koordinatni sistem in jih uporabi. Na ta način delujejo praktično vsi cestni, morski, zračni in drugi navigacijski sistemi. Ti sistemi znajo na zemljevidu določati in prikazovati:

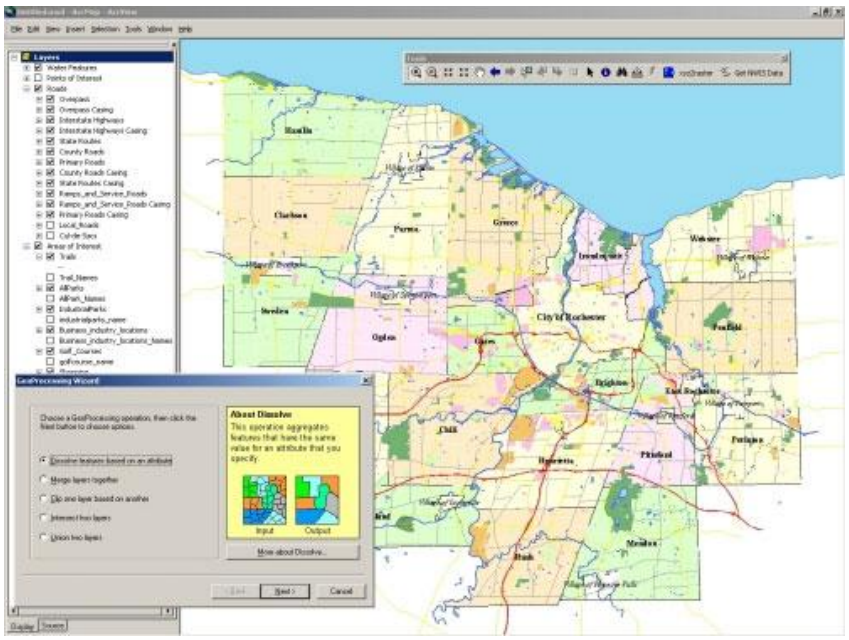
- lokacijo,
- optimalno pot,
- podatke o dogodkih, restavracijah, bolnišnicah v bližini,
- hitrosti vožnje,
- razdalje,
- sled …

Vsi najsodobnejši GIS-i so sestavljeni v trinivojski arhitekturi. V današnjem času se pojavljajo predvsem:

- namizni,
- spletni in
- mobilni GIS-i.

2.2.1 Namizni GIS

GIS-i, ki so razviti kot ločene aplikacije, izgubljajo na pomenu. Ne samo, da jih je težavno integrirati v uporabniško okolje, postali so tudi prezahtevni za razvoj in izboljšave. Primer namiznih GIS-ov prikazuje Slika 1.



Slika 1: Primer namizne aplikacije GIS (4)

2.2.2 Spletni GIS

Spletni sistemi so dostopni preko brskalnikov, zato uporabnikom ni potrebno nameščati nobenih odjemalskih programov. Pri spletnih GIS-ih v vlogi odjemalcev nastopajo brskalniki. Ti so zaradi prihoda novih tehnologij (na primer Ajax) postali dovolj razviti za prikaz in interakcijo z zemljevidi. Najbolj znani med njimi so Google Maps, Garmin, Navteq, Microsoft Bing Maps … Primer spletnih GIS-ov prikazuje Slika 2.

Slika 2: Primer spletne aplikacije GIS (5)

2.2.3 Mobilni GIS

Zaradi razširjenosti hitrega mobilnega omrežja in možnostjo uporabe signalov GPS, je uporaba mobilnih GIS aplikacij močno pridobila na veljavi. Le še vprašanje časa je, kdaj bodo mobilniki dovolj sposobni, da bodo, poleg spletnih, lahko poganjali tudi namizne sisteme. Primer mobilnih GIS-ov prikazuje Slika 3.

Slika 3: Primer mobilne aplikacije GIS (6)

3 Oraclove tehnologije za GIS

3.1 Oracle Spatial

Oracle Spatial je shema z imenom MDSYS in vgrajena množica funkcij ter procedur, ki omogočajo shranjevanje, dostop in analiziranje prostorskih podatkov v bazi Oracle. Oracle Spatial je dostopen z bazo Oracle Database 11g Enterprise Edition, medtem ko je njegova okrnjena različica, imenovana Oracle Locator, na voljo že v brezplačni verziji Oracle Database XE (Express Edition).

Podatkovni model je hierarhičen, sestavljen iz elementov, ki pripadajo geometriji. Vsaka geometrija pa sodi k določenemu sloju.

3.1.1 Element

Element je osnovni gradnik geometrije. Opisujejo ga koordinate (X, Y, Z). Element je lahko točka, niz črt ali poligon.

Točka je določena z dvema ali tremi koordinatami, medtem ko jih lahko črte in poligoni vsebujejo več. Poligon mora biti v primerjavi z nizom črt pravilno zaključen, kar pomeni, da mora imeti obliko zaprtega območja. Podatki oziroma koordinate, ki so shranjeni kot poligoni, imajo najpogosteje začetno točko enako končni. Obstajajo še drugi tipi (krog, pravokotnik, krožni poligoni), ki pa so le izpeljanke osnovnih tipov. Oracle Spatial podpira tudi 3- in 4-dimenzionalne elemente.

3.1.2 Geometrija

Geometrijo predstavlja element ali pa množica elementov. Geometrija lahko vsebuje tudi elemente z različnimi tipi (območje s cestami, poligone z luknjami ...).

3.1.3 Sloj

Sloj je niz geometrij, ki skupaj opisuje neko skupino (ceste, mesta, dejavnosti, omrežja ...).
Razdelitev na sloje nam olajša razvoj in obogati GIS, saj so sloji, ki vsebinsko niso povezani, med seboj lahko ločeni.

3.1.4 Podatkovni tip Spatial in metapodatki

Geometrijo in njen indeks lahko definiramo z jezikom za definiranje podatkov (angl. Data Definition Language – DDL), možna pa je tudi manipulacija z jezikom za ravnanje s podatki (angl. Data Manipulation Language – DML). Tipično so geografski podatki v Oracle bazi shranjeni v tabelah kot SDO_GEOMETRY tip.

```
SQL> SELECT * FROM MVDEMO.ORIENTED_POINTS;

   ID NAME                 SHAPE
   ---------------------- --------
   1  Ljubljana           <Object>
   2  Maribor             <Object>
   3  Novo mesto          <Object>
   4  Celje               <Object>
```

Tip SDO_GEOMETRY je definiran na sledeči način:

```
CREATE TYPE sdo_geometry AS OBJECT (
    SDO_GTYPE NUMBER,
    SDO_SRID NUMBER,
    SDO_POINT SDO_POINT_TYPE,
    SDO_ELEM_INFO SDO_ELEM_INFO_ARRAY,
    SDO_ORDINATES SDO_ORDINATE_ARRAY);
```

Polje SDO_GTYPE določa tip geometrije. Je štirimestno število s formatom DLTT. D določa število dimenzij. Oracle Spatial podpira 2, 3 in 4 dimenzije. V kolikor je geometrija predstavljena v obliki LRS (angl. Linear Referencing System), je potrebno na mesto L vpisati dimenzijo, ki določa merilo. Če geometrija ni predstavljena v obliki LRS ali pa želimo uporabiti privzeto Spatial vrednost (zadnja dimenzija), vpišemo 0. TT pove, katerega tipa je geometrija (00 - nedefiniran, 01 - točka, 02 - niz črt, 03 - poligon, 04 - seznam, 05 - več točk, 06 - več nizov črt, 07 - več poligonov, 08 - 3D telo, 09 - več 3D teles). Vsi elementi geometrije na določenem sloju morajo imeti enako število dimenzij. Kombinacija dvodimenzionalne geometrije s tridimenzionalno ni možna.

Polje SDO_SRID določa koordinatni sistem, ki ga uporablja geometrija. Polje je lahko prazno, kar pomeni, da geometrija nima določenega koordinatnega sistema. V kolikor pa polje vsebuje vrednost, se mora le-ta ujemati z zapisoma v tabeli SDO_COORD_REF_SYS in pogledu USER_SDO_GEOM_METADATA. Vse geometrije morajo pripadati istemu SDO_SRID-u, v nasprotnem primeru ni možno kreirati prostorskega indeksa (angl. Spatial Index). Določitev koordinatnega sistema je zelo pomembna. Če pripadajo koordinate sloja drugemu koordinatnemu sistemu, zna Oracle Spatial narediti transformacijo med koordinatnimi sistemi.

Polje SDO_POINT se uporablja, kadar geometrija nima navedenih vrednost v poljih SDO_ELEM_INFO in SDO_ORDINATES. Polje določa točko z osema X, Y in za tridimenzionalne koordinatne sisteme še z osjo Z.

```
CREATE TYPE sdo_point_type AS OBJECT (X NUMBER,
                                      Y NUMBER,
                                      Z NUMBER);
```

Polje SDO_ELEM_INFO je definirano kot seznam številk, ki določajo pravila in način prikaza geometrij v polju SDO_ORDINATES.

```
CREATE TYPE sdo_elem_info_array AS VARRAY (1048576) OF NUMBER;
```

Polju za vsak element posebej podamo:

- SDO_STARTING_OFFSET,
- SDO_ETYPE,
- SDO_INTERPRETATION.

SDO_STARTING_OFFSET ponazarja odmik od začetne točke geometrije v tabeli SDO_ORDINATES. Vrednost odmika mora pripadati množici celih števil, predstavlja pa začetno točko elementa geometrije. Paziti moramo na pozicije, saj ima prvi element geometrije začetno os točke na poziciji 1 in ne na 0.

SDO_ETYPE označuje vrsto elementa. Element je lahko enostaven (1 - točka, 2 - črta, 1003 - notranji poligon in 2003 - zunanji poligon) ali pa je sestavljen (4 - črta in krožnica, 1005 - zunanji poligon sestavljen iz črt in krožnic, 2005 - notranji poligon sestavljen iz črt in krožnic, 1006 - dva ali več zunanja poligona in 2006 - dva ali več notranja poligona). V kombinaciji s **SDO_INTERPRETATION** številom v celoti določata element. Če je element enostaven, nam število SDO_INTERPRETATION pove natančen tip. Če je element sestavljen, SDO_INTERPRETATION določa število podelementov, ki ga sestavljajo. Primer sestavljene geometrije je na primer poligon z luknjo. Sestavljata ga lahko dva elementa. Prvi je zunanji, ki ga prepoznamo, če ima vrednost v tisočici enako 1, drugi, notranji, ima to vrednost enako 2. Notranjim elementom pravimo tudi luknje, saj predstavljajo območje, ki ga geometrija ne sme vsebovati. Zunanji elementi vsebujejo, poleg vozlišč in črt med njimi, še vrednosti v notranjosti.

Polje SDO_ORDINATES je definirano kot seznam koordinat:

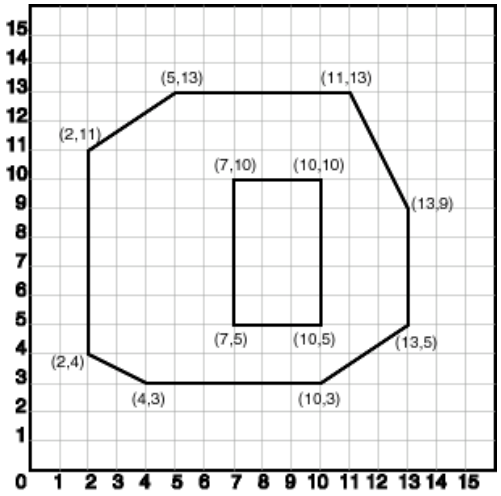
```
CREATE TYPE sdo_ordinate_array AS VARRAY (1048576) OF NUMBER;
```

Vrednosti v seznamu so urejene po dimenzijah. Dvodimenzionalni poligoni so shranjeni v obliki {X1, Y1 … Xn, Yn … X1, Y1}, medtem ko so tridimenzionalni shranjeni kot {X1, Y1, Z1 … Xn, Yn, Zn … X1, Y1, Z1}. Podatek o številu dimenzij je shranjen med meta podatki, do katerih lahko dostopamo preko pogleda xxx_SDO_GEOM_METADATA. Dvodimenzionalna geometrija ima lahko največ 524.288 vozlišč, tridimenzionalna 349.525 in štiridimenzionalna 262.144.

Koordinate notranjih elementov morajo biti shranjene v zaporedju nasprotnem od smeri urinega kazalca, medtem ko so koordinate zunanjih elementov shranjene v zaporedju enako smeri urinega kazalca. Tipi geometrij se morajo ujemati s tipom elementov, drugače jih Spatial ignorira. V splošnem to pomeni, da geometrija ne more biti tipa poligon, hkrati pa vsebovati elemente kot so na primer črta, točka … Veljavnost geometrij lahko preverimo z bazno funkcijo SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT.

V nadaljevanju so predstavljeni trije praktični primeri, kjer bomo prikazali, kako so geometrije shranjene na bazi.

Primer 1: Poligon z luknjo



Slika 4: Poligon z luknjo

Dani poligon (Slika 4) vsebuje naslednje podatke:

- SDO_GTYPE = 2003,
- SDO_SRID = NULL,
- SDO_POINT = NULL,
- SDO_ELEM_INFO = (1,1003,1, 19,2003,1),
- SDO_ORDINATES = (2,4, 4,3, 10,3, 13,5, 13,9, 11,13, 5,13, 2,11, 2,4, 7,5, 7,10, 10,10, 10,5, 7,5).

Geometrija je tipa 2003. 3 določa, da je geometrija poligon, 2 nam pove, da je dani poligon dvodimenzionalen lik.

Geometrija nima določenega koordinatnega sistema.

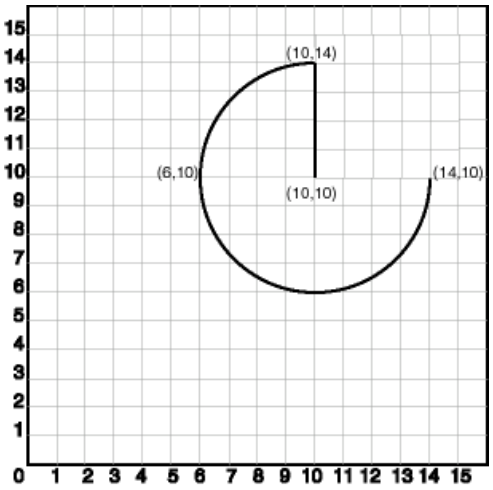
Ker je geometrija poligon, ima vrednosti shranjene v polju SDO_ORDINATES.

Podatke je potrebno brati iz polj SDO_ELEM_INFO in SDO_ORDINATES. V polju SDO_ELEM_INFO so shranjene lastnosti vseh elementov danega poligona v tako imenovanih trojčkih. Vsak trojček določa odmik od začetne točke, tip elementa in razširjeno interpretacijo.

Prvi trojček (1,1003,1) določa prvemu elementu začetno točko na prvi poziciji v polju SDO_ORDINATES. Os x začetne koordinate ima torej vrednost 2, y pa predstavlja naslednje število 4. Začetna točka je torej (2,4). Prvi element je tipa 1003 (zunanji poligon), kar pomeni, da spada med enostavne elemente. Razširjena interpretacija (1) nam pove, da je element povezan med vozlišči, zadnje vozlišče pa mora biti enako začetnemu. Vozlišča prvega elementa so torej (2,4), (4,3), (10,3), (13,5), (13,9), (11,13), (5,13), (2,11), (2,4), vrednosti pa so, poleg vozlišč in črt med vozlišči, vsebovane tudi v notranjosti lika.

Drugi trojček določa drugi element. Element je notranji poligon (2003) in ima začetno točko na 19. poziciji v polju SDO_ORDINATES (7,5). Drugi element je za razliko od prvega predstavljen kot luknja, kar pomeni, da se notranjost poligona (7,5), (7,10), (10,10), (10,5), (7,5) iz prvega poligona izvzame.

Primer 2: Sestavljene črte



Slika 5: Sestavljene črte

Dani poligon (Slika 5) vsebuje naslednje podatke:

- SDO_GTYPE = 2002,
- SDO_SRID = NULL,
- SDO_POINT = NULL,
- SDO_ELEM_INFO = (1,4,2, 1,2,1, 3,2,2),
- SDO_ORDINATES = (10,10, 10,14, 6,10, 14,10).

Geometrija je tipa 2002. Druga 2 določa, da je geometrija niz črt. Prva 2 nam pove, da je dana geometrija predstavljena v dvodimenzionalni obliki.

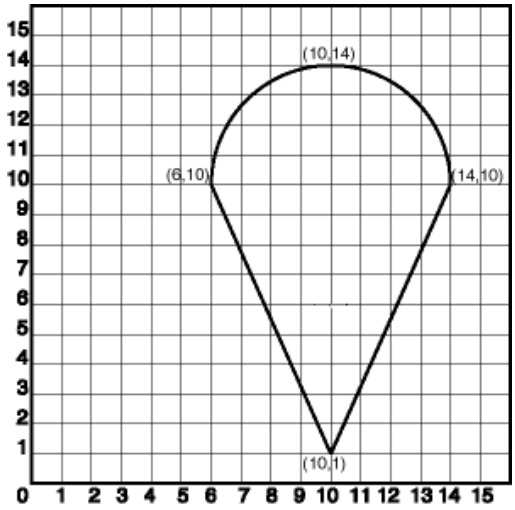
Geometrija nima določenega koordinatnega sistema.

Ker je geometrija poligon, ima vrednosti shranjene v polju SDO_ORDINATES.

Prvi trojček (1,4,2) nam pove, da je element tipa 4 (sestavljen niz črt). Ker je element sestavljen, mu njegova razširjena interpretacija določa naslednja dva podelementa:

- Prvi podelement ima začetno točko (10,10), ki se nahaja na prvi in drugi poziciji v tabeli SDO_ORDINATES. Podelement je tipa 2. Ker ima razširjena interpretacija vrednost 1, je podelement črta, kar pomeni, da naslednja točka (10,14) predstavlja njeno končno točko.
- Drugi podelement je tipa 2. Ker ima razširjena interpretacija vrednost 2, je podelement krožnica. Krožnica potrebuje 3 točke. Prva točka (10,14) predstavlja začetno točko, druga (6,10) je poljubna točka na krožnici, tretja (14,10) pa predstavlja zaključno točko.

Primer 3: Sestavljen poligon



Slika 6: Sestavljen poligon

Dani poligon (Slika 6) vsebuje naslednje podatke:

- SDO_GTYPE = 2003,
- SDO_SRID = NULL,
- SDO_POINT = NULL,
- SDO_ELEM_INFO = (1,1005,2, 1,2,1, 5,2,2),
- SDO_ORDINATES = (6,10, 10,1, 14,10, 10,14, 6,10).

Geometrija je tipa 2003. 3 določa, da je geometrija poligon. 2 nam pove, da je dana geometrija predstavljena v dvodimenzionalni obliki.

Geometrija nima določenega koordinatnega sistema.

Ker je geometrija poligon, ima vrednosti shranjene v polju SDO_ORDINATES.

Prvi trojček (1,1005,2) nam pove, da je element tipa 1005 (sestavljen poligon). Ker je element sestavljen, mu njegova razširjena interpretacija določa naslednja dva podelementa:

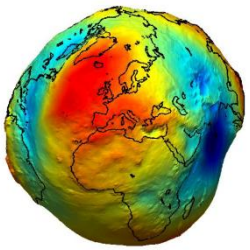
- Prvi podelement ima začetno točko (6,10), ki se nahaja na prvi in drugi poziciji v tabeli SDO_ORDINATES. Podelement je tipa 2. Ker ima razširjena interpretacija vrednost 1, je podelement črta. Naslednja točka (10,14) predstavlja vozlišče, s katerim se začetna točka poveže. Njegovo končno točko predstavlja začetna točka naslednjega podelementa (10,14).
- Drugi podelement ima začetno točko (10,14), ki predstavlja tudi končno točko prvega podelementa. Rišemo namreč povezan poligon, kjer je vsaka naslednja točka povezana s predhodno. Podelement je tipa 2. Ker ima razširjena interpretacija vrednost 2, je podelement krožnica. Krožnica vsebuje podatke o začetni točki (14,10), poljubni točki na krožnici (10,14) in končni točki (6,10). (7)

3.1.5 Koordinatni sistemi in transformacija

Koordinatni sistem so matematična pravila in dogovori, ki določajo način pripisovanja koordinat točkam. Koordinatni sistem je temelj za umestitev objektov in pojavov v prostor. Vsakemu objektu oziroma pojavu lahko določimo njegovo lego. Lega je v danem koordinatnem sistemu predstavljena kot točka, ki je definirana z nizom koordinat. Poznamo več vrst koordinatnih sistemov. Za opisovanje geografskih točk na Zemlji se uporabljajo kartezični ali pa geodetski koordinatni sistemi.

Kartezični koordinatni sistem je pravokotni koordinatni sistem, ki ga določata dve ali tri med seboj pravokotne osi. Imenujemo jih abscisna os (ali os x), ordinatna os (ali os y) in aplikatna os (ali os z). Število osi predstavljajo število dimenzij/razsežnosti. Dvodimenzionalnemu prostoru pravimo tudi kartezična ravnina. Vsaka točka je predstavljena z vrednostma dveh koordinat. Prva je projekcija točke na os x, druga pa projekcija točke na os y. Tridimenzionalni prostor prav tako lahko opišemo s kartezičnim produktom. Poleg osi x in y ga določa še os z. (8)

V 18. stoletju so na osnovi geodetskih meritev prišli do spoznanj, da Zemlja ni okrogla. Meritve so pokazale, da je Zemlja na polih sploščena in ob ekvatorju izbočena. Zaradi vrtenja je centrifugalna sila ob ekvatorju razmeroma velika, medtem ko je na polih skorajda ni. Ker centrifugalna sila nasprotuje sili teže ter tako zmanjšuje privlačno silo jedra, povzroča, da so snovi v notranjosti zemlje bolj oddaljene od središča v bližini ekvatorja. Njeno nepravilno obliko so poimenovali geoid, prikazuje ga Slika 7. (9)



Slika 7: Geoid (10)

Površine geoida ne moremo matematično definirati, zato si v geodetske namene Zemljo predstavljamo kot referenčni elipsoid. Referenčni elipsoid se poskuša Zemlji na določenem območju kar najbolje prilegati. Tabela 1 opisuje najpogostejše referenčne elipsoide. (11)

Geodetski polarni koordinatni sistem je Zemlji prirejeni koordinatni sistem. Geodetske koordinate (f, l, h) so definirane z normalo (pravokotnico) na elipsoid. Geodetska (elipsoidna) širina f je kot med normalo in ekvatorsko ravnino. Geodetska (elipsoidna) dolžina l je kot med ravninama izhodiščnega (greeenviškega) in krajevnega meridiana točke. Elipsoidna višina h je oddaljenost točke na površju Zemlje od elipsoida, vzeto po normali. (12)

Tabela 1: Vrste referenčnih elipsoidov (13)

Ime	Ekvatorska os (m)	Polarna os (m)	Inverz sploščenosti
Clarke 1866	6 378 206.4	6 356 583.8	294.978 698 2
Bessel 1841	6 377 397.155	6 356 078.965	299.152 843 4
International 1924	6 378 388	6 356 911.9	296.999 362 1
Krasovsky 1940	6 378 245	6 356 863	298.299 738 1
GRS 1980	6 378 137	6 356 752.3141	298.257 222 101
WGS 1984	6 378 137	6 356 752.3142	298.257 223 563
Krogla	6 371 000	6 371 000	∞

Danes je najbolj uveljavljen koordinatni sistem WGS 84 (angl. World Geodetic System). Uporabljajo ga vsi sateliti, nadzorni centri in sprejemniki GPS. Središče WGS 84 koordinatnega sistema poteka v težišču zemlje in se vrti skupaj z njo. Višina v tem sistemu je določena kot pravokotna oddaljenost točke od elipsoida, ki določa Zemljo. (13)

Slovenija je z začetkom leta 2008 prešla na nov koordinatni sistem. Gaus-Kruegerjev koordinatni sistem je zamenjal sodobnejši ETRS89. (14)

Osnovni podatki o koordinatnih sistemih so shranjeni v tabelah SDO_COORD_SYS in SDO_COORD_REF_SYS, ki sta prav tako del sheme MDSYS. (7)

Transformacija med koordinatnimi sistemi je postopek, s katerim koordinatni sistem pretvorimo v drug koordinatni sistem. Transformacija je potrebna v primerih, ko točke ne pripadajo istemu koordinatnemu sistemu. Prostorska tabela, v kateri imamo shranjene podatke o slojih, se preko polja SDO_SRID povezuje s pogledom CS_SRS. V pogledu CS_SRS so v polju WKTEXT shranjene definicije koordinatnega sistema, ki ga Oracle Spatial uporablja pri transformaciji. V spodnjem primeru je predstavljena transformacija iz koordinatnega sistema WGS 1984 (SRID = 8307) v koordinatni sistem GRS 1980 (SRID = 4019). S funkcijo SDO_CS.TRANSFORM_LAYER lahko izvozimo vse podatke sloja, ki so transformirani v drug koordinatni sistem, v novo tabelo.

Definicija (WKTEXT) za koordinatna sistema WGS 1984 in GRS 1980:

```
GEOGCS [
  "Longitude / Latitude (WGS 84)",
  DATUM ["WGS 84", SPHEROID ["WGS 84", 6378137, 298.257223563]],
  PRIMEM [ "Greenwich", 0.000000 ],
  UNIT ["Decimal Degree", 0.01745329251994330]]

GEOGCS [
  "Unknown datum based upon the GRS 1980 ellipsoid",
  DATUM ["Not specified (based on GRS 1980 ellipsoid) (EPSG ID 6019)",
    SPHEROID ["GRS 1980 (EPSG ID 7019)", 6378137, 298.257222101]],
  PRIMEM [ "Greenwich", 0.000000 ],
  UNIT ["Decimal Degree", 0.01745329251994328]]
```

Primer klica transformacije:

```
CALL SDO_CS.TRANSFORM_LAYER('mesta', 'geometry', 'mesta_4019', 4019);
```

3.2 Oracle MapViewer

Oracle Mapviewer je orodje, namenjeno izrisovanju (angl. rendering) zemljevidov. Je sredstvo za vizualizacijo geografskih podatkov, s katerimi upravljata Oracle Spatial oziroma Oracle Locator. Oracle Mapviewer je aplikacija J2EE, ki jo lahko namestimo na vsebovalnik J2EE (angl. J2EE container).

Glavne komponente, ki jih Oracle Mapviewer vsebuje, so:

- SDOVIS,
- MapBuilder,
- aplikacijski vmesniki (API),
- Oracle Maps.

Če želimo namestiti Oracle Mapviewer, potrebujemo:

- strežnik Oracle Application Server 10g Release 3 (10.1.3) ali novejšo različico;
- bazo Oracle 9i ali novejšo, ker ima vključena Oracle Spatial in Oracle Locator;
- klient Oracle 9i ali novejši;
- Java SDK 1.5 ali novejšo.

Oracle Mapviewer lahko namestimo na:

- WebLogic Server 10 ali novejšo različico;
- Oracle Fusion Middleware;
- samostojen OC4J, ki predstavlja okrnjeno različico Oraclovega spletnega strežnika in vsebovalnika J2EE.

OC4J je komponenta J2EE strežnika Oracle Application Server v izvajanju. Komponento dobimo na Oraclovi spletni strani. Če imamo nameščeno Javo 1.5 SDK, lahko strežnik zaženemo z ukazom »java –jar oc4j.jar«. OC4J vsebuje svoj spletni vmesnik, preko katerega lahko namestimo MapViewer, podobno, kot če bi ga nameščali na Oracle Fusion Middleware. Razlikuje se le v tem, da je pri nameščanju aplikacije MapViewer na Oracle Fusion Middleware potrebno določiti instanco OC4J. Oracle Fusion Middleware nam omogoča izbrati obstoječo instanco OC4J, ali pa nam kreira novo.

Tudi Mapviewer lahko dobimo na Oraclovi spletni strani. Imenuje se »Oracle Application Server MapViewer & Map Builder«. Trenutno je na voljo najnovejša verzija 11.1.1.2. Zip dokument vsebuje datoteko ear »mapviewer.ear«. Arhivna datoteka ear se uporablja pri nameščanju enega ali več modulov. Pri nameščanju aplikacije moramo biti pozorni, da podamo pravo pot do datoteke »mapviewer.ear«. Primer namestitve MapViewer aplikacije prikazuje Slika 8.

Slika 8: Primer namestitve Oracle MapViewerja na strežnik Oracle Application Server

3.2.1 Koncept

Za razvoj aplikacij Oracle Mapviewer je potrebno razumeti njegov bazni koncept. Mapviewer ima vse definicije shranjene na bazi kot metapodatke. Definicije določajo velikosti zemljevidov, uporabljen koordinatni sistem, uporabljene sloje, stile posameznih slojev, seznam prikazovanih polj posameznih slojev …

Najpomembnejši metapodatki, s katerih MapViewer pridobi največ informacij, so:

- stili,
- sloji in
- zemljevidi.

Podatke o elementih lahko urejamo preko pogledov. Vsak pogled vsebuje polje clob (angl. Character Large Object), ki opisuje definicijo elementa in je predstavljeno v obliki XML. Za poenostavljeno gradnjo metapodatkov je priporočljiva uporaba orodja MapBuilder [3.2.3]. Orodje je zelo prijazno za uporabo, uporabljajo ga lahko tudi uporabniki, ki nimajo znanja o samem konceptu, na primer grafični oblikovalci in podobno.

Stili

V pogledu **USER_SDO_STYLES** lahko določimo stile. Stili oblikujejo geometrije, ki pripadajo določenem sloju. Ker je na zemljevidu lahko več slojev z različnimi geometrijami, moramo vsakemu določiti najprimernejši stil. Vse vrste stilov opisuje Tabela 2.

Tabela 2 : Vrste stilov

Kratica	Tip	Pomen	Uporaba
C	COLOR	barva	stil robov in/ali notranjosti
M	MARKER	označevalnik	opisovanje točkovnih geometrij
L	LINE	črta	opisovanje črtnih geometrij (ceste, reke …)
A	AREA (patern)	področje	stil vzorca notranjosti
T	TEXT	tekst	stil pisave
V	ADVANCED	napredno	napredno označevanje, ki je odvisno od podatkov

Za lažji razvoj je priporočljivo stilom določiti ime v formatu »KRATICA.IME« na primer L.REKE, C.DRZAVE ... Definicija stila je določena v polju DEFINITION v obliki XML. Parser XML obravnava le element (značko) <g>. Element mora imeti atributa:

- class, ki stilu določi tip in
- style, ki opisuje njegovo obliko.

Primer: Zapis XML, ki opisuje sloj

```
<?xml version="1.0" standalone="yes"?>
 <svg width="1in" height="1in">
   <desc></desc>
   <g class="color" style="fill:#0000ff">
     <rect width="50" height="50"/>
   </g>
 </svg>
```

Sloji

Podatki o slojih so združeni v pogledu **USER_SDO_THEMES**. Oracle MapViewer podpira:

- predefinirane sloje,
- JDBC sloje,
- slikovne sloje (angl. image themes),
- georastrske sloje,
- mrežne sloje,
- topološke sloje,
- WFS sloje,
- »annotation« sloje,
- WMS sloje,
- sloje »po meri«.

V polju BASE_TABLE ima vsak sloj podano ime tabele, kjer so shranjeni geometrijski podatki. Pravila, ki definirajo, katere stile sloj uporablja, lahko pridobimo v polju STYLING_RULES. Pravila so opisana v obliki XML.

Vsakemu pravilu lahko določimo:

- stil, ki smo ga definirali v pogledu USER_SDO_STYLES in
- filter, kjer se lahko omejimo po podatkih v geometrijski tabeli. Če filter vrača 1 (true), se sloj prikazuje s tem stilom.

```
SQL> SELECT * FROM USER_SDO_THEMES;

NAME              BASE_TABLE GEOMETRY_COLUMN
STYLING_RULES
----------------- ---------- ---------------
THEME_MESTA       MESTA      LOKACIJA
<?xml version="1.0" standalone="yes"?>
<styling_rules>
  <rule>
    <features style="M.VSA_MESTA_1"> (populacija between 200000
                                                AND 1000000 )
    </features>
    <label column="naziv" style="T.VSA_MESTA_1"> 1 </label>
  </rule>
  <rule>
    <features style="M.VSA_MESTA_2"> ( populacija &lt; 200000)
    </features>
    <label column="naziv" style="T.VSA_MESTA_2"> 1 </label>
  </rule>
</styling_rules>
```

Zemljevidi

Pogled **USER_SDO_MAPS** ima v polju DEFINITION naštete sloje, ki se enotno obravnavajo kot osnovni sloj. Vsakemu sloju je potrebno določiti tudi lestvico prikaza (angl. scale), s katere bo MapViewer razbral, ali se geometrije sloja pri določeni velikosti prikaza (angl. zoom) prikažejo.

```
<?xml version="1.0" standalone="yes"?>
<map_definition>
    <theme name="THEME_MESTA" min_scale="1.5E8" max_scale="0.0"
 scale_mode="RATIO"/>
    <theme name="THEME_DRZAVE" min_scale="8500000.0" max_scale="0.0"
                scale_mode="RATIO"/>
    <theme name="THEME_CESTE" min_scale="1.0E8"
                max_scale="4.5E7" scale_mode="RATIO"/>
    <theme name="THEME_REKE" min_scale="4.5E7" max_scale="0.0"
                scale_mode="RATIO"/>
 </map_definition>
```

3.2.2 SDOVIS

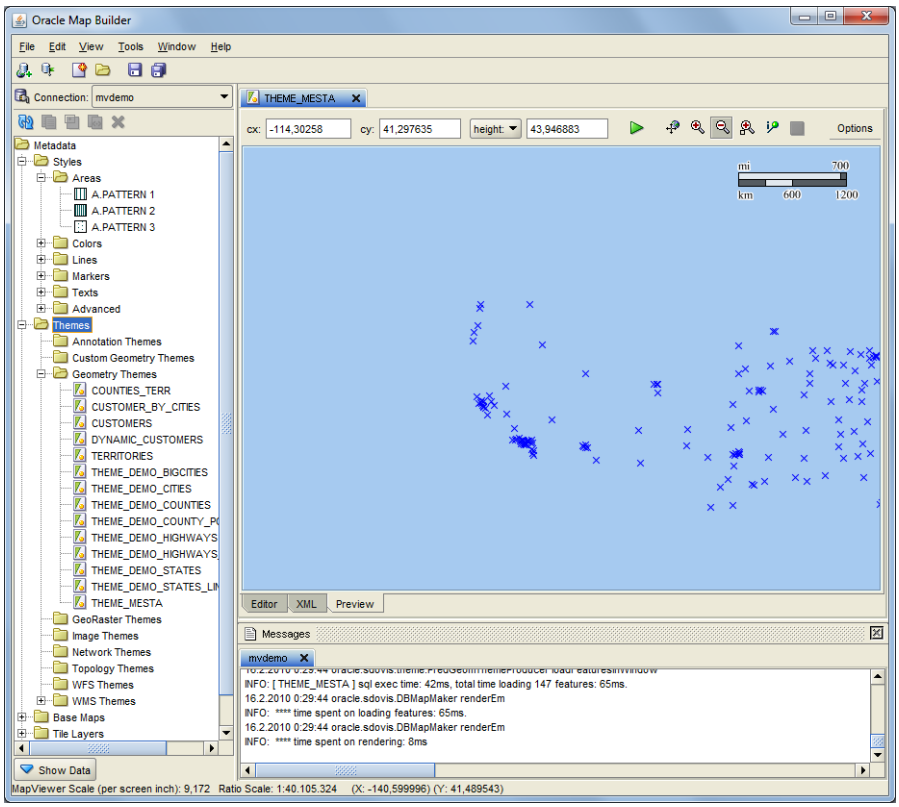
SDOVIS je jedro, ki omogoča izris kartografskih slik. Nahaja se v direktoriju »$MAPVIEWER/web/WEB-INF/lib« kot javin arhiv jar. Mapviewer podpira formate PNG, GIF in JPEG. Če želimo uporabiti drug format, ga moramo implementirati preko vmesnika »oracle.sdovis.CustomImageRenderer«:

```
public interface CustomImageRenderer {
   public String getSupportedFormat();
   public void renderImages(Graphics2D g2, byte[][] images,
                            double[][] mbrs,
                            Rectangle2D dataWindow,
                            Rectangle2D deviceView,
                            AffineTransform at,
                            boolean scaleImage);
}
```

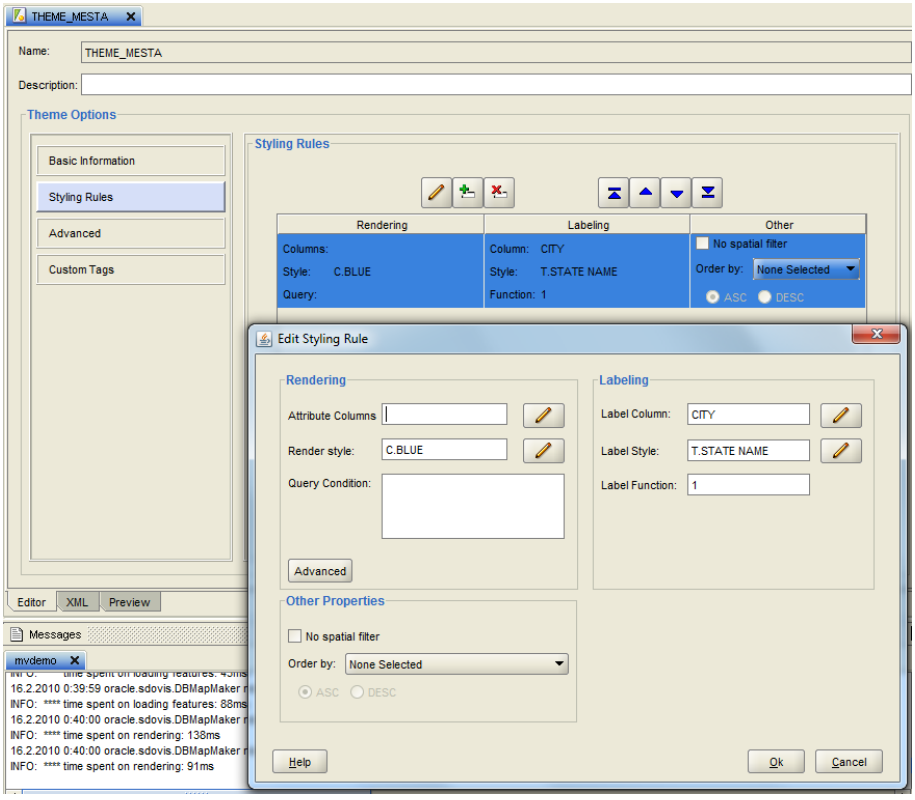
3.2.3 MapBuilder

MapBuilder je samostojna aplikacija, v kateri lahko kreiramo in upravljamo z metapodatki, ki so shranjeni na bazi Oracle in jih uporablja Oracle Mapviewer. Aplikacija olajša in pospeši razvoj. Uporabniku ni potrebno poznati koncepta MapViewerja do potankosti. Dovolj je, da pozna odvisnosti med posameznimi elementi (stili, sloji, zemljevidi).

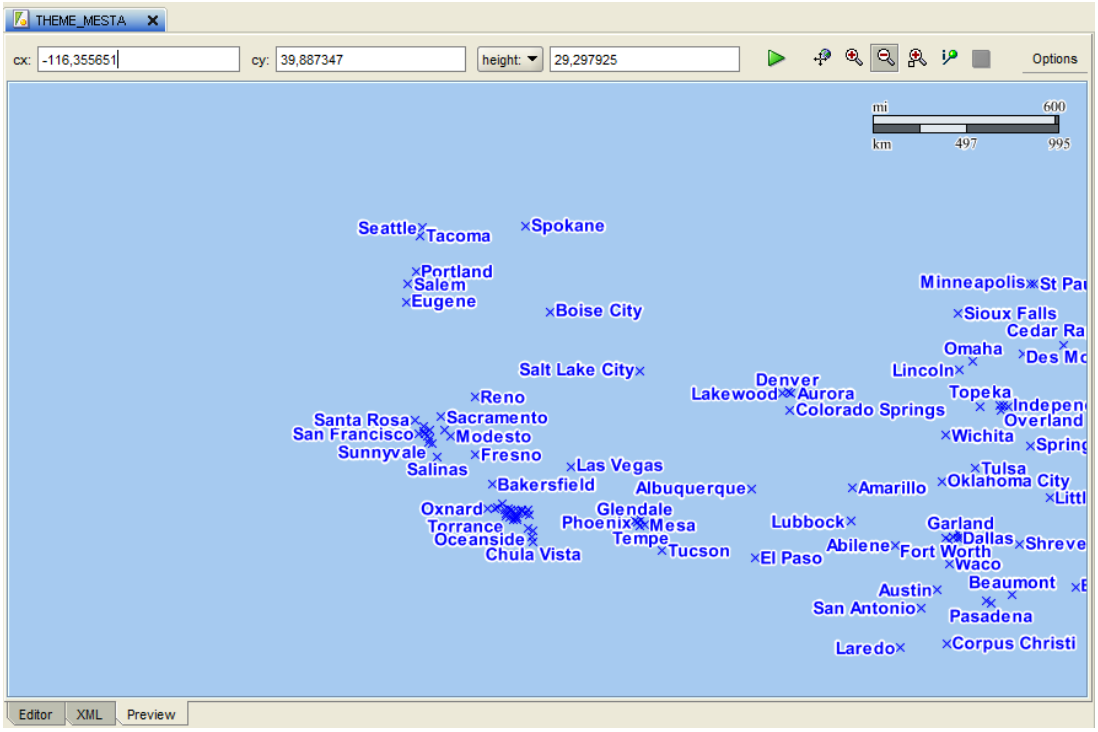
Oracle MapBuilder dobimo na Oraclovi spletni strani. Jar datoteko zaženemo z ukazom »java –jar mapbuilder.jar«. Preko aplikacije je potrebna povezava na shemo Oracle baze. Oracle MapBuilder prebere metapodatke in jih naloži na repozitorij, ki se nahaja na levi strani. Repozitorij in ostali podatki, do katerih pridemo iz njega, so urejeni tako, da lahko kar najlažje definiramo vse metapodatke, ki jih uporablja Oracle MapViewer. Najvišji nivo predstavljajo glavni elementi koncepta MapViewer (stili, sloji, zemljevidi). Podelementi so enaki podelementom [3.2.1]. V spodnjem primeru je predstavljen postopek, s katerim vsakemu elementu (mestu) podamo ime (Slika 9, Slika 10, Slika 11).



Slika 9: Prikaz točk mest, Oracle MapBuilder



Slika 10: Dodajanje lastnosti stilov slojem, Oracle MapBuilder



Slika 11: Prikaz točk mest z nazivi, Oracle MapBuilder

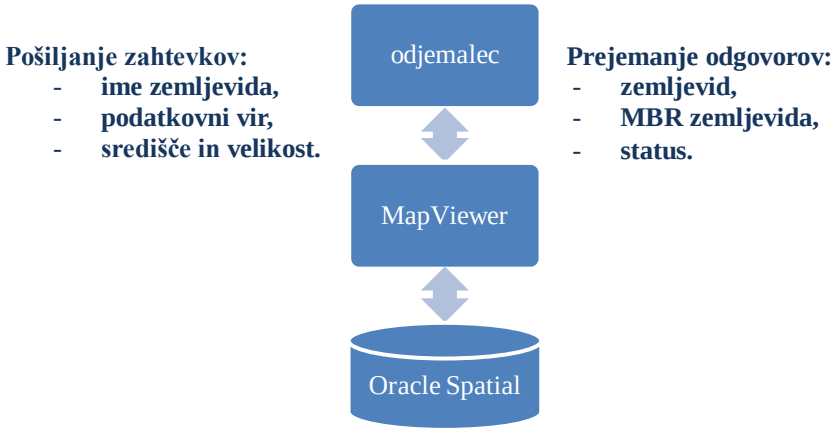
3.2.4 Aplikacijski vmesniki

Aplikacijski vmesniki (v nadaljevanju vmesniki API) omogočajo programski dostop do Mapviewerjevih funkcionalnosti. Omogočen je dostop preko:

- XML,
- Jave,
- PL/SQL in
- JavaScripta, ki temelji na tehnologiji Ajax.

Vmesnik XML

Dostop preko vmesnika XML deluje po principu zahtevek–odgovor, ki ga prikazuje Slika 12.



Slika 12: MapViewer zahtevki in odgovori

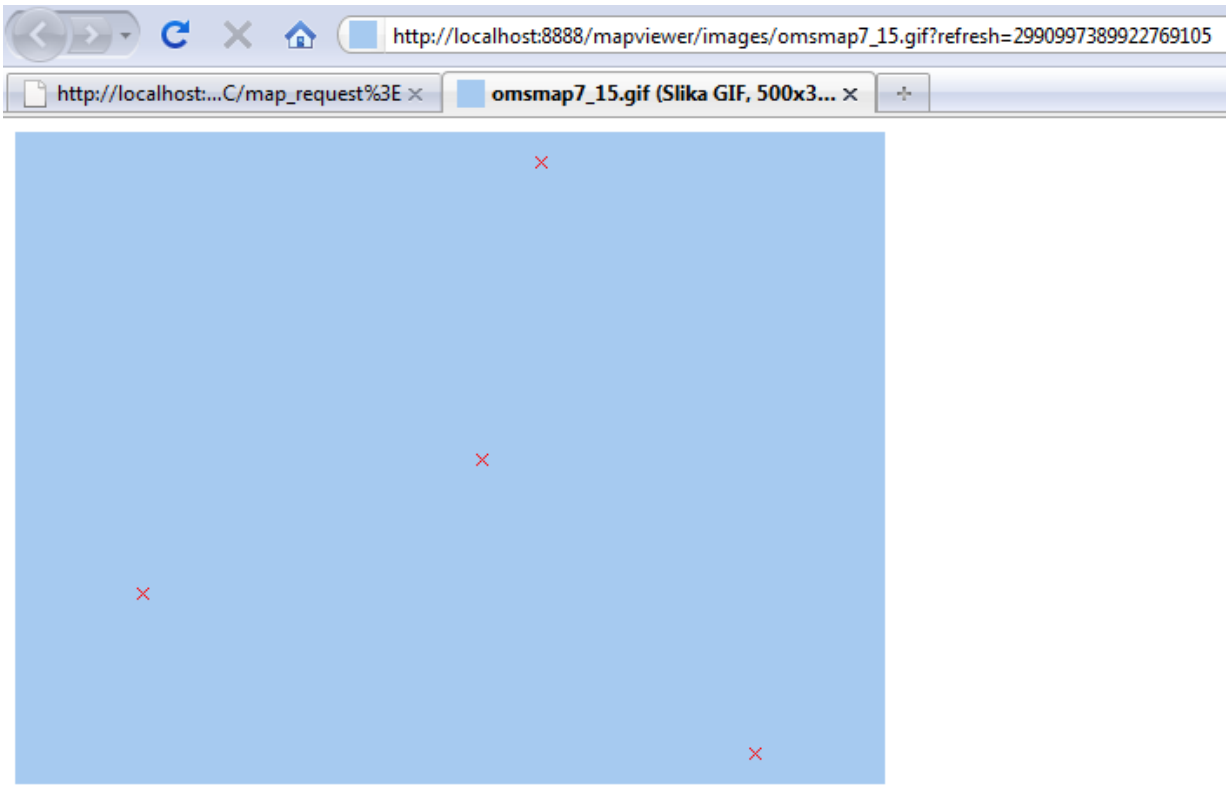
MapViewerju zahtevke podajamo v obliki XML. V zahtevku morajo biti podani podatkovni vir (angl. data source) in vsaj definiran sloj ali središče ali geometrija. Strežnik prejeti dokument XML obdela, generira sliko in vrne odgovor v obliki dokumenta XML. V spodnjem primeru je podan enostaven zahtevek XML, kjer pričakujemo odgovor XML (Slika 13) in sliko (Slika 14). V tabeli TOCKE imamo podane štiri geometrije za točke, ki imajo obliko črke x. Točke, ki sestavljajo dinamični sloj (JDBC), pridobimo iz vprašanja SQL.

Zahtevek:

```
<?xml version="1.0" encoding="UTF-8" ?>
  <map_request datasource="mvdemo">
    <themes>
      <theme name="TOCKE">
        <jdbc_query spatial_column = "SHAPE" datasource = "mvdemo">
            SELECT SHAPE FROM ORIENTED_POINTS
        </jdbc_query>
     </theme>
  </themes>
</map_request>
```

Odgovor:

Slika 13: MapViewerjev odgovor na zahtevo XML



Slika 14: Slika, ki se generira skupaj z odgovorom na zahtevek XML

JavaBean vmesnik API

Do MapViewerja lahko dostopamo tudi preko JavaBean vmesnika API. Vmesnik API omogoča uporabniku prijaznejšo komunikacijo z MapViewerjem. Uporabniku ni potrebno podajati zahtevkov v obliki dokumentov XML. Olajšano je tudi delo pri pridobivanju želenih rezultatov, kot so razne informacije in slike.

```
import oracle.lbs.mapclient.MapViewer;
…
MapViewer mv = new MapViewer
          ("http://localhost:8888/mapviewer/omserver");
mv.addJDBCTheme("mvdemo","TOCKE","SELECT SHAPE FROM
   ORIENTED_POINTS","SHAPE","0","C.RED",null,null,false);
mv.run();
Image img = mv.getGeneratedMapImage();
…
```

Sam razvoj si lahko še nekoliko optimiziramo z uporabo JSP Tag Libraryja. Aplikacija postane razumljivejša, razvoj se lahko pospeši. Ker pa ni možno implementirati vsega, kar znata JavaBean in XML vmesnika API, njegovo uporabo ne priporočamo.

PL/SQL vmesnik API

Zahtevke lahko pošiljamo in sprejemamo odgovore tudi preko PL/SQL vmesnika API. Za delo znotraj Oracle baze je potrebno naložiti paket SDO_MVCLIENT. Ko je paket naložen, lahko, podobno kot pri ostalih vmesnikih, pošiljamo zahtevke in sprejemamo odgovore.

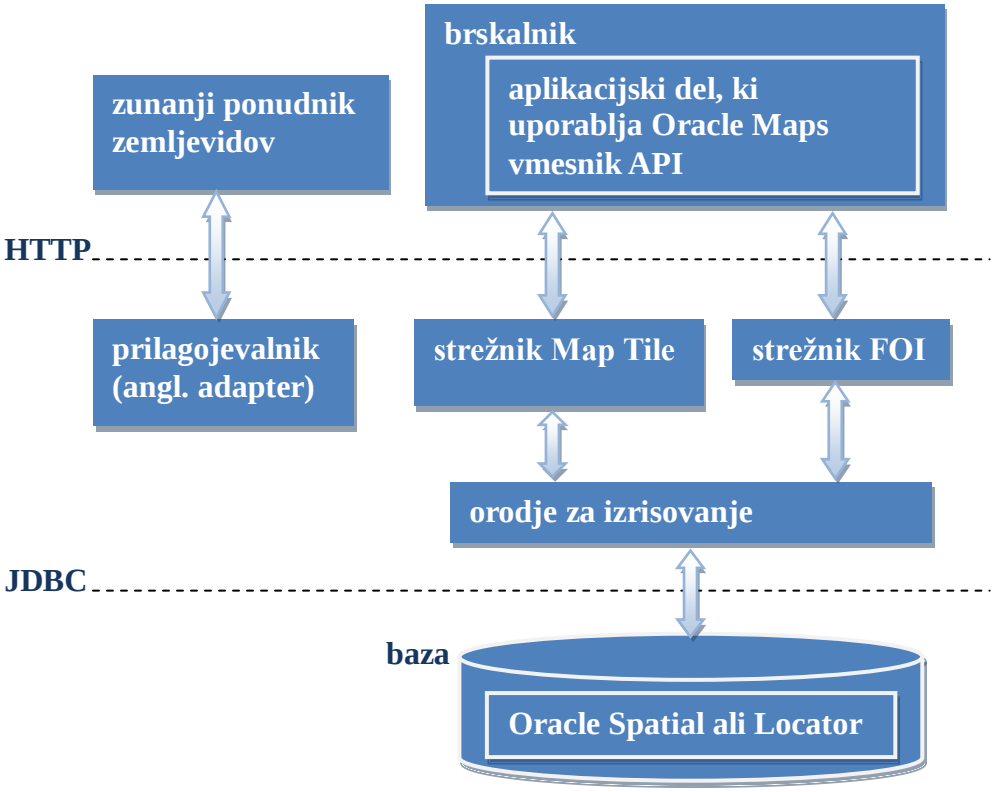
```
Call sdo_mvclient.createmapviewerclient
             (http://localhost:8888/mapviewer/omserver');
call sdo_mvclient.setDataSourceName('mvdemo');
select sdo_mvclient.addJDBCTheme('mvdemo', 'TOCKE',
 'SELECT SHAPE FROM ORIENTED_POINTS',
 'SHAPE', '8307', 'C.RED', null, null, 'FALSE') from dual;
select sdo_mvclient.run() from dual;
select sdo_mvclient.getgeneratedMapImageURL() from dual;
```

JavaScript vmesnik API

JavaScript vmesnik API je del komponente Oracle Maps in bo podrobneje predstavljen v poglavju 3.2.5 - Oracle Maps.

3.2.5 Oracle Maps

Oracle Maps je zbirka tehnologij, ki so vključene k Oracle MapViewerju. Z njihovo pomočjo lahko razvijemo interaktivne spletne aplikacije GIS. Arhitekturo Oracle Maps aplikacij prikazuje Slika 15.



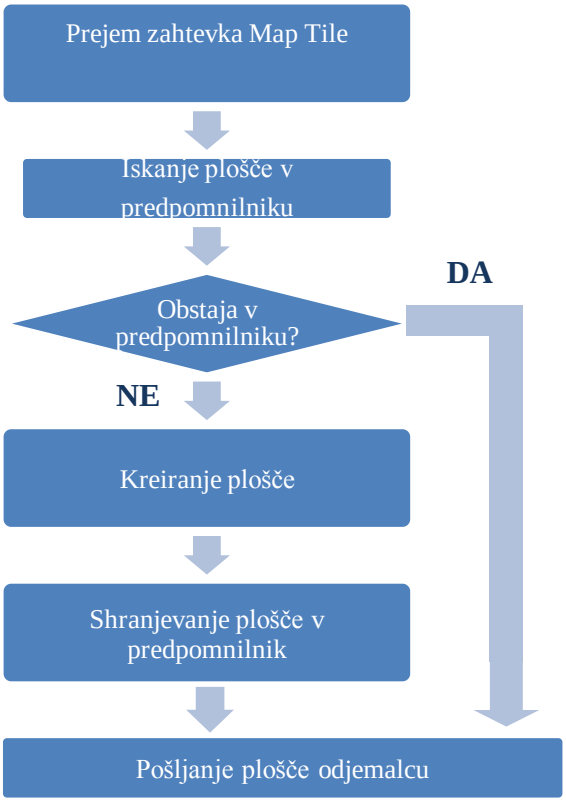
Slika 15: Aritektura Oracle Maps aplikacije

Oracle Maps je sestavljen iz treh glavnih komponent:

- strežnika Map Tile,
- strežnika FOI,
- JavaScript knjižnic, ki temeljijo na tehnologiji Ajax.

Strežnik Map Tile

Map Tile strežnik je komponenta, ki razpolaga z začasno shranjenimi slikami v predpomnilniku (angl. Map Tile – plošče). Strežnik je vzpostavljen zaradi hitrejšega dostopa, saj aplikaciji ni potrebno večkrat izrisovati enakih slik. Strežnik Map Tile prejme zahtevo v obliki XML in kot odgovor odjemalcu pošlje sliko oziroma ploščo. Postopek opisuje Slika 16.



Slika 16: Diagram poteka strežnika Map Tile

Celoten zemljevid je sestavljen iz mreže slik (plošč), ki skupaj sestavljajo celoto (Slika 17). Slike so označene z velikostjo prikaza in šifro lokacije (angl. mash code). Lokacije izračuna klient JavaScript, ki pošlje vse zahtevke strežniku. Ta preveri, če dana slika obstaja v predpomnilniku. Če slike še ni, jo mora izrisati in dodati v predpomnilnik. Sliko nato pošlje odjemalcu.

Slika 17: Šifre lokacij slik

Strežnik FOI

Strežnik FOI teče kot Java servlet. Za razliko od strežnika Map Tile, ki upravlja s slikami na osnovnem sloju, strežnik FOI odgovarja na tako imenovane zahtevke FOI, ki se nahajajo nad osnovnim slojem. Na slojih FOI so lahko prikazane vse Spatial oblike (točke, črte, poligoni). Sloj je lahko predefiniran, lahko pa ga tudi dinamično nastavljamo (JDBC).

JavaScript knjižnice in JavaScript vmesnik API

JavaScript knjižnice omogočajo prikaz in interakcijo s sloji MapViewerja. Brskalnik prikazuje sestavljeni zemljevid. JavaScript knjižnice:

- dobijo zemljevid iz strežnika Map Tile in ga prikažejo kot osnovni sloj na brskalniku;
- pošiljajo zahtevke FOI in dobljene slike prikazujejo nad osnovnim slojem;
- omogočajo interakcijo nad celotnim zemljevidom (navigacija, klikanje, risanje ...).

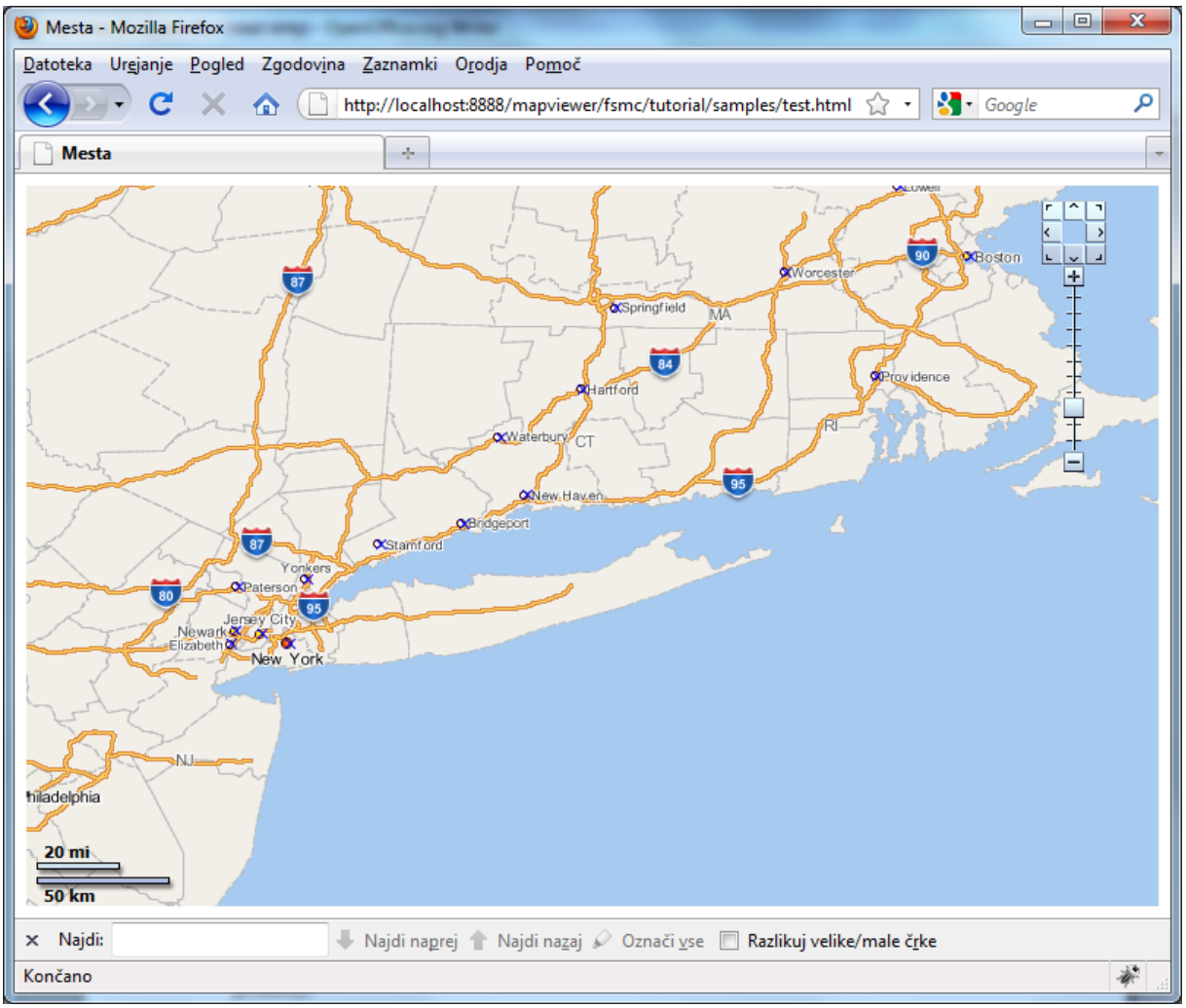
Za dostop do vseh teh funkcionalnosti lahko uporabljamo JavaScript vmesnik API, ki je sestavljen iz številnih razredov:

- MVMapView implementira večino kontrol vmesnika za GIS;
- MVMapTileLayer razred definira sloj plošč;
- MVThemeBasedFOI definira predefinirane sloje FOI;
- FOI definira sloje FOI, definirati ga moramo v celoti;
- MVSdoGeometry definira Spatial geometrijske objekte;
- MVRedLineTool definira orodje za risanje poligonov;
- MVRectangleTool definira orodje za risanje pravokotnikov;
- MVOverviewMap definira komponento, ki prikazuje zemljevid v predogledu;
- MVMapDecoration definira dodatke za zemljevid. (15)

Primer preproste aplikacije (Slika 18):

```
<HTML>
  <HEAD>
    <TITLE>Mesta</TITLE>
  <SCRIPT language="Javascript"
      src="/mapviewer/fsmc/jslib/oraclemaps.js"></SCRIPT>
  <SCRIPT language=javascript>
    function init(){
      var baseURL = "http://"+document.location.host+"/mapviewer";
      // kreiramo MVMapView instanco za prikaz zemljevida
      var mapview = new MVMapView(document.getElementById("map"),
                                  baseURL);
      // dodamo osnovni sloj v ozadje
      mapview.addMapTileLayer(new MVMapTileLayer("mvdemo.demo_map"));
      // dodamo predefiniran FOI sloj
      var themebasedfoi = new MVThemeBasedFOI('themebasedfoi1',
                                              'mvdemo.mesta');
      themebasedfoi.setBringToTopOnMouseOver(true);
      mapview.addThemeBasedFOI(themebasedfoi);
      // nastavimo središče in definiramo koordinatni sistem, ki ga
      // bomo uporabljali
```

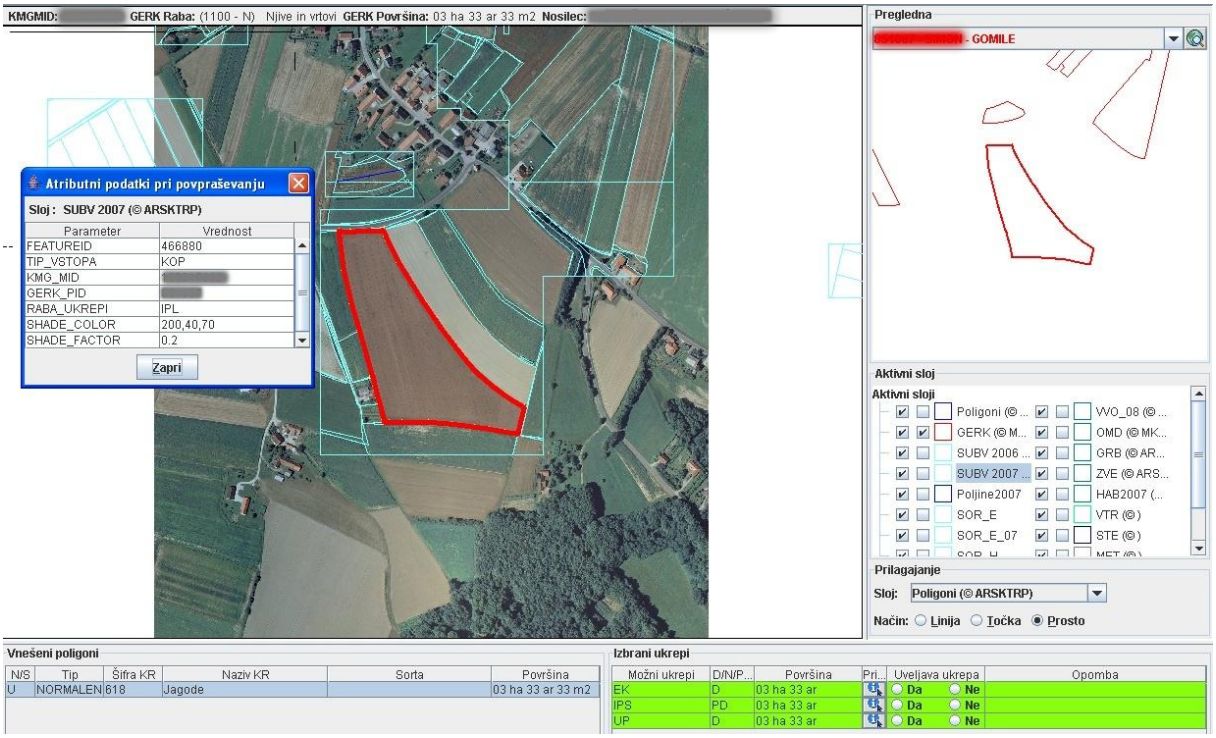
```
        mapview.setCenter(MVSdoGeometry.createPoint(-122.45,
                                                    37.7706,8307));
        //nastavimo velikost prikaza
        mapview.setZoomLevel(4);
        // dodamo orodje za navigacijo na desno stran zemljevida
        mapview.addNavigationPanel('east');
        // dodamo mero
        mapview.addScaleBar();
        // prikažemo zemljevid
        mapview.display();
      }
    </SCRIPT>
  </HEAD>
  <BODY onload="javascript:init()">
        <DIV id="map" style="left:0px; top:0px; width:100%;
             height:100%"></DIV>
  </BODY>
</HTML>
```



Slika 18: Primer Oracle Maps aplikacije

4 Programska rešitev AgriMap

Za izdelavo aplikacije GIS z Oraclovimi tehnologijami je potrebno poznati osnove Oracle Spatiala [3.1] ter nekoliko bolj podrobno Oracle MapViewer [3.2]. Cilj diplomskega dela ni vzpostavitev GIS-a, ki bi nadomestil obstoječega (Slika 19), saj se zavedamo, da bi bila izdelava take aplikacije preobsežna za diplomsko delo. Naša želja je razviti predstavitveno aplikacijo, na podlagi katere se bomo potem lažje odločili, ali bi jo bilo sploh smiselno uvesti. Pri odločanju bomo upoštevali predvsem performančni vidik in možnosti nadgrajevanja.



Slika 19: Obstoječa aplikacija GIS – STE (16)

Aplikacijo smo poimenovali AgriMap. AgriMap mora za potrebe predstavitve prikazovati 16 slojev, ki se morajo med sabo prekrivati. Vsak sloj vsebuje samo geometrije tipa:

- enostaven poligon in
- sestavljen poligon, kot je na primer poligon z luknjo.

Da bodo pri prekrivanju razvidne tudi geometrije spodnjih slojev, jim je potrebno določiti prosojno barvo. Barve morajo biti določene tako, da se bo vseh 16 slojev med seboj jasno razlikovalo.

AgriMap se mora obnašati kot nekakšen portal, kjer bo:

- omogočena uporaba vseh osnovnih funkcionalnosti, ki jih podpira Oracle Maps (prikaz, približevanje, navigacija);
- možno izbirati sloje, ki se prikazujejo;
- možno pridobivati podatke posameznih slojev;

- omogočeno iskanje po zemljevidu (šifra GERK-a, lokacija);
- omogočeno risanje poligonov.

4.1 Razvojno okolje

Najprej je potrebna namestitev komponente MapViewer na vsebovalnik J2EE. Ker smo se odločili, da bomo na začetku razvijali aplikacijo lokalno, smo v ta namen uporabili strežnik OC4J. MapViewer komponento »mapviewer.ear« smo dobili na Oraclovi spletni strani in jo namestili na vsebovalnik strežnika OC4J [3.2]. Kot končni korak pri vzpostavitvi razvojnega okolja je potreben le še zagon strežnika. Izvedemo ga lahko z ukazom »java -server -Xmx384M -jar oc4j.jar«.

AgriMap smo razvijali in testirali v operacijskem sistemu Windows 7 na brskalnikih Firefox 3.5.7 in Internet Explorer 8.

AgriMap se povezuje na Oracle Database 11*g* Enterprise Edition, kjer so shranjeni vsi podatki, ki jih bomo uporabljali pri diplomskem delu.

Izdelavo in urejanje metapodatkov smo si močno olajšali z uporabo orodja Oracle MapBuilder [3.2.3].

Interaktivna spletna stran AgriMap bo temeljila na knjižnicah Oracle Maps JavaScript, predstavljenih v poglavju 3.2.5. Uporabljala bo JavaScript vmesnik API.

Aplikaciji bomo dodali funkcionalnost iskanja. Poizvedovali bomo preko tehnologije Ajax, ki bo sinhrono vračala odgovore, prejete iz strežniške javine strani (JSP). JSP se bo povezoval na bazo Oracle in vračal koordinatne točke.

4.2 Definicija metapodatkov

Metapodatke, ki jih MapViewer uporablja, lahko določamo z orodjem Oracle MapBuilder. Potrebno je definirati lastnosti za:

- stile,
- sloje in
- zemljevide.

4.2.1 Stili

S stili določamo lastnosti prikazovanja slojev. Vsak sloj v AgriMapu mora biti obarvan drugače. Zaradi možnosti prikazovanja več slojev bomo barvam dodali prosojnost. Vsakemu sloju določimo v notranjosti barvo in motnost (angl. opacity), s katero dosežemo prosojnost. Robovom določimo enako barvo, le da je ta brez motnosti. Barvo poimenujemo v formatu C.SLOJ, na primer C.GERK.

Določiti moramo še stil teksta, ki se bo izpisoval na zemljevidu in bo določal geometrije posameznih slojev. Tekstom slojev določimo enako barvo in jih poimenujemo v formatu T.SLOJ, na primer T.GERK.

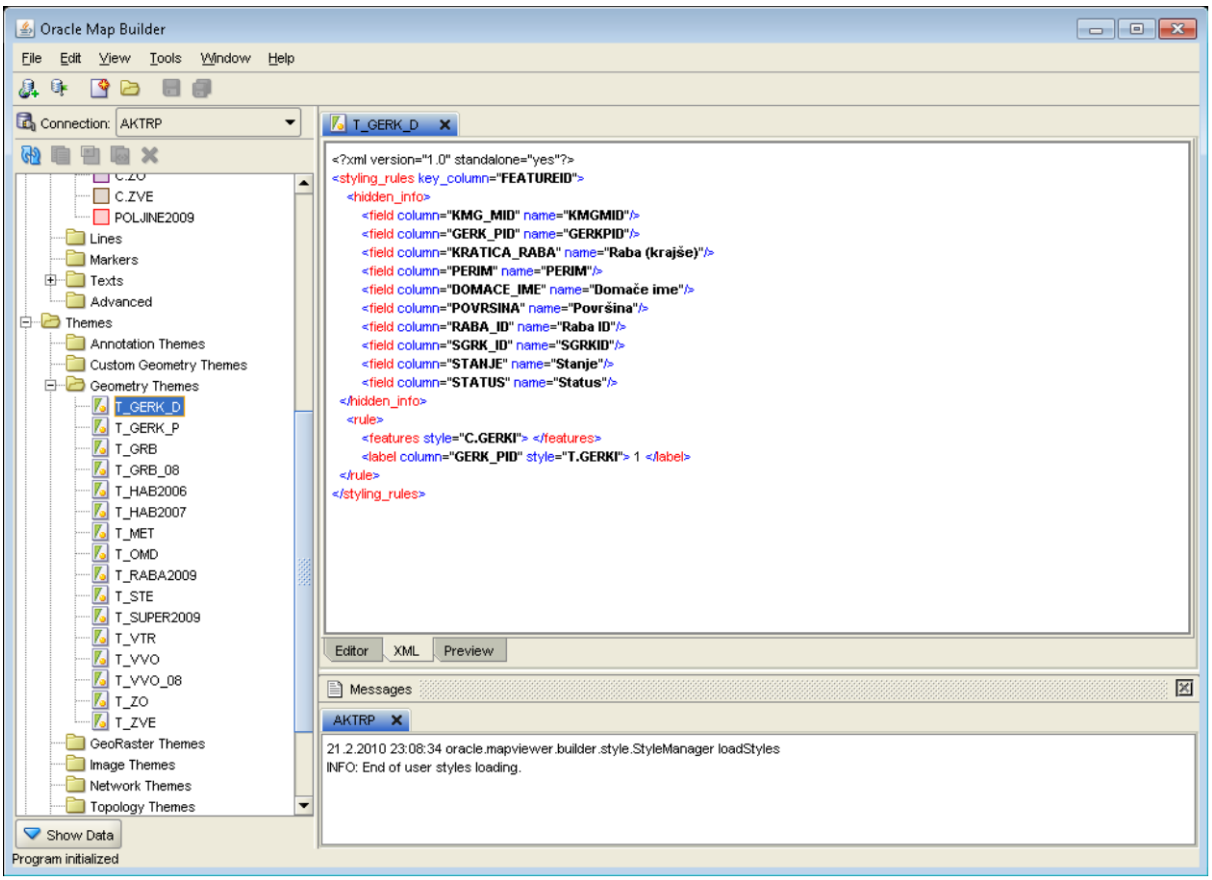
4.2.2 Sloji

Slojem, ki pripadajo določenim geometrijskim tabelam, pravimo geometrijski sloj (angl. geometry theme). V MapBuilderju jih izdelamo preko čarovnika. Vsaki izmed njih navedemo tabelo in polje, kjer se nahajajo geometrijski podatki. Ime ji določimo v formatu THEME_SLOJ, na primer THEME_GERK. Nato nam čarovnik ponudi možnost izbire barve sloja. Na tem mestu izberemo stil, ki smo ga že definirali (C.SLOJ). V naslednjem koraku lahko navedemo stil pisave in polje, ki skupaj določata vsebino geometrije, izrisane na zemljevidu. Izbrati moramo stil, ki smo ga definirali za posamezen sloj (T.SLOJ).

AgriMap mora prikazovati tudi podatke posameznih slojev (informacijsko okno), česar pa se preko čarovnika ne da nastaviti. Polja moramo navesti v meniju napredno (angl. advanced). Tu lahko nastavimo stil, način predpomnenja (angl. caching) in velikost. Vsebino informacijskega okna navedemo v vrstici »Info Columns«. Določimo mu lahko opisna in vrednostna polja. Slika 20 prikazuje dokument sloja XML, ki ga generira MapBuilder.

AgriMap mora prikazovati 16 slojev:

- GERK (grafična enota rabe kmetijskega gospodarstva),
- SUPER2009 (Super sloj),
- RABA2010 (Raba kmetijskih zemljišč),
- GERK_P (GERK-i v prekrivanju),
- OMD (Območja z omejenimi možnostmi za kmetijsko dejavnost),
- GRB (Košnja grbinastih travnikov),
- GRB_08 (Košnja grbinastih travnikov 2008),
- HAB2006 (Ohranjanje posebnih traviščnih habitatov 2006),
- HAB2007 (Ohranjanje posebnih traviščnih habitatov 2007),
- MET (Ohranjanje traviščnih habitatov metuljev),
- STE (Ohranjanje steljnikov),
- VTR (Ohranjanje habitatov ptic vlažnih ekstenzivnih travnikov na območjih Natura 2000),
- VVO (Pokritost tal na vodovarstvenem območju),
- VVO_08 (Pokritost tal na vodovarstvenem območju 2008),
- ZO (Zavarovano območje),
- ZVE (Reja domačih živali v osrednjem območju pojavljanja velikih zveri). (16)



Slika 20: Definicija slojev, Oracle MapBuilder

Poleg omenjenih slojev mora AgriMap prikazovati tudi letalsko sliko. Slika se nahaja na strežniku ArcIMS Geodetske uprave Republike Slovenije. Strežnik ArcIMS omogoča prejemanje dinamičnih zemljevidov. Slike lahko pridobivamo preko protokola WMS. Konektor na strežniku ArcIMS ni omogočen. Da bi konektor omogočili, bi bil potreben poseg na njihovem strežniku. To pa je bilo žal nesprejemljivo, saj spreminjanje produkcijskega okolja, za namen predstavitvenega programa, ni mogoče.

Odločili smo se, da bomo letalsko sliko vključili kasneje, ko bo že znano, ali bo programska rešitev AgriMap sprejeta.

4.2.3 Osnovni zemljevid

Osnovni zemljevid si lahko predstavljamo kot sloj, ki se nahaja najnižje in služi za podlago vsem ostalim. Vključimo mu lahko poljubno mnogo slojev. Osnovni zemljevid bo vedno prikazan, vključen bo k sloju plošč [4.2.4], kar pomeni, da se bo tudi predpomnil posebej.

Osnovni zemljevid pri AgriMapu bi moral vsebovati sloj z letalsko karto Slovenije. Sloja za potrebe predstavitvenega programa nismo mogli definirati.

4.2.4 Sloj plošč (angl. Map Tile Layer)

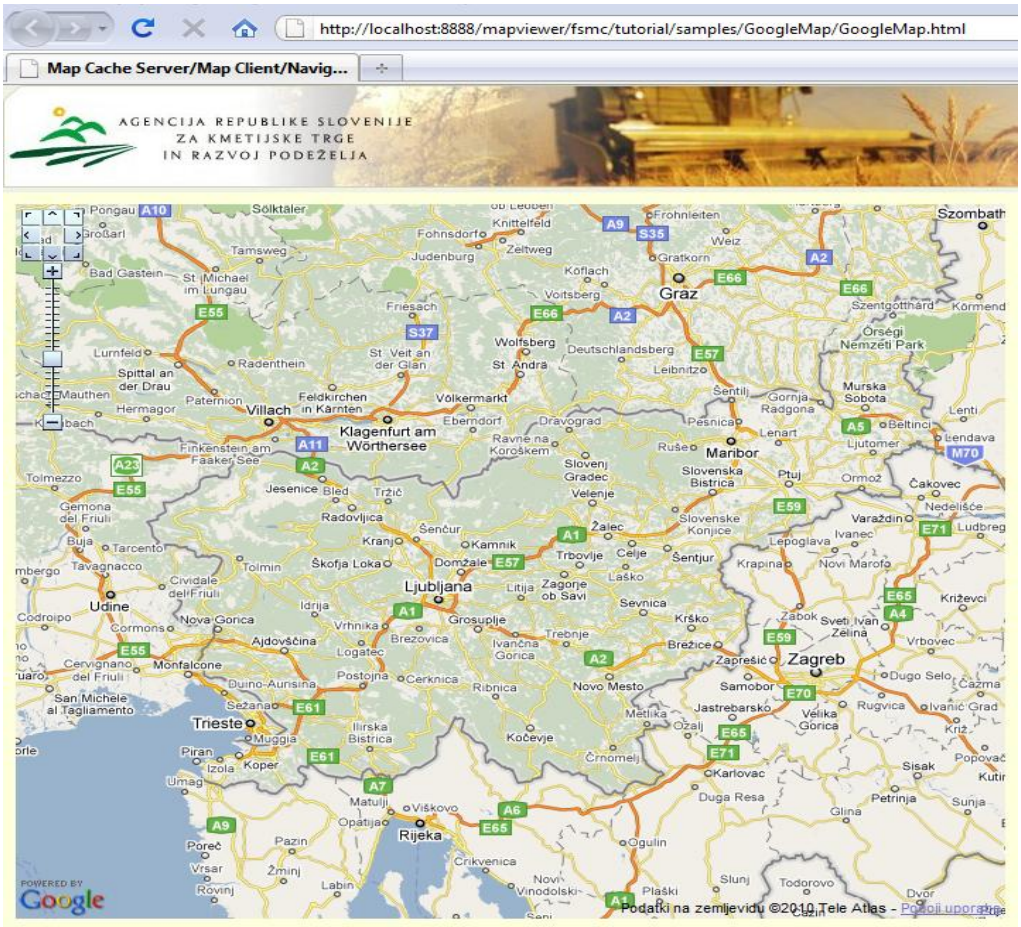
Ker letalske slike nismo mogli vključiti, smo aplikaciji AgriMap poskušali dodati vsaj eno satelitsko sliko brezplačnih virov (Google Maps, Navteq). Predvsem zanimiva in preprosta je vključitev Google Mapsa, saj ima Oracle MapViewer podporo za njegov vmesnik API. Za pridobitev Google Maps vmesnika API potrebujemo ključ, ki ga dobimo na Googlovi strani. Poleg prijave moramo vnesti še povezavo do strani, ki bo vmesnik API uporabljala. Slika 21 prikazuje Googlov sloj plošč, ki ga uporablja aplikacija MapViewer. (18)

Vmesnik API pridobimo takole:

```
<script src="http://maps.google.com/maps?file=api&v=2&key=[KLJUČ]"
        type="text/javascript">
</script>
```

Do Googlovega sloja plošč lahko sedaj dostopamo preko vmesnika API:

```
mapview = new MVMapView(document.getElementById("map"), baseURL);
tileLayer = new MVGoogleTileLayer();
mapview.addMapTileLayer(baseMap);
```



Slika 21: Googlov sloj plošč

Vključitev Googlovega sloja bi bila zelo primerna rešitev, žal pa smo tudi tukaj naleteli na težave. Ker Google Maps uporablja drugačen koordinatni sistem kot ga uporabljajo naši sloji, bi bila potrebna transformacija. Geometrijski podatki naših slojev pa nimajo definiranega koordinatnega sistema, zato Oracle Spatial transformacije ne more izvajati. Slojem bi bilo potrebno definirati koordinatni sistem (polje SRID), za kar bi bilo potrebno odstraniti prostorski indeks na tabeli, jim posodobiti vsa polja in prostorski indeks ponovno kreirati. V spodnjem primeru je predstavljen postopek določanja že definiranega koordinatnega sistema, s šifro 3333, slojem.

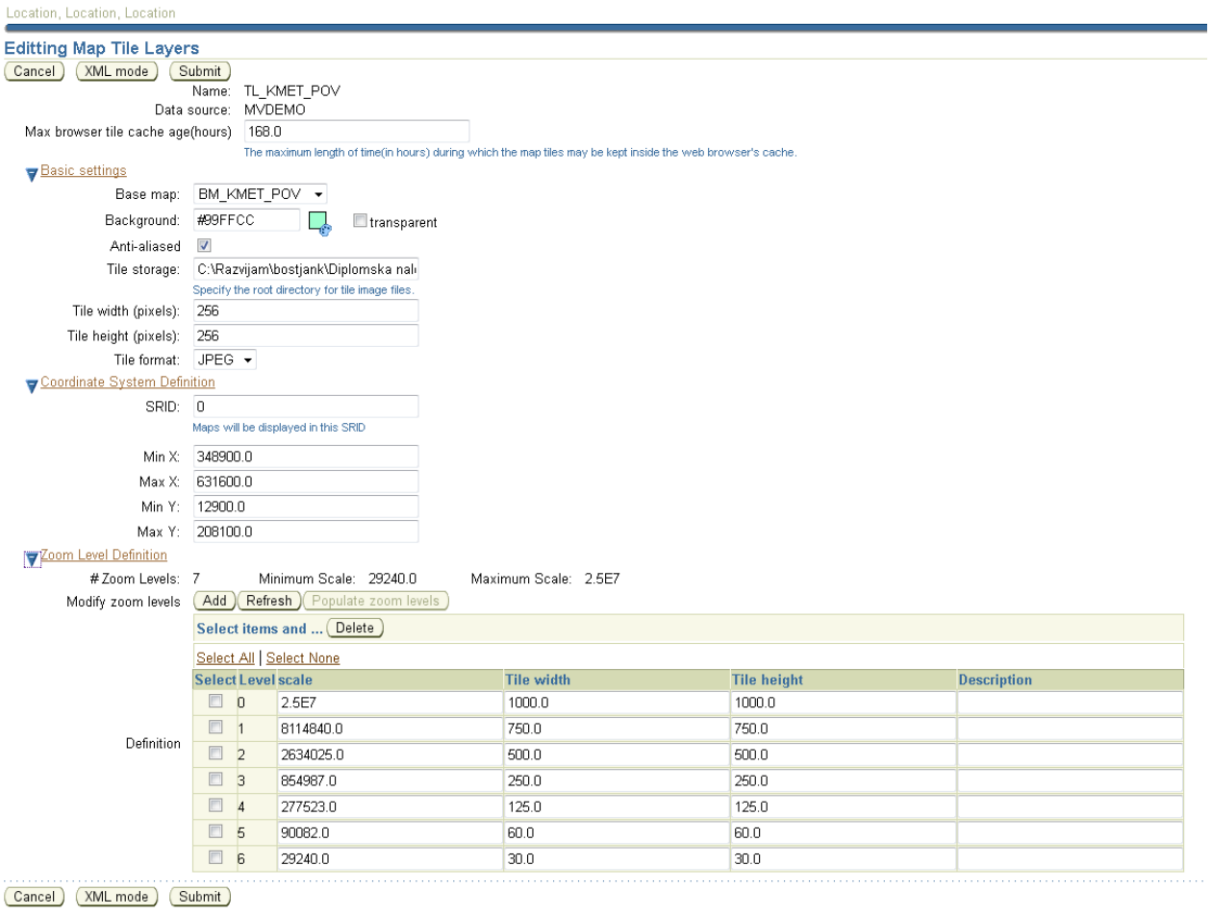
```
DROP SPATIAL INDEX ON SLOJ_1;
UPDATE SLOJ_1 S SET S.GEOMETRY.SDO_SRID = 3333;
CREATE INDEX IDX_SLOJ_1
          ON SLOJ_1(GEOMETRY) INDEXTYPE IS MDSYS.SPATIAL_INDEX;
```

Ker tudi baze ne smemo spreminjati za potrebe predstavitvenega programa, nam ni preostala druga možnost kot je ta, da na osnovnem sloju ne prikazujemo niti slik niti geometrijskih podatkov. Če bo AgriMap s strani AKTRP-ja sprejet, bomo kasneje podatke uredili (definirali koordinatni sistem). V MapBuilderju lahko prazen zemljevid naredimo tako, da mu dodamo nek neveljaven element.

```
<?xml version="1.0" standalone="yes"?>
<map_definition>
  <theme name="Geometry Themes"/>
</map_definition>
```

Sloj plošč je najlažje definirati preko spletnega vmesnika za MapViewer (Slika 22), lahko pa ga tudi preko Oracle MapBuilder orodja. Novi sloj plošč kreiramo na zavihku »Upravljanje« (angl. management). Ko mu podamo podatkovni vir, nam ponudi možnost izbire osnovnega zemljevida. Navesti moramo tudi:

- število ur hranjenja slik (plošč) v predpomnilniku brskalnika;
- lokacijo slik;
- število stopenj velikosti prikaza;
- minimalno mero (angl. minimum scale);
- maksimalno mero (angl. maximum scale);
- identifikacijsko številko koordinatnega sistema (SRID), v katerem bodo prikazani zemljevidi;
- meje koordinatnega sistema;
- velikost slik (plošč).



Slika 22: Definicija sloja plošč preko Oracle MapViewer spletnega vmesnika

4.3 Oracle Maps

Vzpostavljen MapViewer strežnik in pripravljeni metapodatki nam služijo kot osnova pri gradnji AgriMap aplikacije (Slika 23). AgriMap bomo gradili z Oracle Maps vmesnikom API. Zemljevid kreiramo z objektom razreda MVMapView. Podati mu moramo oznako html (div), znotraj katere se prikazuje, in url strežnika MapViewer.

```
mapview = new MVMapView(document.getElementById("map"),
                        "http://"+document.location.host+"/mapviewer");
```

Objektu mapview lahko sedaj dodamo sloj plošč, ki ima definiran koordinatni sistem in velikosti prikaza, nima pa dodane nobenega sloja.

```
mapview.addMapTileLayer(new MVMapTileLayer("mvdemo.TL_KMET_POV"));
```

Sloj FOI dodamo v funkciji addThemeBasedFOI. Funkcija doda objektu mapview vse sloje s seznama loArrayThemes. S funkcijo enableAutoWholeImage pospešimo delovanje. Mapviewer sam izbira, kdaj bo sliko izrisal kot celoto in kdaj bo onemogočena interakcija odjemalca s sloji. Na ta način se zmanjša število slik, ki se izrišejo, in podatkov, ki se prenašajo iz strežnika. Privzeto je prikazan sloj GERK-ov.

```
function addThemeBasedFOI() {
  for (var i in loArrayThemes) {
    var loThemebasedfoi = new MVThemeBasedFOI(i, "mvdemo." +
                                              loArrayThemes[i]);
    loThemebasedfoi.enableAutoWholeImage(true);
    if(i == 'loThemebasedfoiGerki')
      loThemebasedfoi.enableLabels(true);
    else{
      loThemebasedfoi.enableLabels(false);
      loThemebasedfoi.setVisible(false);
    }
  mapview.addThemeBasedFOI(loThemebasedfoi);
}
```

Zemljevidu smo dodali tudi kontrole za navigacijo, določanje velikosti prikaza z opisi in kontrolo, ki prikazuje mero (angl. scale). Zemljevid z vsemi kontrolami prikažemo s funkcijo mapview.display().

Naslednji korak pri razvoju je bila izdelava funkcionalnosti, kjer bodo uporabniki sami določali, kateri sloji se prikazujejo. V ta namen smo na grafičnem vmesniku AgriMap aplikacije dodali izbirna polja za vsak sloj.

```
<input type="checkbox" id="loThemebasedfoiGrb"
       value="loThemebasedfoiGrb" onclick="setVisible(this)" />
```

Prikazovanje slojev se nastavlja v funkciji:

```
function setVisible(item) {
  loThemebasedfoi = mapview.getThemeBasedFOI(item.value);
  loThemebasedfoi.setVisible(!loThemebasedfoi.isVisible());
}
```

Podatke posameznih slojev, ki se prikazujejo znotraj informacijskega okna, smo že določili z MapBuilder orodjem. Informacijsko okno se prikaže, ko kliknemo na določeno točko na zemljevidu. Prikažejo se vedno podatki tistega sloja, ki je v izbrani točki najvišje, kar v našem primeru pomeni, da je sloj na objekt mapview dodan kot zadnji. Privzeta funkcionalnost lahko postane precej moteča, ker AgriMap vključuje 16 slojev, kjer se geometrije pogosto prekrivajo. Za prikaz podatkov določenega sloja bi bilo potrebno skriti vse prekrivajoče se sloje, ki smo jih dodali kasneje. Funkcionalnost smo popravili tako, da smo uporabniku dodali še možnost izbire tistega sloja, ki se prikaže nazadnje.

```
<input type="radio" value="ioThemebasedfoiGrb"
       name="loRadioLabel" onclick="setDefaultLabel(this)" />
```

Za zaporedje prikazovanja slojev je dovolj nastavljanje indeksa sloja. Če želimo, da se določeni sloj prikazuje kot zadnji, mu moramo določiti najvišji indeks, vsem ostalim pa omenjeni indeks popraviti.

```
function resetThemeIndex(aoRadioItem){
      loThemebasedfoi = ioMapview.getThemeBasedFOI(aoRadioItem.value);
      loThemebasedfoi.enableLabels(true);
      ioMapview.setThemeIndex(loThemebasedfoi, 100); // zadnji indeks
      for (i in loArrayThemes){
        loThemebasedfoi = ioMapview.getThemeBasedFOI(i);
        if(i != aoRadioItem.value){
          loThemebasedfoi.enableLabels(false);
          ioMapview.setThemeIndex(loThemebasedfoi,
           ioMapview.getOrigThemeIndex(loThemebasedfoi));//orig. indeks
        }
      }
      // ponovno prikažemo, da pobriše labele ostalih slojev
      ioMapview.display();
    }
```

AgriMap mora podpirati tudi iskanje po zemljevidu. Parametri, ki jih pri iskanju vnesemo, so:

- lokacija (X, Y) in
- GERK pid (šifra GERK-a).

Iskanje po lokacijah smo dodali predvsem za potrebe testiranja. Iskanje se izvede z JavaScript funkcijo.

```
function setXY() {
  ioMapCenterLon=parseInt(
                    document.getElementById("ioTexFieldX").value);
  ioMapCenterLat=parseInt(
                    document.getElementById("ioTexFieldY").value);
  ioMPoint= MVSdoGeometry.createPoint(ioMapCenterLon,ioMapCenterLat,0);
  ioMapview.setCenter(ioMPoint);
}
```

Iskanje po šifri GERK-a smo realizirali z klicem Ajax.

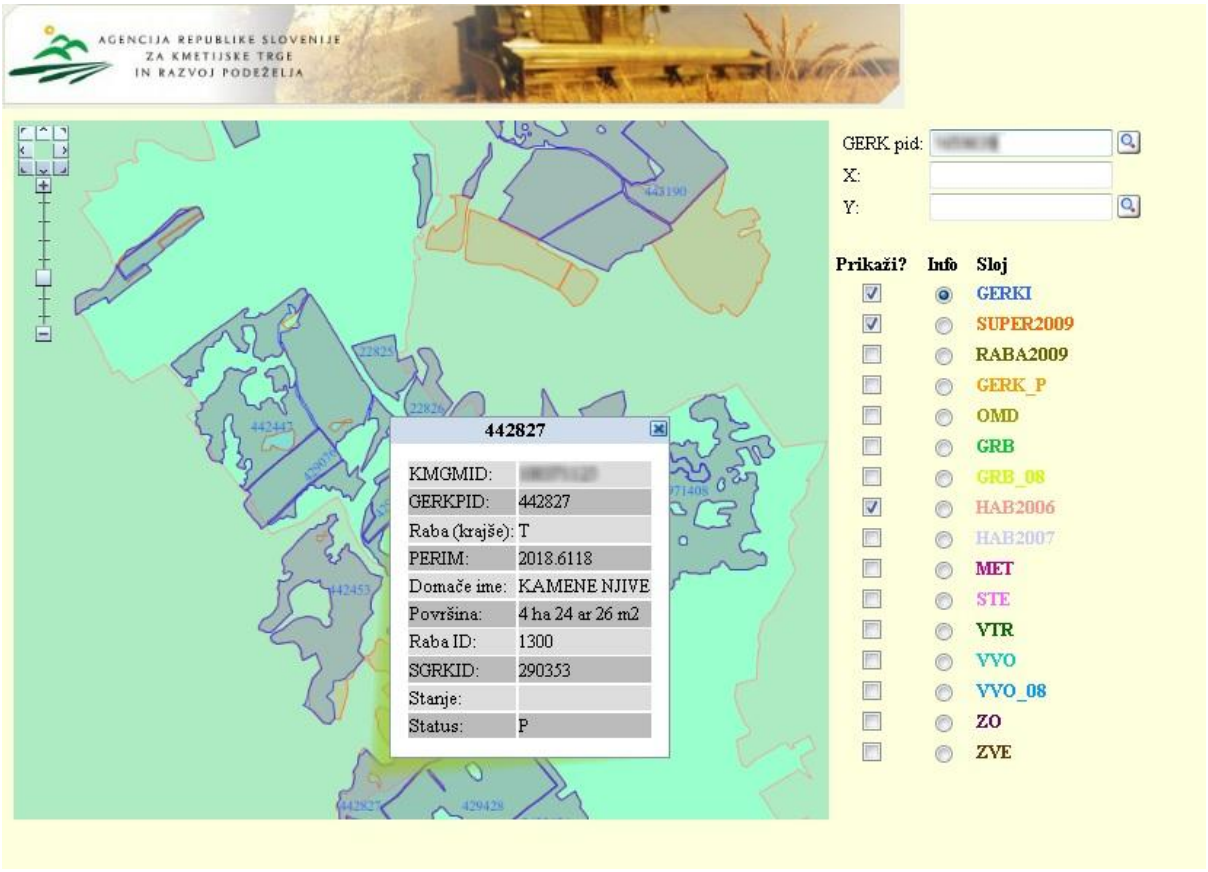
```
AJAX.open("GET", "dbGetLocation.jsp?gerk=" +
document.getElementById("ioTexFielGerkID").value , false);
AJAX.send(null);
var loXml = AJAX.responseText;
```

Stran dbGetLocation.jsp se povezuje na bazo. Poizvedba na bazi se omejuje po šifri GERK-a in kot odgovor prejme koordinati X in Y.

```
Connection loConnection = DriverManager.getConnection(lsUrl,
                                                      lsUsername,
                                                      lsPassword);
String lsSqlCall = String.format("SELECT X_C, Y_C
                                    FROM KSS2010P.SBV_GERKI_D
                                   WHERE GERK_PID = %d ", liGerkId);
Statement lsStmt = loConnection.createStatement();
ResultSet lsRSet = lsStmt.executeQuery(lsSqlCall);
lsRSet.next();
liX = lsRSet.getInt("X_C");
liY = lsRSet.getInt("Y_C");
```

Klic Ajax prejme odgovor v obliki XML. Dobljeni koordinati nastavimo za središče zemljevida AgriMap.

```
ioMapCenterLat = parseInt(
        loXmlDoc.getElementsByTagName("X")[0].firstChild.data);
ioMapCenterLon = parseInt(
loXmlDoc.getElementsByTagName("Y")[0].firstChild.data);
ioMPoint = MVSdoGeometry.createPoint(ioMapCenterLon,ioMapCenterLat,0);
ioMapview.setCenter(ioMPoint);
```



Slika 23: Aplikacija GIS – AgriMap

5 Zaključek

V diplomskem delu smo predstavili Oraclove tehnologije za razvoj GIS-ov. Izdelali smo tudi GIS, poimenovan AgriMap. AgriMap je spletni predstavitveni GIS, s katerim bomo poskušali prepričati AKTRP k postopni nadomestitvi obstoječega STE.

Za izdelavo AgriMapa je potrebno poznati Oraclovi tehnologiji Oracle Spatial in Oracle MapViewer. Delo smo si olajšali z orodjem Oracle MapBuilder, kjer smo pripravili potrebne metapodatke. AgriMap uporablja iste podatke kot jih uporablja STE. Podatki se nahajajo na bazi Oracle 11g Enterprise Edition.

AgriMap nas je s svojim delovanjem zelo presenetil, saj nismo pričakovali hitrosti in odzivnosti, ki jo dosega. Na primerjavo med sistemoma bomo morali še nekoliko počakati, saj nam še ni uspelo vključiti letalske slike Slovenije. Ker smo se odločili, da z razvojem nadaljujemo, smo MapViewer namestili na strežnik Oracle Application Server. Za pospešitev razvoja bomo poskušali AgriMap vključiti k nadgrajeni tehnologiji JSP, imenovani Application Development Framework (ADF), ki jo tudi razvija Oracle.

Če bo AgriMap s strani AKTRP sprejet, bomo morali urediti prostorske podatke. Pri njihovem urejanju bomo največ pozornosti namenili definiciji koordinatnega sistema, ki ga uporabljajo prostorski podatki. S tem bomo Oracle Spatialu omogočili izvedbo transformacije med koordinatnimi sistemi. Omogočili bomo tudi protokol WMS na strežniku ArcIMS, ki vrača letalske slike. Omenjen protokol bomo uporabili pri definiciji sloja WMS, ki ga bomo vključili k osnovnemu zemljevidu.

AgriMapu bomo implementirali tudi risanje poligonov, kot ga podpira STE. Poligone bomo risali na izbrani sloj. Končni korak pri razvoju bo vključitev poslovne logike iz STE.

Dodatek A

AgriMap.html

```
<html>
<head>
    <meta http-equiv="Content-Type" content="text/html;
          charset=windows-1250">
    </meta>
    <title>AgriMap</title>
    <script type="text/javascript" language="Javascript"
            src="/mapviewer/fsmc/jslib/oraclemaps.js"></script>
    <script type="text/javascript" language=javascript>
      var ioMapview;
      var ioMapCenterLon =  491528;
      var ioMapCenterLat =  111768;
      var ioMPoint;
      var loArrayThemes = { 'ioThemebasedfoiGerki' : 'T_GERK_D',
                            'ioThemebasedfoiSuper09' : 'T_SUPER2009',
                            'ioThemebasedfoiRaba2009' : 'T_RABA2009',
                            'ioThemebasedfoiGerkiP' : 'T_GERK_P',
                            'ioThemebasedfoiOmd' : 'T_OMD',
                            'ioThemebasedfoiGrb' : 'T_GRB',
                            'ioThemebasedfoiGrb08' : 'T_GRB_08',
                            'ioThemebasedfoiHab06' : 'T_HAB2006',
                            'ioThemebasedfoiHab07' : 'T_HAB2007',
                            'ioThemebasedfoiMet' : 'T_MET',
                            'ioThemebasedfoiSte' : 'T_STE',
                            'ioThemebasedfoiVtr' : 'T_VTR',
                            'ioThemebasedfoiVvo' : 'T_VVO',
                            'ioThemebasedfoiVvo08' : 'T_VVO_08',
                            'ioThemebasedfoiZo' : 'T_ZO',
                            'ioThemebasedfoiZve' : 'T_ZVE' };

      // AJAX odgovor s strežnika (lokacija)
      function sinhronAjaxSetLocation(){
        var loRet = "";

        if (window.XMLHttpRequest) {
          AJAX = new XMLHttpRequest();
        } //IE
        else {
          AJAX = new ActiveXObject("Microsoft.XMLHTTP");
        }

        if (AJAX) {
          AJAX.open("GET", " DBGetLocation.jsp?gerk=" +
              document.getElementById("ioTexFielGerkID").value , false);
          AJAX.send(null);
          var loXml = AJAX.responseText;

          // code for IE
          if (window.ActiveXObject) {
              var loDoc = new ActiveXObject("Microsoft.XMLDOM");
              loDoc.async= " false";
              loDoc.loadXML(loXml);
          }
```

```
    // code for Mozilla, Firefox, Opera, etc.
    else {
        var parser = new DOMParser();
        var loDoc = parser.parseFromString(loXml, "text/xml");
    }

    var loXmlDoc = loDoc.documentElement;
    ioMapCenterLat = parseInt(loXmlDoc.getElementsByTagName("X")
                        [0].firstChild.data);
    ioMapCenterLon = parseInt(loXmlDoc.getElementsByTagName("Y")
                        [0].firstChild.data);

    if (AJAX.status != 200) {
      alert("Ajax napaka. Status klicane strani je različen od
            200.");
      return false;
    }
  }
  else {
    loRet = "Povezava na bazo ni uspela.";
  }
  setLocationShowMap(); // nastavimo lokacijo v mapviewer
  return loRet;
}

function showMap() {
  var lsBaseURL  = "http://" + document.location.host + "/mapviewer";
  var liMapZoom  = 3;
  ioMPoint = MVSdoGeometry.createPoint(
                  ioMapCenterLon,ioMapCenterLat,0);
  ioMapview = new MVMapView(document.getElementById("map"),
                              lsBaseURL);
  ioMapview.addMapTileLayer(new
             MVMapTileLayer("mvdemo.TL_KMET_POV"));
  ioMapview.setCenter(ioMPoint);
  ioMapview.setZoomLevel(liMapZoom);
  var loNav = new MVNavigationPanel() ;
  loNav.setZoomLevelInfoTips({0:"Dalec", 3:"Priblizano", 6:"Blizu"});
  loNavPan = new MVMapDecoration(loNav, 0, 0, null, null, 4, 4) ;

  // dodamo vse sloje, privzeto so gerki na vrhu
  addThemeBasedFOI('ioThemebasedfoiGerki');

  ioMapview.addMapDecoration(loNavPan) ;
  ioMapview.display();
}

function addThemeBasedFOI(asDefaultTheme) {
  for (var i in loArrayThemes) {
    var loThemebasedfoi = new MVThemeBasedFOI(i, "mvdemo." +
                                          loArrayThemes[i]);
    loThemebasedfoi.enableAutoWholeImage(true);
    if(i == asDefaultTheme)
      loThemebasedfoi.enableLabels(true);
    else{
      loThemebasedfoi.enableLabels(false);
      loThemebasedfoi.setVisible(false);
    }
    ioMapview.addThemeBasedFOI(loThemebasedfoi);
  }
}
```

```
function setVisible(aoCheckBoxItem) {
  loThemebasedfoi = ioMapview.getThemeBasedFOI(aoCheckBoxItem.value);

  loThemebasedfoi.setVisible(!loThemebasedfoi.isVisible());

  for( i = 0; i < document.ioForm.loRadioLabel.length; i++ ) {
    if( document.ioForm.loRadioLabel[i].checked == true ){
      // določimo sloj, ki se prikaže na vrhu (poligon, labela, info)
      resetThemeIndex(document.ioForm.loRadioLabel[i]);
      break;
    }
  }
}

function setLocationShowMap() {
  ioMPoint = MVSdoGeometry.createPoint(ioMapCenterLon,
                                       ioMapCenterLat,0);
  ioMapview.setCenter(ioMPoint);
}

function setXY() {
  ioMapCenterLon =  parseInt(document.getElementById("ioTexFieldX")
                        .value);
  ioMapCenterLat =  parseInt(document.getElementById("ioTexFieldY")
                        .value);
  setLocationShowMap();
}

function setDefaultLabel(aoRadioItem) {
  if(aoRadioItem.checked){
    // preverimo, če je sloj nastavljen kot visible
    loCheckBoxTheme = document.getElementById(
                        aoRadioItem.value.replace("ioThemebasedfoi",
                                                  "ioCheckbox"));
    if(!loCheckBoxTheme.checked){ // nastavimo sloj in CB na visible
      loCheckBoxTheme.checked = true;
      setVisible(loCheckBoxTheme);
      return;
    }
    else
      resetThemeIndex(aoRadioItem); // samo ponovno nastavimo indeks
  }
}

// nastavimo indeks sloja, da se prikaže pravi info okno
function resetThemeIndex(aoRadioItem){
  loThemebasedfoi = ioMapview.getThemeBasedFOI(aoRadioItem.value);
  loThemebasedfoi.enableLabels(true);
  ioMapview.setThemeIndex(loThemebasedfoi, 100); // zadnji indeks

  for (i in loArrayThemes){
    loThemebasedfoi = ioMapview.getThemeBasedFOI(i);
    if(i != aoRadioItem.value){
      loThemebasedfoi.enableLabels(false);
      ioMapview.setThemeIndex(loThemebasedfoi,
       ioMapview.getOrigThemeIndex(loThemebasedfoi));          }
  }


  // ponovno prikažemo, da pobriše labele ostalih slojev
  ioMapview.display();
```

```
      }
    </script>

</head>

<body onload="showMap();" bgcolor="#FEFFDD">

  <div id="logo" style="position:absolute; left:0px; top:0px; height:50px">
    <img src="logo_arsktrp.jpg" alt="" />
  </div>

  <div id="map" style="position:absolute; left:10px; top:100px;
width:700px; height:600px"></div>

  <div style="position:absolute;top:210px; left:710px;" class="noprint">
    <form name="ioForm">
     <table>
      <tr>
        <td width="68" align="center">
          <b>Prikaži?</b>
        </td>
        <td width="47" align="center">
          <b>Info</b>
        </td>
        <td width="105">
          <b>Sloj</b>
        <td width="4">
      </tr>

      <tr>
        <td align="center" width="68">
          <input type="checkbox" id="ioCheckboxGerki"
                 value="ioThemebasedfoiGerki"
                 onclick="setVisible(this)" checked />
        </td>
        <td align="center" width="47">
          <input type="radio" value="ioThemebasedfoiGerki"
                 name="loRadioLabel" checked
                 onclick="setDefaultLabel(this)" />
        </td>
        <td width="105">
          <b style="color: #3366FF">GERKI</b>
        <td width="4">
      </tr>

      <tr>
        <td align="center" width="68">
          <input type="checkbox" id="ioCheckboxSuper09"
                 value="ioThemebasedfoiSuper09"
                 onclick="setVisible(this)" />
        </td>
        <td align="center" width="47">
          <input type="radio" value="ioThemebasedfoiSuper09"
                 name="loRadioLabel" onclick="setDefaultLabel(this)" />
        </td>
        <td width="105">
          <b style="color: #FF6600">SUPER2009</b>
        <td width="4">
      </tr>
      <tr>
        <td align="center" width="68">
```

```
    <input type="checkbox" id="ioCheckboxRaba2009"
           value="ioThemebasedfoiRaba2009"
           onclick="setVisible(this);" />
  </td>
  <td align="center" width="47">
    <input type="radio" value="ioThemebasedfoiRaba2009"
           name="loRadioLabel" onclick="setDefaultLabel(this);" />
  </td>
  <td width="105">
    <b style="color: #666600">RABA2009</b>
  <td width="4">
</tr>

<tr>
  <td align="center" width="68">
    <input type="checkbox" id="ioCheckboxGerkiP"
           value="ioThemebasedfoiGerkiP"
           onclick="setVisible(this)" />
  </td>
  <td align="center" width="47">
    <input type="radio" value="ioThemebasedfoiGerkiP"
           name="loRadioLabel" onclick="setDefaultLabel(this)" />
  </td>
  <td width="105">
    <b style="color: #FF9900">GERK_P</b>
  <td width="4">
</tr>

<tr>
  <td align="center" width="68">
    <input type="checkbox" id="ioCheckboxOmd"
           value="ioThemebasedfoiOmd" onclick="setVisible(this)" />
  </td>
  <td align="center" width="47">
    <input type="radio" value="ioThemebasedfoiOmd"
           name="loRadioLabel" onclick="setDefaultLabel(this)" />
  </td>
  <td width="105">
    <b style="color: #999900">OMD</b>
  <td width="4">
</tr>

<tr>
  <td align="center" width="68">
    <input type="checkbox" id="ioCheckboxGrb"
           value="ioThemebasedfoiGrb" onclick="setVisible(this)" />
  </td>
  <td align="center" width="47">
    <input type="radio" value="ioThemebasedfoiGrb"
           name="loRadioLabel" onclick="setDefaultLabel(this)" />
  </td>
  <td width="105">
    <b style="color: #00CC33">GRB</b>
  <td width="4">
</tr>

<tr>
  <td align="center" width="68">
    <input type="checkbox" id="ioCheckboxGrb08"
           value="ioThemebasedfoiGrb08" onclick="setVisible(this)" />
  </td>
```

```
  <td align="center" width="47">
    <input type="radio" value="ioThemebasedfoiGrb08"
           name="loRadioLabel" onclick="setDefaultLabel(this)" />
  </td>
  <td width="105">
    <b style="color: #CCFF00">GRB_08</b>
  <td width="4">
</tr>

<tr>
  <td align="center" width="68">
    <input type="checkbox" id="ioCheckboxHab06"
           value="ioThemebasedfoiHab06" onclick="setVisible(this)" />
  </td>
  <td align="center" width="47">
    <input type="radio" value="ioThemebasedfoiHab06"
           name="loRadioLabel" onclick="setDefaultLabel(this)" />
  </td>
  <td width="105">
    <b style="color: #FF9999">HAB2006</b>
  <td width="4">
</tr>

<tr>
  <td align="center" width="68">
  <input type="checkbox" id="ioCheckboxHab07"
         value="ioThemebasedfoiHab07" onclick="setVisible(this)" />
  </td>
  <td align="center" width="47">
    <input type="radio" value="ioThemebasedfoiHab07"
           name="loRadioLabel" onclick="setDefaultLabel(this)" />
  </td>
  <td width="105">
    <b style="color: #CCCCFF">HAB2007</b>
  <td width="4">
</tr>

<tr>
  <td align="center" width="68">
    <input type="checkbox" id="ioCheckboxMet"
           value="ioThemebasedfoiMet" onclick="setVisible(this)" />
  </td>
  <td align="center" width="47">
    <input type="radio" value="ioThemebasedfoiMet"
           name="loRadioLabel" onclick="setDefaultLabel(this)" />
  </td>
  <td width="105">
    <b style="color: #CC0099">MET</b>
  <td width="4">
</tr>

<tr>
  <td align="center" width="68">
    <input type="checkbox" id="ioCheckboxSte"
           value="ioThemebasedfoiSte" onclick="setVisible(this)" />
  </td>
  <td align="center" width="47">
    <input type="radio" value="ioThemebasedfoiSte"
           name="loRadioLabel" onclick="setDefaultLabel(this)" />
  </td>
  <td width="105">
```

```
    <b style="color: #FF66FF">STE</b>
  <td width="4">
</tr>

<tr>
  <td align="center" width="68">
    <input type="checkbox" id="ioCheckboxVtr"
           value="ioThemebasedfoiVtr" onclick="setVisible(this)" />
  </td>
  <td align="center" width="47">
    <input type="radio" value="ioThemebasedfoiVtr"
           name="loRadioLabel" onclick="setDefaultLabel(this)" />
  </td>
  <td width="105">
    <b style="color: #006600">VTR</b>
  <td width="4">
</tr>

<tr>
  <td align="center" width="68">
    <input type="checkbox" id="ioCheckboxVvo"
           value="ioThemebasedfoiVvo" onclick="setVisible(this)" />
  </td>
  <td align="center" width="47">
    <input type="radio" value="ioThemebasedfoiVvo"
           name="loRadioLabel" onclick="setDefaultLabel(this)" />
  </td>
  <td width="105">
    <b style="color: #00CCCC">VVO</b>
  <td width="4">
</tr>

<tr>
  <td align="center" width="68">
    <input type="checkbox" id="ioCheckboxVvo08"
           value="ioThemebasedfoiVvo08" onclick="setVisible(this)" />
  </td>
  <td align="center" width="47">
    <input type="radio" value="ioThemebasedfoiVvo08"
           name="loRadioLabel" onclick="setDefaultLabel(this)" />
  </td>
  <td width="105">
    <b style="color: #0099FF">VVO_08</b>
  <td width="4">
</tr>

<tr>
  <td align="center" width="68">
    <input type="checkbox" id="ioCheckboxZo"
           value="ioThemebasedfoiZo" onclick="setVisible(this)" />
  </td>
  <td align="center" width="47">
    <input type="radio" value="ioThemebasedfoiZo"
           name="loRadioLabel" onclick="setDefaultLabel(this)" />
  </td>
  <td width="105">
    <b style="color: #660066">ZO</b>
  <td width="4">
</tr>

<tr>
```

```
        <td align="center" width="68">
          <input type="checkbox" id="ioCheckboxZve"
                 value="ioThemebasedfoiZve" onclick="setVisible(this)" />
        </td>
        <td align="center" width="47">
          <input type="radio" value="ioThemebasedfoiZve"
                 name="loRadioLabel" onclick="setDefaultLabel(this)" />
        </td>
        <td width="105">
          <b style="color: #663300">ZVE</b>
        <td width="4">
      </tr>

     </table>
    </form>
  </div>
  <div style="position:absolute;top:80px; left:720px;" class="noprint">
    <table >
      <tr>
        <td colspan="3"> 
        </td>
      </tr>
      <tr>
        <td>GERK pid:</td>
        <td>
          <input type="text" name="ioTexFielGerkID" id="ioTexFielGerkID" />
        </td>
        <td>
          <input type="image" src="search.png" name="OK" value="Isci"
                 title="Išči" onClick="sinhronAjaxSetLocation();" />
        </td>
      </tr>
      <tr>
        <td>X: </td>
        <td>
          <input type="text" name="ioTexFieldX" id="ioTexFieldX" />
        </td>
        <td> </td>
      </tr>
      <tr>
        <td>Y: </td>
        <td>
          <input type="text" name="ioTexFieldY" id="ioTexFieldY" />
        </td>
        <td>
          <input type="image" src="search.png" name="OK" value="Isci XY"
                 title="Išči XY" onClick="setXY();" />
        </td>
      </tr>
    </table>
  </div>
</body>
</html>
```

Dodatek B

DBGetLocation.jsp

```
<%@ page
  language="java"
  import="java.sql.*, javax.naming.*,
  javax.sql.DataSource,java.util.regex.*"
  contentType="text/html;charset=windows-1250"
%>
<%
  response.setHeader("Cache-Control","no-cache"); //HTTP 1.1
  response.setHeader("Pragma","no-cache"); //HTTP 1.0
  response.setDateHeader ("Expires", 0);

  int liGerkId = -1;
  int liX;
  int liY;
  Connection loConnection = null;

  try{
    liGerkId = Integer.parseInt(request.getParameter("gerk"));
  }
  catch(Exception e){ return;}

  try {
    // JDBC gonilnik
    String driverName = "oracle.jdbc.driver.OracleDriver";
    Class.forName(driverName);

    // Povezava na bazo
    String lsUrl = "jdbc:oracle:thin:@//server:1521/baza.domena.si";
    String lsUsername = "AgriMap";
    String lsPassword = "Pa$$w0rd";
    loConnection = DriverManager.getConnection(lsUrl,
                                               lsUsername,
                                               lsPassword);
  } catch (ClassNotFoundException e) {
        // ne more naložiti baznega gonilnika
  } catch (SQLException e) {
        // povezava na bazo ni uspela
  }
  String lsSqlCall = String.format("SELECT X_C,
                                           Y_C
                                      FROM SHEMA.GERKI
                                     WHERE GERK_PID = %d ", liGerkId);
  Statement lsStmt = loConnection.createStatement();
  ResultSet lsRSet = lsStmt.executeQuery(lsSqlCall);
  lsRSet.next();
  liX = lsRSet.getInt("X_C");
  liY = lsRSet.getInt("Y_C");
  String lsResult = "<mapCoordinate>";
  lsResult += "<X>" + liX + "</X>";
  lsResult += "<Y>" + liY + "</Y>";
  lsResult += "</mapCoordinate>";

  if(lsResult.length() > 0)
    out.println(lsResult);
%>
```

Slike

Tabele

Viri in literatura

(1) (2010) What is GIS? Dostopno na:
http://www.gis.com/content/what-gis

(2) (2010) Geographic information system. Dostopno na:
http://en.wikipedia.org/wiki/Geographic_information_system

(3) (2010) GNSS. Dostopno na:
http://www.gu-signal.si/index.php

(4) (2010) Monroe County. Dostopno na:
http://www.monroecounty.gov/gis-What_is_GIS.php#Mapping

(5) (2010) Navteq. Dostopno na:
http://www.navteq.com

(6) (2010) Loja GPS. Dostopno na:
http://www.lojagps.com/catalogo

(7) (2010) Oracle Spatial Documentation. Dostopno na:
http://download.oracle.com/docs/cd/E11882_01/appdev.112/e11830.pdf

(8) (2010) Kartezični koordinatni sistem. Dostopno na:
http://sl.wikipedia.org/wiki/Kartezi%C4%8Dni_koordinatni_sistem

(9) (2010) Geoid. Dostopno na:
http://sl.wikipedia.org/wiki/Geoid

(10) (2010) Geograficamente. Dostopno na:
http://geograficamente.files.wordpress.com/2009/03/c71_geoid_smooth4.jpg

(11) (2010) Matematika v navtiki. Dostopno na:
http://www.fpp.edu/~dhostnikar/Matematika%20v%20navtiki/Skripta%201.%20del.pdf

(12) (2010) Kuhar, M. Koordinatni sistemi. Dostopno na:
http://www.fgg.uni-lj.si/~/mkuhar/Pouk/Geod/gradivo/Koordinatni_sistemi.pdf

(13) (2010) Reference ellipsoid. Dostopno na:
http://en.wikipedia.org/wiki/Reference_ellipsoid

(14) (2010) Državni koordinatni sistem. Dostopno na:
http://e-prostor.gov.si/index.php?id=104

(15) (2010) Oracle Fusion Middleware MapViewer Documentation. Dostopno na:
http://www.oracle.com/technology/products/mapviewer/pdf/mapviewer_10133_ug.pdf

(16) (2010) GIS aplikacija - Zbirna vloga. Dostopno na: http://www.arsktrp.gov.si/fileadmin/arsktrp.gov.si/pageuploads/Obrazci/2008/Zbirna_vloga/Ponujanje_ukrepov.doc

(18) (2010) Google Maps vmesnik API. Dostopno na: http://code.google.com/intl/sl/apis/maps/signup.html