

Univerza
v Ljubljani
Fakulteta
*za gradbeništvo
in geodezijo*

*Janova 2
1000 Ljubljana, Slovenija
telefon (01) 47 68 500
faks (01) 42 50 681
fgg@fgg.uni-lj.si*



Univerzitetni program Geodezija,
smer Geodezija

Kandidat:
Matjaž Škrinjar

Vzpostavitev in oblikovanje spletne navigacijske karte Evrope

Diplomska naloga št.: 758

Mentor:
doc. dr. Dušan Petrovič

Ljubljana, 4. 7. 2008

IZJAVA O AVTORSTVU

Podpisani **MATJAŽ ŠKRINJAR** izjavljam, da sem avtor diplomske naloge z naslovom:
»VZPOSTAVITEV IN OBLIKOVANJE SPLETNE NAVIGACIJSKE KARTE
EVROPE«.

Izjavljam, da prenašam vse materialne avtorske pravice v zvezi z diplomsko nalogo na UL,
Fakulteto za gradbeništvo in geodezijo.

Ljubljana, 20.6.2008

BIBLIOGRAFSKO – DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK: 004.738.5:528.9:912(043.2)
Avtor: Matjaž Škrinjar
Mentor: doc. dr. Dušan Petrovič
Naslov: Vzpostavitev in oblikovanje spletne navigacijske karte Evrope
Obseg in oprema: 93 str., 23 pregl., 45 sl., 4 pril.
Ključne besede: spletna karta, odprtokodna programska oprema, Open GIS

Izveček

Spletna kartografija v zadnjih letih doživlja velik napredek. Pri tem igrajo pomembno vlogo standardi konzorcija OGC ter med drugim tudi razmah odprtokodne programske opreme na tem področju. V diplomski nalogi opisujemo postopke vzpostavitve in oblikovanja spletne navigacijske karte Evrope na podlagi odprtokodne programske opreme ter standardov konzorcija OGC. Med drugim smo opisali OGC specifikacijo WMS, ki predpisuje obliko standardnega vmesnika za izmenjavo slikovnih kart preko spleta ter specifikacijo SLD, ki definira jezik za izdelavo uporabniških stilov pri oblikovanju kart. Prav tako smo opisali odprtokodno programsko opremo, ki jo potrebujemo za vzpostavitev spletne karte. Spletno karto Evrope smo izdelali na podlagi podatkov prostorske podatkovne baze *Navstreets*, ki jo zagotavlja *Navteq* – eden največjih dobaviteljev prostorskih podatkov za cestnavigacijske sisteme na svetu. Opisali smo predvsem tiste sestavine baze *Navstreets*, ki so pomembne za kartografsko predstavitev. Spletna karta Evrope bo v končni fazi omogočala poizvedbo po lokaciji ter izračun najkrajših/najhitrejših poti med različnimi kraji v Evropi. Karta pa bo dostopna na spletnem portalu www.yellowlbs.com.

BIBLIOGRAPHIC – DOCUMENTALIST INFORMATION

UDC: 004.738.5:528.9:912(043.2)
Author: Matjaž Škrinjar
Supervisor: Assist. prof. dr. Dušan Petrovič
Title: Navigational web map of Europe – design and implementation
Notes: 93 p., 23 tab., 45 fig., 4 ann.
Key words: web map, open source software, Open GIS

Abstract

Web cartography is going through a rapid development in a last few years. OGC standards have an important role during that development, along with the open source web cartography software. This graduation thesis presents procedures for implementing and designing a navigational web map of Europe, using open source software and OGC standards. OGC WMS and SLD specifications are described. Open source software required for web map implementation is also described. Web map of Europe is made of using the Navstreets spatial database. This is a Navteq product, being one of the largest suppliers of spatial data for vehicle navigation systems. All Navstreets database components that are important for cartographic visualization are presented. When made, navigational web map of Europe will offer services like location finding and calculating a fastest/shortest route between places. A map will be accessible on web portal www.yellowlbs.com

KAZALO VSEBINE

1 UVOD	1
2 ODPRTI STANDARDI.....	3
2.1 Uvod	3
2.2 »Open GIS« standardi – uvod	4
2.3 Standardi in specifikacije konzorcija OGC	5
3 ODPRTOKODNA PROGRAMSKA OPREMA	10
3.1 Uvod	10
3.2 Lastnosti odprtokodne programske opreme	10
3.3 Uspešni odprtokodni projekti	12
3.4 Licenciranje odprtokodne programske opreme	13
3.5 Odprtokodna programska oprema za spletno kartografijo	14
4 GIS (GEOGRAFSKI INFORMACIJSKI SISTEM)	17
4.1 Opredelitev geografskega informacijskega sistema	17
4.2 Prostorski podatki	18
4.3 Prostorska podatkovna baza kot model stvarnosti	18
4.4 Kartografski podatkovni model	19
5 KARTOGRAFIJA IN SPLET.....	21
5.1 Internet	21
5.2 Internetni protokoli	21
5.3 Svetovni splet	22
5.4 Internet in kartografija	22
5.5 Oblikovanje spletne karte	26
5.6 Tipi aplikacij za spletno kartografijo	30
5.7 Uporabniški vmesnik spletne karte	31
6 »STYLED LAYER DESCRIPTOR« (SLD).....	34
6.1 Uvod	34
6.2 Zgradba SLD datoteke	34
6.3 Filter	36
6.4 Simbolizatorji (»Symbolizers«)	37

7 PREDSTAVITEV PROSTORSKE PODATKOVNE BAZE »NAVSTREETS« ...	50
7.1 Uvod	50
7.2 Opredelitev območja, ki ga zajema baza »Navstreets« – Evropa	51
7.3 Homogenost baze in ažuriranje podatkov v bazi	52
7.4 Specifikacija podatkovnih slojev baze »Navstreets«	53
8 VZPOSTAVITEV SPLETNE KARTE	72
8.1 Opis odprtokodnih aplikacij, ki so potrebne za vzpostavitev spletne karte	72
8.2 Programska arhitektura sistema za strežbo kart	74
8.3 Priprava podatkov, oblikovanje in vzpostavitev spletne karte Evrope	75
8.4 Uvoz »shape« datotek v bazo »PostgreSQL/PostGIS« in indeksiranje podatkov	78
8.5 Nastavitve na »Geoserverju«	80
8.6 Opredelitev matematičnih elementov karte	82
8.7 Izdelava spletne strani	84
9 ZAKLJUČEK	88
VIRI	90

KAZALO TABEL:

Tabela 1a: Opis posameznih parametrov zahteve »GetMap«.	8
Tabela 1: Primeri uporabe elementa <Stroke>.	38
Tabela 2: Primeri uporabe elementa <Graphic>.	41
Tabela 3: Primeri uporabe elementa <Fill>.	43
Tabela 4: Primeri uporabe elementa .	46
Tabela 5: Primeri uporabe elementa <Halo>.	47
Tabela 6: Primeri uporabe elementa <LinePlacement>.	47
Tabela 7: Primeri uporabe elementa <PointPlacement>.	48
Tabela 8: Seznam podatkovnih slojev baze »Navstreets«.	50
Tabela 9: Opis atributne tabele podatkovnega sloja »Streets«.	54
Tabela 10: Opis atributne tabele podatkovnih slojev »MajHwys« in »SecHwys«.	57
Tabela 11: Opis atributne tabele podatkovnega sloja »Railrds«.	58
Tabela 12: Opis atributne tabele podatkovnega sloja »Landmark«.	59
Tabela 13: Opis atributne tabele podatkovnih slojev »AdminBndy«.	60
Tabela 14: Opis atributne tabele podatkovnega sloja »CartoCountry«.	61
Tabela 15: Opis atributne tabele podatkovnih slojev »LandUseA« in »LandUseB«.	62
Tabela 16: Opis atributne tabele podatkovnega sloja »Oceans«.	63
Tabela 17: Opis atributne tabele podatkovnega sloja »WaterPoly«.	64
Tabela 18: Opis atributne tabele podatkovnega sloja »WaterSeg«.	65
Tabela 19: Opis atributne tabele podatkovnega sloja »Islands«.	66
Tabela 20: Opis atributne tabele podatkovnega sloja »NamedPlc«.	67
Tabela 21: Opis atributne tabele podatkovnih slojev »POI«.	69
Tabela 22: Parametri generalizacije za posamezne podatkovne sloje.	76

KAZALO SLIK:

Slika 1: Enostavna izmenjava prostorskih podatkov na osnovi tehnologije »OpenGIS«.	5
Slika 2: »Suse« – priljubljena distribucija »Linuxa« za namizne uporabnike.	13
Slika 3: Zgodovina prenosov odprtokodne aplikacije »Geoserver« v obdobju 2001-2008.	14
Slika 4: »Geoserver« – priljubljeni odprtokodni kartografski strežnik.	15
Slika 5: Glavne sestavine računalniške strojne in programske opreme v GIS.	17
Slika 6: Kartografski podatkovni model in osnovni tipi grafičnih objektov.	19
Slika 7: Komunikacija odjemalec/strežnik.	24
Slika 8: Komunikacija odjemalec/strežnik: strežniški pristop.	25
Slika 9: Komunikacija odjemalec/strežnik: odjemalski pristop.	26
Slika 10: Grafične spremenljivke.	29
Slika 11: Postopki kartografske generalizacije.	30
Slika 12: Nekaj tipičnih sestavin spletne karte na primeru »Yahoo Maps«.	32
Slika 13: Prikaz zgradbe preprostega SLD-ja.	35
Slika 14: »Styled Layer Descriptor« - primer uporabe filtra.	36
Slika 15: Struktura zapisa linijskega simbolizatorja.	37
Slika 16: Struktura zapisa točkovnega simbolizatorja.	40
Slika 17: Struktura zapisa poligonskega simbolizatorja.	43
Slika 18: Struktura zapisa tekstovnega simbolizatorja.	45
Slika 18a: Homogenost podatkov v bazi »Navstreets«.	53
Slika 19: Podatkovni sloj »Streets« (grafika).	54
Slika 20: Podatkovna sloja »MajHwys« in »SecHwys« (grafika).	56
Slika 21: Podatkovni sloj »Railrds« (grafika).	57
Slika 22: Podatkovni sloj »Landmark« (grafika).	58
Slika 23: Podatkovni sloji »AdminBndy1« – »AdminBndy5« (grafika).	59
Slika 24: Podatkovni sloj »CartoCountry« (grafika).	60
Slika 25: Podatkovna sloja »LandUseA« in »LandUseB« (grafika).	61
Slika 26: Podatkovni sloj »Oceans« (grafika).	63
Slika 27: Podatkovni sloj »WaterPoly« (grafika).	64
Slika 28: Podatkovni sloj »WaterSeg« (grafika).	65
Slika 29: Podatkovni sloj »Islands« (grafika).	66

Slika 30: Podatkovni sloj »NamedPlc« (grafika).	67
Slika 31: Podatkovni sloji »POI« (grafika).	68
Slika 32: Shema programske arhitekture sistema za strežbo kart.	74
Slika 33: Generalizacija geometrije v programskem okolju »ArcMap«.	76
Slika 34: Oblikovanje točkovnih kartografskih znakov v programskem okolju »Gimp« ..	78
Slika 35: PgAdminIII - administracijski program baze Postgres.	79
Slika 36: Definiranje podatkovnega skladišča na »Geoserverju«, za primer Slovenije.....	80
Slika 37: Nastavitve v »Geoserverjevem« oknu »FeatureTypeEditor«.	81
Slika 38: Definiranje podatkovnih skupin na »Geoserverju«.	82
Slika 39: Spletna karta Evrope: primer 1.....	85
Slika 40: Spletna karta Evrope: primer 2.....	85
Slika 41: Spletna karta Evrope: primer 3.....	86
Slika 43: Spletna karta Evrope: primer 4.....	86
Slika 44: Spletna karta Evrope: primer 5.....	87

1 UVOD

Internetna tehnologija je že nekaj časa prisotna na domala vseh področjih človekovega delovanja. Z internetom se srečujemo praktično na vsakem koraku – uporabljamo ga v šoli, v službi, doma oz. kjerkoli imamo dostop do ustreznega stacionarnega ali mobilnega omrežja. Elektronsko pošto ter splet lahko pregledujemo, medtem ko čakamo na avtobus oz. ko se vozimo v službo. V poslovnem svetu in tudi drugje se je uveljavil izrek, ki pravi: Če te ni na spletu, pomeni, da ne obstajaš. Internet je od svojega začetka – z omrežjem *ARPANet*, v začetku šestdesetih, do danes doživel ogromen napredek, še posebej v zadnjem desetletju, ko lahko brez pretiravanja rečemo, da je praktično spremenil svet.

Internet je med drugim vnesel mnogo prepaha tudi na področje kartografije. Karte so se iz tradicionalnega fizičnega medija preselile v digitalno obliko, poleg tega so karte postale mnogo bolj dostopne ter uporabne. Pomemben mejnik v spletni kartografiji je bil dosežen konec devetdesetih let prejšnjega stoletja, ko je število prenosov spletnih kart preseгло število kart, natisnjenih na papir (Peterson, 2005). Kartografija na internetu se je razvijala postopoma, skupaj z razvojem drugih internetnih tehnologij. Prve spletne karte so bile praktično le skenirane analogne karte in v primerjavi s klasičnimi kartami niso ponujale nobene dodatne funkcionalnosti (v nekaterih pogledih so bile celo manj uporabne). Danes pa obstaja na trgu cela vrsta najrazličnejših dinamičnih kart z visoko stopnjo interaktivnosti, ki so zelo prijazne do uporabnika.

Pri razvoju spletne kartografije igrajo pomembno vlogo odprti standardi, ki jih razvija konzorcij OGC. Odprti standardi zagotavljajo konkurenčnost ter s tem povečujejo kakovost na področju spletne kartografije. Z uporabo odprtih standardov so prostorski podatki, ki jih strežemo preko omrežja lažje dostopni, omogočena je njihova ponovna ter večkratna uporaba. V preteklosti je bila namreč medopravilnost med različnimi sistemi otežena. Nad podatki je bilo potrebno izvajati različne konverzije ter graditi ločene vmesnike za dostop do podatkov na različnih platformah. Prednost standardnega vmesnika po OGC specifikaciji je v tem, da »z izgradnjo enega dostopaš do mnogih«. Kartografski strežniki oz. odjemalci kart različnih proizvajalcev lahko tako enostavno strežejo oz. zahtevajo karte ne ozirajoč se na proizvajalca programske opreme.

Naslednja pomembna stvar, ki se je v zadnjem času zgodila na področju spletne kartografije, je med drugim tudi razvoj t.i. odprtokodne programske opreme za spletno kartografijo. Takšna programska oprema praviloma uporablja odprte standarde ter je ponavadi prosto dostopna. Uporabniku omogoča več možnosti izbire programske opreme, poleg tega je možna poljubna modifikacija programa glede na specifične potrebe uporabnika. Odprtokodna programska oprema v zadnjih letih pridobiva na veljavi, in sicer predvsem na račun nekaterih uspešnih odprtokodnih projektov. Zaupanje v odprtokodno programsko opremo pa pridobivajo tudi manj napredni uporabniki, ki nimajo nujno znanja razvijalcev programske opreme.

V diplomski nalogi opisujemo postopke vzpostavitve in oblikovanja spletne karte Evrope na podlagi brezplačne odprtokodne GIS programske opreme ter odprtih standardov. Med drugim smo opisali OGC specifikacijo WMS, ki predpisuje obliko standardnega vmesnika za izmenjavo slikovnih kart preko spleta ter specifikacijo SLD, ki definira jezik za izdelavo uporabniških stilov pri oblikovanju kart. Prav tako smo opisali odprtokodno programsko opremo, ki jo potrebujemo za vzpostavitev spletne karte.

Spletno karto Evrope smo izdelali na podlagi podatkov prostorske podatkovne baze *Navstreets*, ki jo zagotavlja *Navteq* – eden največjih dobaviteljev prostorskih podatkov za cestno-navigacijske sisteme na svetu. Opisali smo tiste sestavine baze *Navstreets*, ki so pomembne za kartografsko predstavitev baze. Spletna karta bo v končni fazi omogočala poizvedbo po lokaciji ter izračun najkrajših oz. najhitrejših poti med različnimi kraji. Karta bo dostopna na spletnem portalu www.yellowlbs.com.

2 ODPRTI STANDARDI

2.1 Uvod

Standardizacija je postopek razvoja in uporabe niza pravil in dogovorov s čim večjim možnim številom potencialnih uporabnikov, zato da bi ustvarili jasnost in enotnost tam, kjer je različnost nezaželjena. Standardizacija je povezan proces (usklajen protokol): razvoja in opredelitve, formalnega sprejema, uveljavitve in tudi praktične uporabe standardov (Šumrada, 2005a).

Odpri standard pa je enostavno povedano proizvod skupine neodvisnih posameznikov, ki delujejo skupno pri razvijanju rešitev, ki se nanašajo na splošne potrebe in cilje ter od katerega imajo koristi tako proizvajalci kot tudi potrošniki. S sprejemom standardov lahko industrija doseže izredne stvari. Odpri standard deluje kot načrt oz. kot zavarovalna polica, ki omogoča različnim industrijam in tehnologijam, kot so npr. zdravstveno varstvo, finančne storitve, energetske storitve, avtomobilska industrija ter druge, da si delijo informacije, jih hitreje ter enostavneje povezujejo, in to z manjšimi stroški. Medopravilnost prinaša boljše dobrine, storitve ter pametnejše podatke (Saidin, 2007).

Odpri standardi so po svoji naravi neodvisni od platforme, razviti so na podlagi skupnega konsenza, tehnološko so nevtralni. Pomen odprtih standardov lahko ponazorimo s primerom iz ameriške zgodovine. Leta 1904 je v ameriškem mestu Baltimore izbruhnil požar. Na pomoč so poklicali gasilce iz vseh okoliških zveznih držav, ti pa so po prihodu požar lahko le opazovali, saj priključki na njihovih ceveh niso bili združljivi s hidranti v mestu. Standardni priključek bi lahko v tem primeru preprečil veliko katastrofo. Bogat izdelovalec hidrantov bi brez odprtih standardov priključek lahko patentiral in sam začel proizvodnjo, s tem bi pregnal manjše proizvajalce ter tako ustvaril monopol. V kolikor bi bil vzorec priključka na voljo brezplačno, bi konkurenčna podjetja lahko izdelovala svoje priključke in s tem bi bil zagotovljen prosti trg. Sporočilo tega primera ni, da se iz posla izloči monopolista, ampak da se igra odpre za vse igralce (Internet 3).

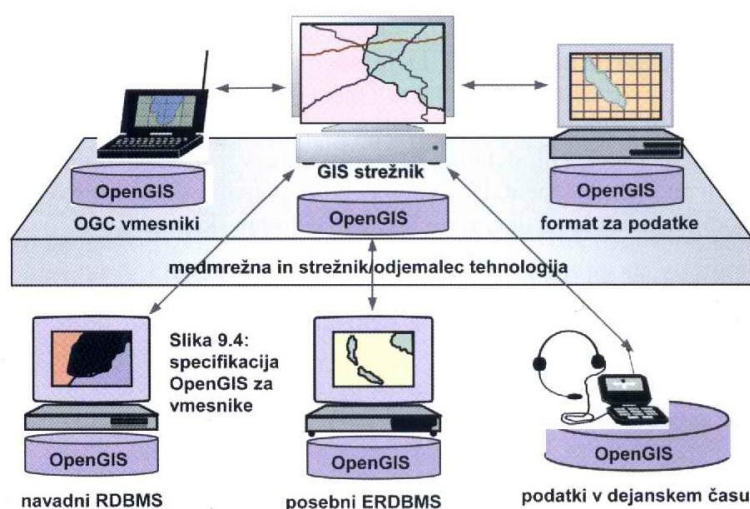
Evropska unija predpisuje naslednje minimalne karakteristike, ki jih mora imeti specifikacija in njen spremljajoči dokument, da se lahko smatra kot odprti standard:

- Standard mora biti privzet s strani neprofitne organizacije, njegov razvoj mora temeljiti na odprtem odločanju, na voljo mora biti vsem zainteresiranim strankam (konsenz se sprejema na osnovi večine).
- Dokument s specifikacijo standarda je dostopen brezplačno oz. za neznamen znesek. Vsem mora biti dovoljeno kopiranje, distribuiranje in brezplačna uporaba standarda oz. za neznamen znesek.
- Intelektualna lastnina, t.j. morebitni patenti prisotni znotraj standarda morajo nepreklicno postati brezplačno dostopni.
- Omejitev na ponovno uporabo standarda ne sme biti.

2.2 »Open GIS« standardi – uvod

Na področju GIS je »odprta standardizacija« v domeni konzorcija OGC (*Open Geospatial Consortium*). OGC je mednarodno industrijsko združenje podjetij, vladnih agencij in univerz, ki sodelujejo pri razvoju javno dostopnih standardov za zagotovitev potrebne povezljivosti različnih sistemov GIS. Glavni »proizvod« konzorcija OGC so t.i. *Open GIS* specifikacije oz. tehnični dokumenti z navodili za izgradnjo odprtega standardnega vmesnika, ki omogoča izmenljivost in deljivost prostorskih podatkov. Produkti in storitve, ki so izvedeni na osnovi *Open GIS* specifikacij omogočajo enostavno izmenjavo prostorskih podatkov po omrežju in medmrežju med različnimi sistemi GIS.

Različna GIS programska orodja za shranjevanje prostorskih podatkov uporabljajo različne nestandardne formate, zato imajo uporabniki GIS programske opreme pogosto težave z branjem oz. zapisovanjem podatkov v formatih, ki jih njihov sistem ne podpira. Izmenjava prostorskih podatkov med različnimi orodji GIS je bila v preteklosti zelo omejena, z množično uporabo interneta in izmenjavo podatkov preko omrežja pa je postala težava še toliko očitnejša, saj večkratna in ponovna uporaba podatkov ni bila mogoča. Implementacija OGC vmesnika v sisteme GIS rešuje težavo, saj uporabniki dostopajo do podatkov preko standardnega vmesnika (slika 1).



Slika 1: Enostavna izmenjava prostorskih podatkov na osnovi tehnologije »OpenGIS« (vir: Šumrada, 2005a).

Vsi računalniški sistemi, od delovne postaje, osebnega računalnika, prenosnikov in ročnih računalnikov ter mobilnih telefonov, ki so opremljeni s specifikacijo OGC za vmesnike, lahko enostavno izmenjujejo prostorske podatkovne nize v zapisu GML (Šumrada, 2005a).

2.3 Standardi in specifikacije konzorcija OGC

OpenGIS standardi in specifikacije so tehnični dokumenti namenjeni razvijalcem programske opreme in detajlno opisujejo postopke za izgradnjo *OpenGIS* vmesnika v njihove produkte in storitve. Nekaj *OpenGIS* specifikacij in standardov, ki med drugim posegajo tudi na področje spletne kartografije opisujemo v nadaljevanju:

- Spletni servisi OGC (*OpenGIS web services*),
- GML (*Geography Markup Language*),
- SLD (*Styled Layer Descriptor*),
- Specifikacija izvedbe enostavnih pojavov (*OpenGIS Simple Features Implementation Specification*),
- idr.

Spletni servisi OGC (*OpenGIS Web Services*)

Spletni servis je aplikacija, ki sprejema in obdeluje uporabniške oz. druge programske zahteve po omrežju, kot je denimo internet. Izvedbeno spletni odjemalec pošlje zahtevo kot

sporočilo spletnemu strežniku, ki prejeto zahtevo obdela in ustrezno odgovori odjemalcu. Spletni servis se tako opredeli kot izmenjava sporočil ali podatkov med spletnim strežnikom in ustreznim odjemalcem, kjer se oba predvsem pojmuteta kot ustrezni aplikaciji oz. delujoča programa. Spletni servis OGC opredeljuje standardne vmesnike, protokole in poenoten način kodiranja prostorskih podatkov za uporabo na medmrežju ter zlasti na spletu. Ti standardni spletni servisi so posebej namenjeni za podporo izmenjavi prostorskih podatkov, aplikacij in navodil po omrežju v dejanskem času (Šumrada, 2005a).

Specifikacije, ki tvorijo skupino spletni servisi OGC so:

- *OpenGIS* specifikacija za spletne kartografske servise (*OpenGIS WMS – Web Map Service Specification*),
- *OpenGIS* specifikacija za objektne spletne servise (*OpenGIS WFS – Web Feature Service Specification*),
- *OpenGIS* specifikacija za spletni prostorski podatkovni sloj (*OpenGIS WCS – Web Coverage Service Specification*)

Specifikacija OGC za spletne kartografske servise (WMS)

OGC definira WMS kot spletni kartografski servis, ki na osnovi prostorskih podatkov, proizvaja dinamične karte. Karta je pri tem opredeljena kot grafična predstavitev prostorskih podatkov v ustreznem digitalnem slikovnem formatu, primernem za prikaz na računalniškem zaslonu. WMS strežnik ne zagotavlja dostopa do dejanskih prostorskih podatkov, ampak le do kart, ki so običajno posredovane v JPEG, GIF, PNG, idr. slikovnih datotekah oz. lahko tudi v vektorskih formatih, kot sta npr. SVG ali PDF.

Specifikacija OGC za spletne kartografske servise predpisuje standardne protokole za komunikacijo med kartografskim strežnikom ter odjemalcem. Z implementacijo te specifikacije, WMS strežniki na poenoten način strežejo podatke, na drugi strani pa jih WMS odjemalci na poenoten način sprejemajo. Tako lahko odjemalci enostavno združujejo podatke iz različnih porazdeljenih WMS strežnikov.

Odjemalec (ki je ponavadi ustrezna spletna aplikacija) komunicira z WMS strežnikom preko treh standardnih operacij. Z namenom pridobitve informacij o podatkih in funkcijah,

ki jih ponuja WMS strežnik, odjemalec od strežnika najprej pridobi metapodatke (zahteva *GetCapabilities*). Metapodatki so zapisani v XML (*Extensible Markup Language*) formatu in vsebujejo naslednje informacije:

- slikovni formati (npr. GIF, JPG, SVG), ki jih podpira kartografski strežnik,
- seznam podatkovnih slojev, ki jih ponuja strežnik,
- seznam in natančen opis posameznih oblikovalskih stilov,
- veljavno območje prikaza podatkov, ki jih ponuja strežnik,
- seznam prostorskih referenčnih sistemov, v katerih so dostopni podatki,
- kontaktni podatki administratorja WMS strežnika, idr.

Na osnovi teh informacij odjemalec nato izdela zahtevo po karti (*GetMap*), v kateri se opredelijo:

- podatkovni sloji, ki jih želimo prikazati na karti (*Layers*),
- oblikovalski stili, ki jih želimo prikazati na karti (*Styles*),
- prostorski referenčni sistem (*SRS – Spatial Reference System*),
- slikovni format karte (*Format*),
- območje prikaza (*Bounding Box*),
- velikost vrnjene slike – dolžina in višina v pixlih (*Width, Height*)

Tretja standardna operacija, ki pa ni obvezna za WMS strežnik, je zahteva *GetFeatureInfo*. Na podlagi te zahteve lahko uporabnik preko uporabniškega vmesnika izvaja poizvedbe o različnih pojavih na karti.

Vse tri zahteve odjemalec WMS strežniku podaja v HTTP obliki preko URL naslova (*Universal Resource Locator*). Vsak takšen URL pa vsebuje: ime protokola (http), gostitelja, pot, znak »?«, verzijo specifikacije, tip zahteve (*GetCapabilities, GetMap, GetFeatureInfo*), tip storitve (WMS), nato pa še nekatere specifične parametre, ki se nanašajo na tip zahteve in ki so med seboj ločeni z znakom »&«. Primer zahtev *GetCapabilities* in *GetMap* opisujeta spodnja izraza:

Primer zahteve *GetCapabilities*:

<http://imegostitelja.si/geoserver/wms?VERSION=1.1.1&REQUEST=GetCapabilities&SERVICE=WMS&>

Na osnovi te zahteve bo WMS strežnik vrnil XML datoteko z opisom zmogljivosti WMS strežnika (metapodatki).

Primer zahteve *GetMap*:

http://imegostitelja.si/geoserver/wms?VERSION=1.1.1&REQUEST=GetMap&SERVICE=WMS&LAYERS=Oceans,Countries,Cities&STYLES=SimplePoly,Country,Cities&BBOX=14.30,45.00,15.30,46.15&SRS=EPSG:4326&WIDTH=600&HEIGHT=400&FORMAT=image/png&

Na osnovi te zahteve bo WMS strežnik vrnil karto v obliki slikovne datoteke. Posamezne parametre zahteve *GetMap* opisuje tabela 1a.

Tabela 1a: Opis posameznih parametrov zahteve »GetMap«.

<i>http://monolit.si/geoserver/wms?</i>	Protokol (<i>http://</i>), gostitelj (<i>imegostitelja.si/</i>), pot (<i>geoserver/wms</i>)
<i>VERSION=1.1.1&</i>	Verzija WMS specifikacije
<i>REQUEST=GetMap&</i>	Tip WMS zahteve
<i>SERVICE=WMS&</i>	Tip storitve
<i>LAYERS=Oceans,Countries,Cities&</i>	Podatkovni sloji
<i>STYLES=SimplePoly,Country,Cities&</i>	Oblikovalski stili
<i>BBOX=14.30,45.00,15.30,46.15&</i>	Območje prikaza: spodnji levi kot (x, y), zgornji desni kot (x, y)
<i>SRS=EPSG:4326&</i>	Identifikator za tip prostorskega referenčnega sistema: EPSG:4326 je identifikator elipsoida WGS-84
<i>WIDTH=600&HEIGHT=400&</i>	Velikost vrnjene slike: dolžina in višina v pixlih
<i>FORMAT=image/png&</i>	Format vrnjene slikovne datoteke (png)

Specifikacija OGC za objektne spletne servise (WFS)

Za razliko od specifikacije WMS, ki specificira način izmenjave kart preko spleta v obliki slikovnih datotek, pa WFS (*Web Feature Service*) specifikacija standardizira način izmenjave dejanskih podatkov preko spleta. WFS strežnik uporabniku vrne podatke v GML datoteki in uporabnik lahko na ta način ureja podatke (briše in spreminja vsebino, dodaja nove elemente) ter izvaja prostorske analize in poizvedbe na podatkih. Poizvedovanja se sestavljajo s pomočjo posebnega jezika CQL (*OpenGIS Common Query Language*). Odjemalec v CQL jeziku sestavi poizvedbo ter jo posreduje strežniku, ta pa iskane podatke vrne odjemalcu v GML datoteki.

Specifikacija OGC za spletni prostorski podatkovni sloj (WCS)

WCS je servis, ki zagotavlja dostop do rastrskih podatkov preko omrežja. Za razliko od WMS, ki uporabniku vrne le slikovno datoteko, WCS vrne dejanske podatke – t.j. rastrsko sliko, skupaj s pripadajočimi atributi. Na podlagi tega lahko uporabnik izdeluje različne poizvedbe ter analize nad rastrskimi podatki. Rastrski podatki opisujejo zvezno spreminjaje pojavov, za razliko od vektorskih podatkov, ki jih vrne WFS – ti opisujejo diskretne prostorske pojave, ki so predstavljeni s točkami, linijami ter ploskvami.

GML (*Geography Markup Language*)

GML je na XML temelječ jezik za označevanje, ki služi za shranjevanje in prenos prostorskih podatkov po medmrežju. Predstavlja nevtralen kodni format, ki je neodvisen od internih formatov raznih proizvajalcev orodij GIS. Omogoča usklajeno (opisno) kodiranje različnih oziroma načeloma kakršnihkoli, tako grafičnih kot opisnih, prostorskih podatkov. Hkrati zagotavlja ustrezno podporo za poenostavitev prenosa in shranjevanje, razne obdelave, analize in prikazovanje prostorskih podatkov (Šumrada, 2005a).

SLD (*Styled Layer Descriptor*)

Styled Layer Descriptor je OGC specifikacija, ki definira XML slovnico za izdelavo uporabniških stilov pri oblikovanju kart. SLD specifikacijo podrobneje opisujemo v nadaljevanju diplomske naloge.

Specifikacija izvedbe enostavnih pojavov (*OpenGIS Simple Features Implementation Specification*)

Specifikacija izvedbe enostavnih pojavov zagotavlja standardne metode, da lahko sistemi GIS posredujejo in izmenjujejo enostavno (vektorsko) geometrijo (točke, linije in poligoni), uporabljeni prostorski referenčni sistem in dodane tematske attribute pojavov (Šumrada, 2005a).

3 ODPRTOKODNA PROGRAMSKA OPREMA

3.1 Uvod

Odpriokodna programska oprema v zadnjih letih postaja dobra alternativa »običajni« – lastniški programski opremi. Vse več uporabnikov spoznava njen pomen in prednosti, ki jih prinaša. Nekateri uspešni odprtokodni projekti so dokazali, da lahko takšen tip programske opreme povsem konkurira lastniškimi. Odpriokodna programska oprema se praviloma naslanja na odprte standarde ter je v splošnem prosto dostopna. Uporabniku med drugim omogoča večjo mero svobode pri izbiri programske opreme ter možnost poljubne prilagoditve programa po meri uporabnika.

Vsak program deluje na osnovi inštrukcij, ki računalniku narekujejo, kako naj se le ta izvaja. Razvijalec programske opreme te inštrukcije najprej zapiše v človeku berljivem zapisu ali t.i. izvorni kodi (*source code*). Preden računalnik izvrši ta navodila, pa jih pretvori v kodo, ki je razumljiva računalniku. Ta koda se imenuje objektna koda in je zapisana v binarnem formatu. Vsaka programska oprema je dobavljiva skupaj z objektno kodo. Razlika med »običajno« – lastniško programsko opremo in odprtokodno programsko opremo je v tem, da je slednja dostopna tudi z izvorno kodo, ki je uporabniku razumljiva ter mu med drugim omogoča vpogled v samo logiko delovanja programa. Lastniška programska oprema je ponavadi plačljiva, prepovedano je njeno razmnoževanje, kopiranje in posojanje, medtem ko odprtokodna programska oprema nima teh omejitev. Njena uporaba je v splošnem brezplačna, poleg tega pa ima uporabnik pravico spreminjati izvorno kodo in si s tem prilagoditi delovanje programa po svojih potrebah.

3.2 Lastnosti odprtokodne programske opreme

Pri opisovanju odprtokodne programske opreme velja izpostaviti predvsem njene naslednje dobre lastnosti:

- visoka stopnja stabilnosti in zanesljivost programske opreme,
- visoka stopnja varnosti (manjša ranljivost za različne viruse),
- visoka fleksibilnost,
- praktično neomejena možnost modifikacije in prilagoditve programa glede na potrebe uporabnika,

- manjši stroški povezani z nakupom programske opreme,
- ne ostaneš zaklenjen znotraj enega proizvajalca programske opreme,
- programska oprema se neprenehoma razvija,
- uporaba odprtih standardov.

Odprtokodna programska oprema ima predvsem naslednje dobre lastnosti. Programske opreme ne razvija eno samo podjetje ampak razpršena neformalna skupnost razvijalcev. Odprtokodni projekti se razvijajo na internetu, na očeh široke odprtokodne skupnosti. Aktivni udeleženci skupnosti sodelujejo v diskusijah, razvijajo programsko kodo ter poročajo o odkritih napakah v programih. Več oči več vidi in za razliko od lastniške programske opreme, ki jo razvija zaprta in ponavadi manjša skupina razvijalcev, so tu morebitne težave (npr. z varnostjo) lahko hitreje odkrite in odpravljene. Poleg tega so nove, izboljšane verzije programa dostopne zelo hitro (lahko celo vsakodnevno). Znana filozofija odprtokodne skupnosti je namreč »izdajaj hitro, izdajaj pogosto« (*»release fast, release often«*). Razvijalci odprtokodne programske opreme so pri svojem delu tudi zelo motivirani, saj vedo, da s tem, ko vsak prispeva del programske kode, lahko skupaj razvijejo res dober program.

Zaradi svojih dobrih lastnosti odprtokodna programska oprema v zadnjih letih pridobiva čedalje več zaupanja s strani uporabnikov. Kljub vsemu pa ponekod še vedno obstaja prepričanje, da je njen razvoj v domeni nekaterih alternativnih skupin, katere posamezniki imajo tipične lastnosti računalniških zanesenjakov oz. t.i. hekerjev. Vse to je mogoče res držalo v času, ko so nastajali prvi odprtokodni programi, danes pa so stvari drugačne in odprtokodni programi so postali profesionalni, upravljanje z njimi pa je postalo povsem običajna praksa. Veliko pomembnih odprtokodnih projektov namreč podpirajo tudi komercialne družbe, kot sta npr. IBM in SUN (Saccon, 2003).

Poleg dobrih lastnosti odprtokodne programske opreme pa je potrebno omeniti tudi nekatere tipične pomanjkljivosti:

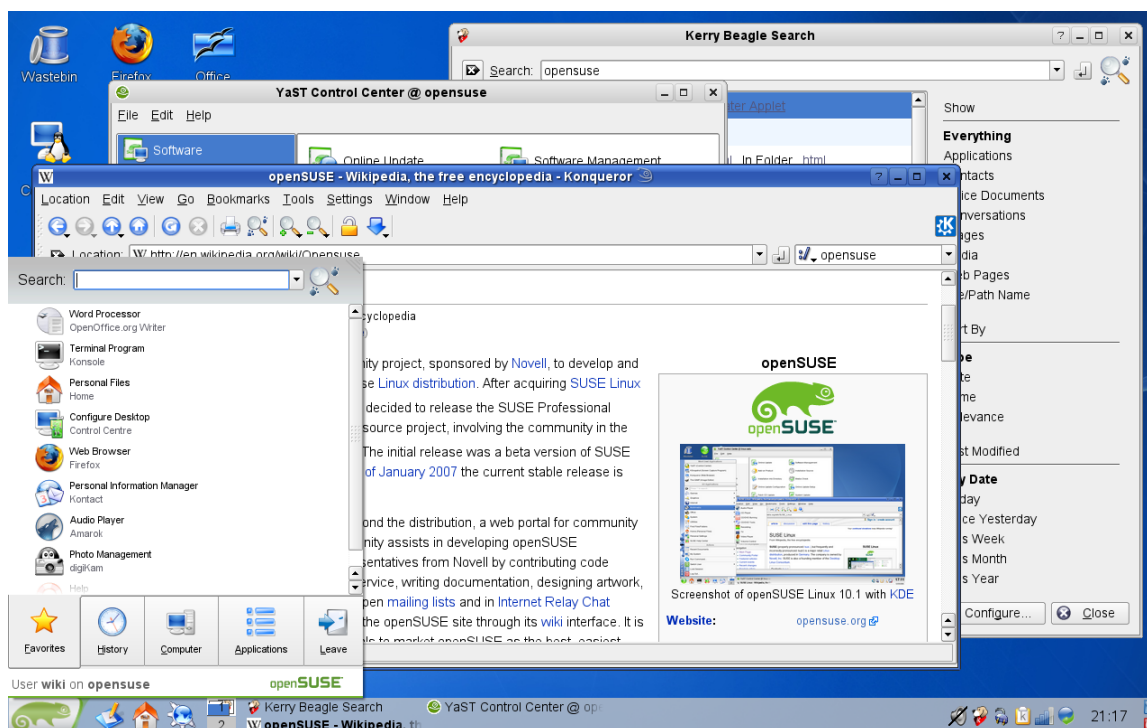
- v splošnem slabši uporabniški vmesnik odprtokodnih aplikacij,
- ponavadi ni zagotovljene zanesljive tehnične podpore.

Razlog za pomanjkljivosti tiči predvsem v naravi razvijalcev odprtokodne programske opreme, ki so ponavadi strokovnjaki, kar se tiče funkcionalnosti ter na drugi strani ne posvečajo velike pozornosti ergonomiji programa. Uporabniška izkušnja pa je pogosto povezana s kakovostjo grafičnega vmesnika in kvaliteto tehnične dokumentacije. Odprtokodni projekti, ki kot taki nimajo komercialnega ozadja, med prostovoljci nimajo ustreznih grafičnih oblikovalcev iz piscev tehnične dokumentacije, proračun pa jim ne omogoča plačevanja zunanjih storitev. Kljub temu pa skoraj vse odprtokodne skupnosti zagotavljajo vsaj kakšno obliko tehnične podpore. Ponavadi v obliki različnih »mailing« list ter uporabniških forumov, v nekaterih primerih pa celo komercialno podporo (Saccon, 2003).

3.3 Uspešni odprtokodni projekti

Nekateri uspešni odprtokodni programi, ki so si uspeli pridobiti širok krog uporabnikov so:

- *Linux* (visoko zmogljiv in priljubljen odprtokodni operacijski sistem),
- *Apache* (priljubljen spletni strežnik),
- *Mozilla Firefox* (spletni brskalnik),
- *Mozilla Thunderbird* (e-poštni odjemalec),
- *OpenOffice* (odprtokodni pisarniški paket).



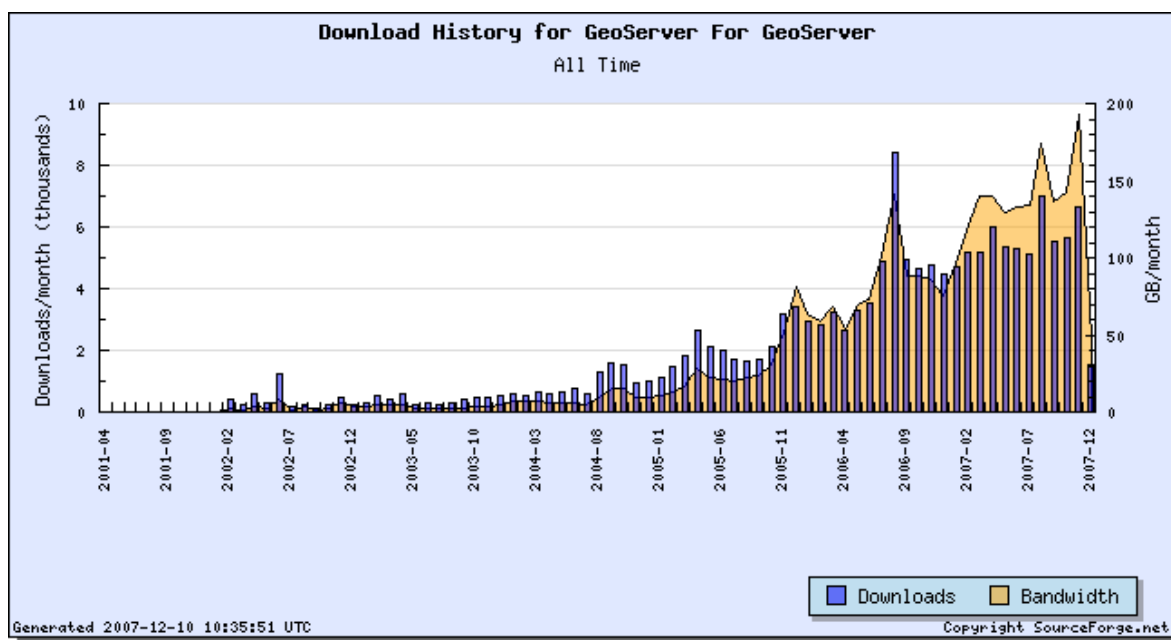
Slika 2: »Suse« – priljubljena distribucija »Linuxa« za namizne uporabnike (vir: Wikipedia).

3.4 Licenciranje odprtokodne programske opreme

Kljub temu, da odprtokodna programska oprema ponuja uporabniku veliko mero svobode v zvezi z njeno uporabo, pa le ta ni dostopna povsem brez omejitev. Uporabniki programa so zavezani upoštevati pogoje, ki so zapisani znotraj licenčne pogodbe. Obstajajo različne licence, ki uporabniku dajejo različne pravice. Najbolj uporabljene licence so: GPL, LGPL, BSD. GPL (*General Public Licence*) je ena izmed najbolj uporabljenih licenc, med drugimi jo uporablja tudi *Linux*. Ta licenca je najbolj restriktivna med vsemi in predpisuje, da mora biti vsak nov program, ki je izpeljan iz GPL licenciranega programa, dostopen skupaj z izvorno kodo. BSD licenca je na drugi strani med manj restriktivnimi in dovoljuje uporabo, modifikacijo in distribucijo novega programa brez zahteve po objavi izvorne kode novega programa. Pravila, ki jih mora izpolnjevati licenca odprtokodne programske opreme, predpisuje organizacija OSI (*Open Source Initiative*). OSI je neprofitna organizacija, ustanovljena leta 1998 z namenom, da poudarja in promovira vrednote odprte kode ter skrbi za standardizacijo na tem področju.

3.5 Odprtokodna programska oprema za spletno kartografijo

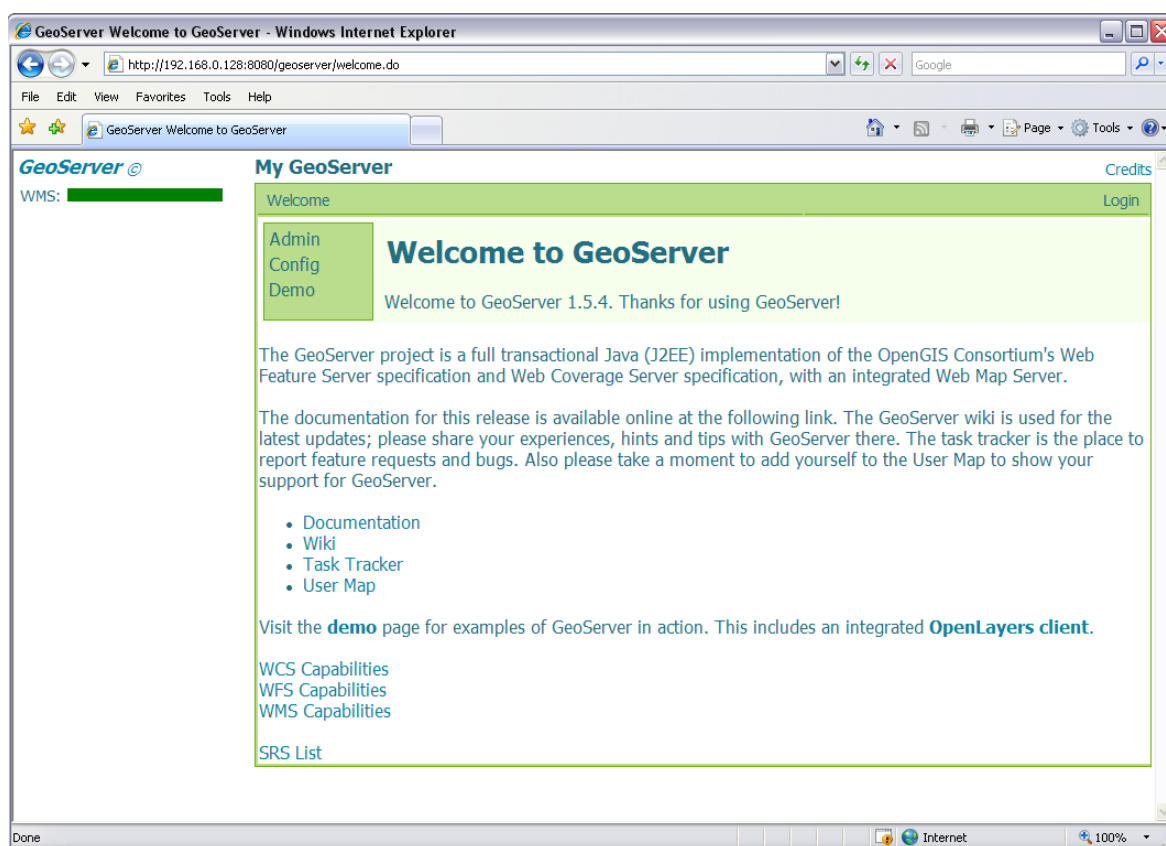
K razmahu odprtokodne programske opreme za spletno kartografijo so v veliki meri pripomogli OGC standardi. Omogočeno je enostavno vključevanje standardov v aplikacije, poleg tega lahko odprtokodna skupnost aktivno sodeluje pri razvoju standardov. Odprtokodna programska oprema je zelo popularna v akademskih sferah, podpirajo pa jo tudi različne neprofitne organizacije kot npr. FAO in UN, kar dokazuje, da je tehnično povsem primerna za njihove potrebe. Na drugi strani pa mnoga podjetja še vedno nimajo zaupanja v to alternativno obliko programske opreme (Nasr, 2007). Kljub temu pa nekateri statistični podatki kažejo, da uporaba odprtokodne programske opreme za spletno kartografijo in GIS v zadnjih letih konstantno narašča. Slika 3 prikazuje statistiko prenosov priljubljenega kartografskega strežnika *Geosever* od nastanka projekta leta 2001 do danes.



Slika 3: Zgodovina prenosov odprtokodne aplikacije »Geoserver« v obdobju 2001-2008 (vir:Internet 5).

Za vzpostavitev spletne karte se poleg specifične programske opreme, ki je namenjena izključno spletni kartografiji oz. GIS, večinoma uporablja odprtokodna infrastruktura, ki je že na voljo. Npr. odprtokodni operacijski sistemi (različne distribucije *Linuxa*), spletni strežniki (kot npr. *Apache*) in podatkovne baze (kot npr. *MySQL* in *PostgreSQL*). Priljubljene odprtokodne aplikacije, specifične za spletno kartografijo in GIS so:

- *MapServer*, *Geoserver* (priljubljena kartografska strežnika, ki izpolnjujeta OGC standarde),
- *PostGres/PostGIS* (sistem za upravljanje z bazo podatkov in njegova prostorska razširitev),
- *Geotools* (Java knjižnica, ki v skladu z OGC standardi zagotavlja ustrezne metode za manipulacijo s prostorskimi podatki),
- *OpenLayers* (JavaScript knjižnica, ki zagotavlja ustrezne metode za postavitev dinamičnega pregledovalnika kart v spletni brskalnik),
- *MapBuilder* (še en priljubljen pregledovalnik spletnih kart).



Slika 4: »Geoserver« – priljubljeni odprtokodni kartografski strežnik.

Odprtokodna GIS programska oprema je zelo konkurenčna lastniški programski opremi. Nasr (Nasr, 2007) opisuje primerjavo med odprtokodnim kartografskim strežnikom *Mapserver* (ki je primerljiv tudi z *Geoserverjem*) ter nekaterimi lastniškimi programi kot so *ArcIMS*, *GeoMedia*, *WebMap* ter *MapExtreme*. Ugotavlja, da *MapServer* izpolnjuje več

OGC specifikacij kot večina komercialnih programov. Za razliko od slednjih uporabnikom omogoča uporabo različnih komercialnih podatkovnih formatov (*Oracle*, *shape* datoteke, idr.) in tudi nekaterih odprtih formatov, kot je npr. *PostGIS* format. Pomanjkljivost *MapServerja* pa se je izkazala pri sami instalaciji programa, pri kateri je avtor naletel na nekaj težav. Z odpravo takšnih tipičnih lepotnih napak, ki so značilne za mnoge odprtokodne projekte, velja v prihodnosti računati na še večjo zaupanje in uporabo tega alternativnega vira programske opreme tudi na področju spletne kartografije.

4 GIS (GEOGRAFSKI INFORMACIJSKI SISTEM)

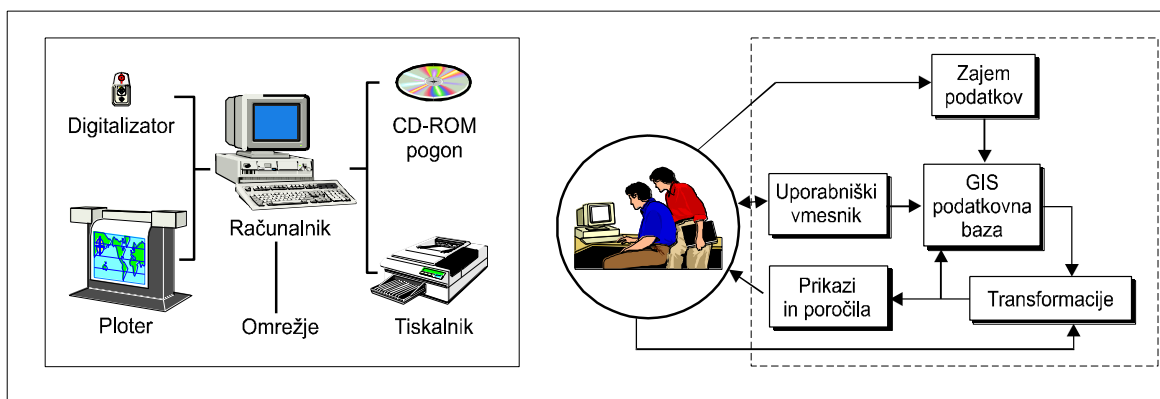
Digitalna in s tem spletna kartografija ter sistemi GIS se v mnogočem prepletajo. Digitalne karte so zgrajene na osnovi podatkov GIS prostorske podatkovne baze, na drugi strani so lahko oblikovane z različnimi orodji GIS. Za to, da lahko karto ustrezno oblikujemo, je med drugim potrebno poznati lastnosti podatkovne baze in zapisa podatkov v bazi. V nadaljevanju na kratko opisujemo splošno teorijo sistemov GIS ter prostorskih podatkovnih baz.

4.1 Opredelitev geografskega informacijskega sistema

Geografski informacijski sistem je sestav izkušenega osebja, uporabnikov, lokacijskih in opisnih podatkov, analitičnih metod, računalniških programov, strojne in komunikacijske opreme, ki so organizirani za upravljanje ter samodejno obdelavo podatkov s ciljem zagotavljanja geografskih informacij skozi razne predstavitve prostorskih podatkov. Potrebne funkcionalnosti v orodju GIS so tako predvsem naslednje:

- sposobnost zajemanja, vzdrževanja in posredovanja sestavljenih prostorskih podatkov,
- sposobnost povezovanja geokodiranih lokacijskih in opisnih podatkov,
- sposobnost izvajanja vektorskih ploskovnih in mrežnih prostorskih analiz ter rastrskega modeliranja s tehniko prekrivanja raznih podatkovnih (tematskih) slojev,
- sposobnost raznolikega prikazovanja geokodiranih prostorskih podatkov

(Šumrada, 2005a). Glavne sestavine računalniške strojne in programske opreme prikazuje slika 5.



Slika 5: Glavne sestavine računalniške strojne in programske opreme v GIS (vir: Burrough, 1998).

4.2 Prostorski podatki

Prostorski podatki so značilni podatki o izbranih geografskih pojavih. V geografskem informacijskem sistemu sestavljata podatke dve značilni sestavini: prostorski in opisni atributi. Grafika se v sistemih GIS nanaša predvsem na tiste značilnosti in lastnosti geografskih pojavov, ki so podane s koordinatami.

- Prostorski atribut pojava podajajo njegove grafične, lokacijske, geometrijske in topološke značilnosti.
- Opisni atributi podajajo tematsko vsebino geografskega pojava. GIS ponavadi dovoljuje številčne, znakovne, časovne in datumske opisne podatke.

(Kvamme et al, 1997)

4.3 Prostorska podatkovna baza kot model stvarnosti

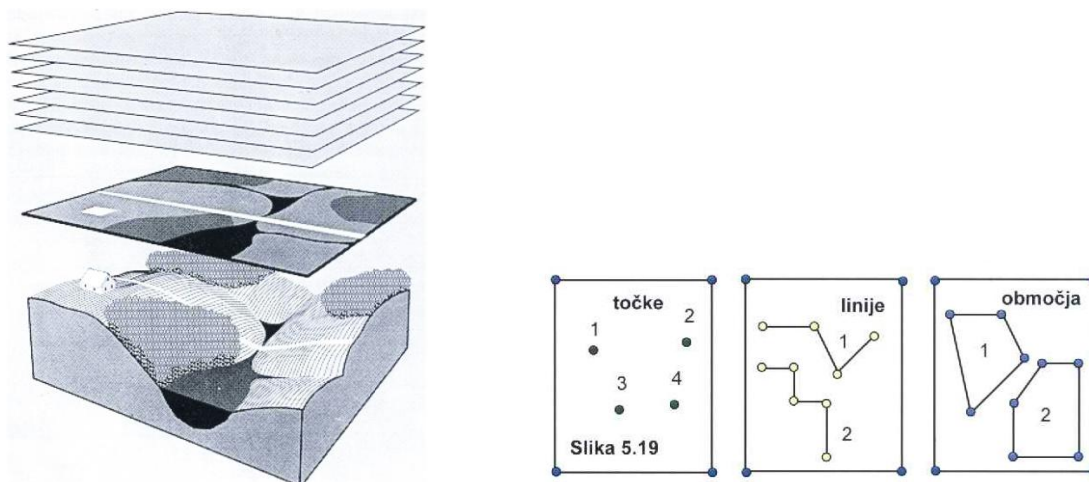
Svet v katerem živimo želimo razumeti, razložiti in opisati, vendar je le ta kompleksen in kot celota težko razumljiv. V ta namen si ustvarjamo modele stvarnosti, ki na poenostavljen način predstavljajo različne vidike realnosti. Podatkovne baze so zgrajene iz takšnih modelov stvarnosti in predstavljajo temeljni korak k spoznavanju narave in stanja realnosti. Prostorska podatkovna baza je zbirka podatkov s prostorsko komponento, ki deluje kot model stvarnosti, saj (vir: internet 1):

- podatkovna baza na poenostavljen način predstavlja stvarni pojav,
- izbrani stvarni pojav je »dovolj pomemben« za predstavitev v digitalni obliki,
- digitalna predstavitev stvarnega pojava se lahko nanaša na preteklost, sedanost ali prihodnost.

Pojmovno in izvedbeno ločimo dve povezani ravni ponazoritve stvarnosti. Podatkovni model obravnavamo kot interpretacijo področja obravnave na t.i. tipski ravni, kjer so pojmovno opredeljeni izbrani poenostavljeni stvarni objektni tipi, ki se izvedbeno imenujejo razredi. Ti so nadalje formalno opredeljeni v t.i. shemah podatkovnega modela. Na pojavni ravni nastopajo dejanski podatki o posameznem pojavu ali objektu, ki so vsebinsko skladni s pripadajočo shemo objektnega tipa. Podatke o več dejanskih prostorskih objektih lahko smiselno združimo v večjo logično enoto, ki jo imenujemo podatkovni sloj. Ta je nadalje lahko fizično izveden kot podatkovni niz ali datoteka oz. kot skupina internih zapisov v bazi podatkov GIS (Šumrada, 2005a).

4.4 Kartografski podatkovni model

V sodobnih sistemih GIS prevladuje t.i. kartografski podatkovni model, ki je izveden prevladujoče kot sestavljena relacijska baza podatkov. To je tradicionalni dvorazsežni (2D) vektorski podatkovni model baze GIS, ki izhaja iz kartografskega načina izdelave topografskih kart velikih in srednjih meril. Tradicionalni podatkovni model, ki je v rabi v sistemih GIS, temelji na razstavitvi vsebine topografske karte na ustrezne po vsebini ločene plasti. Take plasti ali podatkovni sloji denimo ločeno vsebujejo relief, vodovje, zgrajene objekte in vegetacijo kot sklop enakovrednih prostorskih pojavov. Zajeti so vsak na svojem vsebinskem sloju. Za podajanje prostorskih pojavov na tematskih plasteh ponavadi uporabljamo njihove koordinate kot osnovni način geokodiranja. Digitalni prostorski podatki se lahko prikažejo v kateremkoli merilu in so neodvisni od kartografske projekcije. Geometrični in opisni podatki o prostorskih pojavih so navadno shranjeni ločeno. Določen objekt na karti je v značilni relacijski podatkovni bazi zaradi normalizacije razstavljen na številne pomenske elemente, ki so praviloma shranjeni ločeno v različnih tabelah. Kartografski podatkovni model je miselni in tehnološki pristop, ki ga je uvedla družba *ESRI* in je med drugim tradicionalno izveden tudi v njihovih orodjih GIS. (Šumrada, 2005a)



Slika 6: Kartografski podatkovni model in osnovni tipi grafičnih objektov (vir: Šumrada, 2005a).

2R objektni podatkovni model za kartografsko ponazoritev lege v sistemih GIS je sestavljen iz treh osnovnih tipov grafičnih elementov ali gradnikov, to so točka, linija in

območje (ploskev). Geometrična predstavitev in pomen grafičnih objektnih tipov sta namenjena podajanju posplošene abstrakcije poljubno sestavljene geometrije izbranih stvarnih geografskih objektov ali pojavov. V kartografskem podatkovnem modelu služi predvsem za predstavitev osnovnih kartografskih lastnosti geografskih objektov.

- Točka je najbolj enostaven grafični objektni tip ali razred brez dimenzij (0R), ki ima samo lokacijo brez dodatnih prostorskih lastnosti. Ima svoj identifikator in navadno se prikaže z ustreznimi pogojnimi znaki.
- Linija povezuje vsaj dve točki in je enorazsežen grafični razred (1R), ki ima prostorsko lokacijo, dolžino, oznako in obliko. Linijski objekti so lahko sestavljeni iz enega ali več povezanih usmerjenih odsekov, ki so po obliki preme linije ali krivulje. Edina merljiva lastnost linije je njena dolžina.
- Območje (ploskev) je dvorazsežen grafični objektni tip, ki ima površino in obseg kot edini merljivi lastnosti (2R). Obliko, velikost in lego ploskve določajo usmerjeni odseki, ki ga obkrožajo. Vsako območje določajo vsaj tri obodne linije (meja), katere določajo vsaj tri lomne točke.

(Kvamme et al, 1997)

5 KARTOGRAFIJA IN SPLET

5.1 Internet

Internet kot sodobno sredstvo komunikacije ima zelo velik ter stalen vpliv na naša življenja. Redkokateri vidik našega vsakdanjega življenja in delovanja ni vsaj malo povezan z internetom. Internet nam omogoča dostop do podatkov, shranjenih na strežnikih, ki se lahko nahajajo na drugem koncu sveta in to takorekoč v trenutku. Je nepogrešljiv vir informacij in v današnji informacijski družbi bi si življenje brez njega najbrž težko zamišljali.

Internet je opisan na veliko različnih načinov. Enostavno bi ga lahko opisali kot sistem za prenos datotek med računalniki. Te datoteke lahko vsebujejo tekst, slike, grafike, zvok, animacije, filme in celo računalniške programe. V smislu strojne opreme lahko internet obravnavamo kot fizično zbirko računalnikov, usmerjevalnikov in hitrih komunikacijskih linij, v smislu programske opreme pa je to zbirka računalniških omrežij, ki temeljijo na TCP/IP protokolu. V smislu vsebine bi ga lahko opredelili kot zbirko porazdeljenih virov. Končna in najpomembnejša definicija pa izhaja iz človekovega zornega kota in internet opredeljuje kot veliko in vseskozi razširjajočo skupnost ljudi, ki prispevajo k njeni vsebini in na drugi strani uporabljajo njene vire (Peterson, 2005).

5.2 Internetni protokoli

Za prenos podatkov med različnimi računalniki se uporabljajo internetni protokoli, ki jih lahko smatramo tudi kot nekakšno zbirko navodil za medsebojno komunikacijo. Vsi računalniki na internetu komunicirajo med sabo s pomočjo TCP/IP (*Transmission Control Protocol / Internet Protocol*) protokola. Podatki se na osnovi tega protokola prenašajo iz strežnika – t.j. programa na oddaljenem računalniku, k odjemalcu – t.j. brskalniku na lokalnem računalniku (in obratno). Internet pa sestoji še iz različnih drugih protokolov oz. servisov med katerimi so najpomembnejši:

- svetovni splet (WWW – *World Wide Web*),
- elektronska pošta,
- FTP (*File Transfer Protocol*) – namenjen je izmenjavi datotek med oddaljenimi računalniki,

- *telnet* – omogoča dostop do oddaljenega računalnika in uporabe programov ter drugih virov na tem računalniku.

5.3 Svetovni splet

Eden najpomembnejših internetnih servisov je zagotovo svetovni splet ali na kratko splet. Gre za sistem povezanih hipertekstovnih dokumentov, ki so napisani na osnovi standardiziranega označevalnega jezika HTML (*HyperText Markup Language*) in lahko v osnovi vsebujejo tekst ter različne večpredstavnostne vsebine kot so slike, zvok, video posnetki. Takšnim hipertekstovnim dokumentom z drugo besedo pravimo spletne strani. Spletne strani so shranjene na spletnih strežnikih, do njih pa dostopamo preko spletnih brskalnikov na osnovi t.i. URL-jev (*Uniform Resource Locator*) oz. enoličnih internetnih naslovov. Za prenos HTML strani iz strežnika do odjemalca (oz. spletnega brskalnika) se uporablja protokol HTTP (*Hypertext Transfer Protocol*).

Prvotna zasnova spleta je bila namenjena predvsem prenosu informacij ter enostavnih spletnih vsebin, ki so vsebovale le tekst ter spremljajoče večpredstavnostne elemente. HTML ni omogočal izdelave zahtevnejših spletnih strani, prav tako ni omogočal poganjanja različnih aplikacij preko spleta. Ker so potrebe po zahtevnejših spletnih vsebinah naraščale, se je sčasoma razvil nov razširjen HTML, ki ima sposobnost vključevanja različnih programskih skriptnih jezikov kot so npr. CGI (*Common Gateway Interface*), ASP (*Active Server Page*), JSP (*Java Server Page*). Ti omogočajo izdelavo dinamičnih in interaktivnih spletnih strani. S splošnim naraščanjem procesorskih sposobnosti osebnih računalnikov pa je postala ena izmed možnosti tudi procesiranje spletnih vsebin na odjemalčevi strani. V HTTP dokumente se tako lahko vpenja različne t.i. Java skripte oz. odjemalcu se lahko posreduje t.i. javanske programčke (*Java applets*), ki prenesejo del procesiranja na stran odjemalca.

5.4 Internet in kartografija

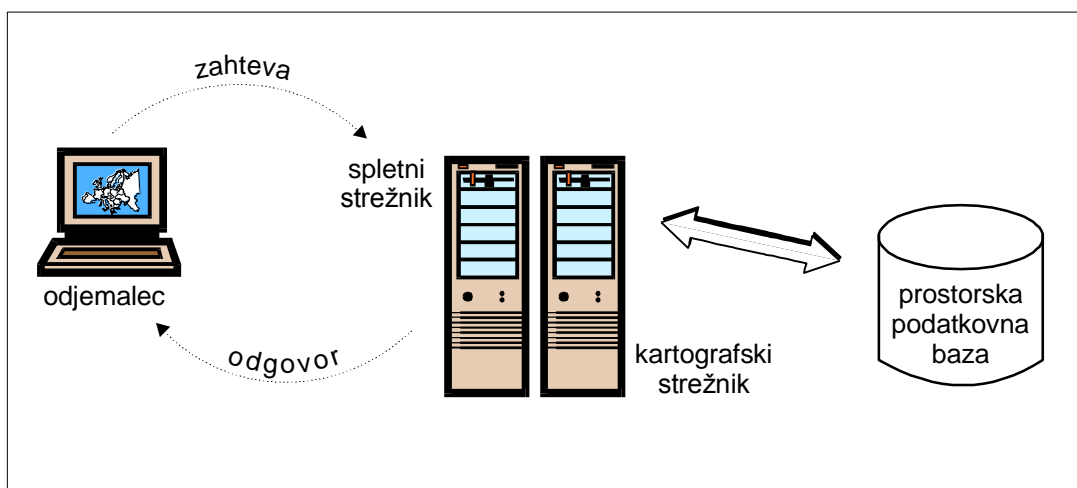
Internet je na novo definiral način uporabe kart. S tem ko karte niso več omejene le na fizični medij (papir, plastika, blago, kovina, idr.), je postala distribucija kart do uporabnika hitrejša, karte se pogosteje posodablja (npr. vremenske karte celo večkrat na dan) in kar je najpomembneje, karte so v primerjavi s klasičnimi bolj interaktivne. Skonstruirane so

lahko tako, da omogočajo interakcijo s spletno podatkovno bazo in tako končnemu uporabniku ponujajo višji nivo uporabnosti. Internet omogoča tudi lažjo distribucijo različnih oblik kartografskih prikazov kot so npr. animacije. Skratka - internet končnemu uporabniku predstavlja hitrejši način distribucije kart in možnost različnih oblik interakcij med človekom in karto (Peterson, 2005).

Na drugi strani pa je internet prinesel tudi nekaj negotovosti na področje kartografije. Ljudje elektronske karte dojemajo na drugačen način kot papirnate in razliko v percepciji je zato treba vzeti v obzir med procesom oblikovanja. Poleg tega ima tradicionalna kartografija precej bolj izpopolnjen kartografski jezik, saj se je le ta lahko razvijal preko stoletij, medtem, ko je spletna kartografija precej mlajša in ustrezen jezik za spletno kartografijo se še vedno razvija. Zaradi same narave interneta, ki ga uporabljajo različni uporabniki z različnimi sistemskimi nastavitvami (ločljivost zaslona, barvne nastavitve, idr.) pa kartograf prav tako nima več popolne kontrole nad končnim rezultatom, saj ne more z gotovostjo vedeti kaj točno bo uporabnik karte videl na svojem zaslonu (Gillavry, 2000).

Komunikacija odjemalec/strežnik

Način uporabe in dostopa do spletnih kart je različen. Danes še vedno daleč najbolj razširjen način uporabe spletnih kart je zgolj pregledovanje rastrskih kart, kjer je interakcija uporabnika omejena na povečavo, premikanje in morebiti na izbiro prikazanih slojev. Takšne karte so ponavadi predhodno pripravljene in spletni strežnik jih enostavno posreduje odjemalcu v obliki slikovnih datotek. Pri tem ne potrebuje dodatne programske opreme za strežbo kart. Drugi način (ki ga opisujemo v nadaljevanju) je dinamična strežba, pri kateri se karte sestavljajo »sproti« – v realnem času. Do teh kart ponavadi dostopamo tako, da uporabnik spletnemu strežniku preko svojega spletnega brskalnika pošlje zahtevo po karti. Spletni strežnik zahtevo sprejme ter jo posreduje naprej ustrezni GIS/kartografski programski opremi (kartografski strežnik). Ta na osnovi zahteve sestavi karto in jo vrne v vektorskem (GML, SVG, idr.) oz. rastrskem (JPG, GIF, idr.) formatu. Spletni strežnik nato karto posreduje odjemalcu v obliki spletne strani (slika 7).



Slika 7: Komunikacija odjemalec/strežnik.

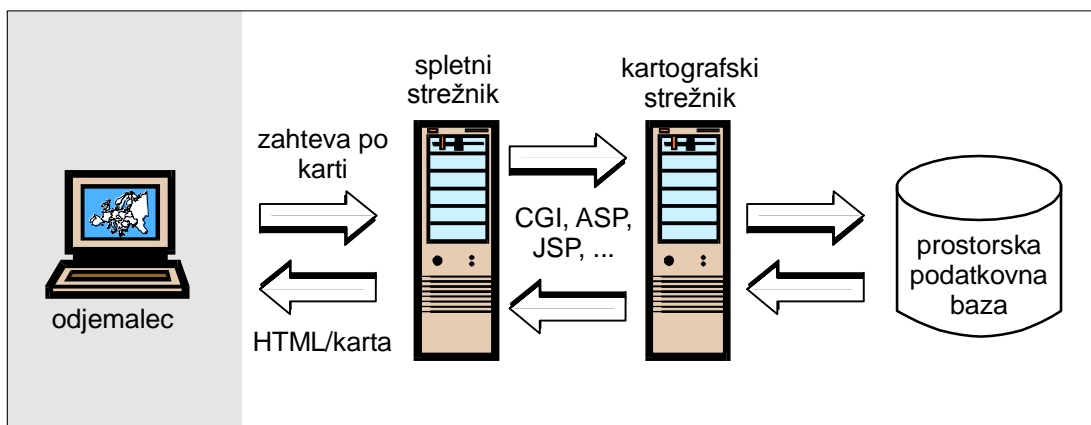
Geoproceniranje se razdeli med odjemalčevo stran, ki ponavadi sestoji iz spletnega brskalnika in na strežniško stran, ki sestoji iz spletnega strežnika, kartografskega strežnika in prostorske podatkovne baze. Glede na to, koliko procesiranja je na odjemalčevi oz. strežniški strani razlikujemo različne izvedbene pristope. V GIS in kartografiji so pomembni predvsem naslednji trije:

- strežniški pristop (celotno procesiranje se izvaja na strani strežnika),
- odjemalski pristop (del procesiranja se prenese na odjemalca),
- hibridne strategije (kombinacije strežniškega in odjemalskega pristopa).

Strežniški pristop

Pri strežniškem pristopu odjemalec potrebuje le spletni brskalnik, preko katerega komunicira s strežnikom in s katerim nato prikazuje vrnjene HTML datoteke. Vso procesiranje se izvaja na strežniški strani. Dobra stran strežniškega pristopa je, da je vsa kompleksna programska oprema in prostorska podatkovna baza shranjena na enem strežniku oz. na sistemu povezanih strežnikov ter je tako centralizirana. Upravljalce podatkov tako lažje upravlja s sistemom, poleg tega s tem tudi bistveno pocenimo razvoj in vzdrževanje sistema. Podatki, ki jih spletni strežnik vrne, so v standardnem HTML formatu, ki ga lahko prikaže katerikoli spletni brskalnik. To pa ima bistven vpliv tudi na učinkovitost in zanesljivost ter hkrati pritegne velik krog uporabnikov. Slaba stran strežniškega pristopa je v tem, da je uporabnik omejen le na pregledovanje karte in nima

dodatnih GIS funkcionalnosti (Trinidad et al, 2000). Slabost pa je tudi v hitrosti prenosa pri obsežnejših rastrskih že izdelanih upodobitvah.



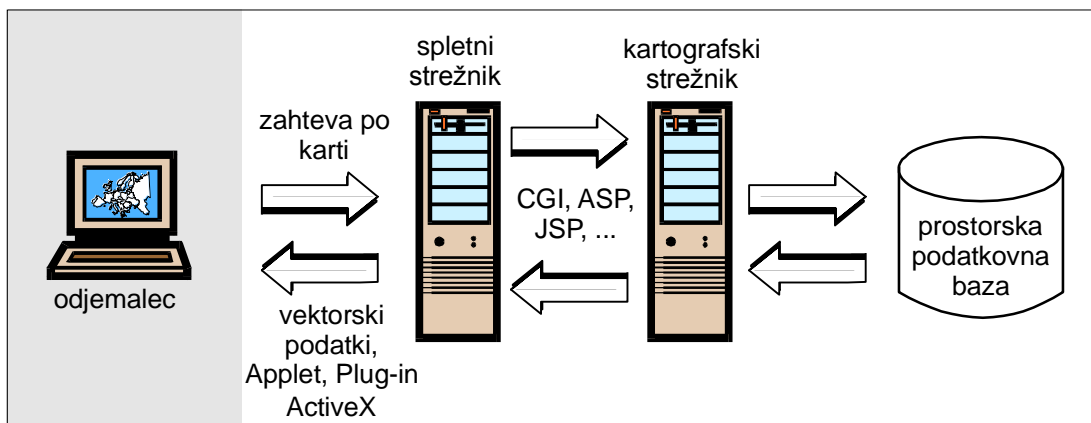
Slika 8: Komunikacija odjemalec/strežnik: strežniški pristop.

Odjemalski pristop

Pri odjemalskem pristopu se del geoprocesiranja izvede na strani odjemalca. Namesto, da vsa opravila izvede strežnik, se določeni GIS postopki naložijo na odjemalca, kjer se izbrani prostorski podatki lahko obdelujejo lokalno. Obstajata dve različici odjemalskega pristopa: posredovanje GIS programčkov odjemalcu in stalni GIS programčki ter dodatki na odjemalcu. V prvem primeru se na podlagi zahteve po podatkih, ki jo uporabnik sproži v spletnem brskalniku, odjemalcu posreduje izbrane podatke ter ustrezne programčke (*Java applet*, *ActiveX*, idr.). Na osnovi teh potem odjemalec sam izdela prostorsko analizo podatkov. Drugi pristop pa temelji na trajnem dodajanju GIS funkcionalnosti neposredno v odjemalčev brskalniki. To se lahko naredi bodisi z prednaložitvijo dodatkov (*plug-ins*) v odjemalčev brskalniki oz. z namestitvijo nove trajne aplikacije na strani odjemalca (Šumrada, 2005b).

Dobra stran odjemalskega pristopa je, da za razliko od strežniškega pristopa odjemalec prejme vektorske podatke, ki so mnogo bolj kompaktni in na katerih je mogoče lokalno izvajati različne GIS operacije. Težave pa lahko nastopijo pri distribuciji podatkov in programčkov. Nobena programska oprema (niti *Java*) ne deluje v vseh računalniških okoljih. Zaradi tega lahko izgubimo nekatere potencialne uporabnike, ki nimajo združljivih računalniških platform. Poleg tega je posredovanje programčkov preko interneta lahko

zamudno. Uporabnik želi dostop do informacij takoj, ko vstopi na spletno stran in mnoge uporabnike bo zamudno čakanje na rezultate poizvedbe odvrnilo od naše storitve (Trinidad et al, 2000).



Slika 9: Komunikacija odjemalec/strežnik: odjemalski pristop.

5.5 Oblikovanje spletne karte

Pri oblikovanju spletne karte veljajo podobna pravila kot pri oblikovanju klasične analogne karte z nekaterimi razlikami zaradi same narave medija, preko katerega je karta distribuirana. Digitalna karta je izdelana na osnovi podatkov prostorske podatkovne baze. Podatki v bazi so predstavljeni s tremi osnovnimi abstraktnimi oblikami oz. grafičnimi objektnimi tipi: s točko, linijo in ploskvijo, načeloma pa so lahko tudi 3R. Vsak pojav je ponavadi opisan tudi z enim ali več atributi. Grafika opredeljuje položaj in prostorsko obliko pojava, medtem ko atribut podaja njegove opisne in časovne lastnosti. Na podlagi sorodnih atributov pojave združujemo v razrede. S postopki kartografskega oblikovanja bomo pojave nekega razreda na karti predstavili s specifično obliko, ki jih bo ločevala od pojavov drugega razreda. Kartografsko oblikovanje temelji na teoriji kartografskih izraznih sredstev, grafičnih spremenljivk, kartografske generalizacije ter matematičnih elementov karte.

Merilo karte in kartografska projekcija

Merilo karte in kartografska projekcija sodita med matematične elemente karte. Pojem »merilo« se običajno nanaša na razmerje dveh količin, podanih v istih merskih enotah

(Peterca, 2001). Merilo karte tako označuje razmerje med velikostjo pojava v naravi in velikostjo istega pojava, prikazanega na karti. Merilo je eden izmed najvažnejših podatkov na karti in od njega je med drugim odvisen tudi način prikaza posameznih pojavov na karti. Pri manjših merilih se na karti prikazujejo le najpomembnejši pojavi, ki so ustrezno poenostavljeni. Večje kot je merilo, bolj podrobna je vsebina in kompleksnost prikaza. Pojem merila karte ima pri digitalni interaktivni karti drugačen pomen kot pri klasični analogni karti. Merilo prikaza ni več fiksno, ampak se le to lahko glede na uporabniško zahtevo spreminja, še vedno pa določa stopnjo podrobnosti vsebine karte, ki s tem posledično tudi omejuje smiselni razpon meril prikaza. Naloga kartografa je, da za posamezne razpone meril definira vsebino in obliko kartografskega prikaza. Merilo je na karti lahko predstavljeno na opisni, številčni ali grafični način.

Ker želimo ukrivljeno trirazsežno površje Zemlje prikazati na računalniškem zaslonu, t.j. v ravnini, je potrebno izvesti ustrezno preslikavo. Preslikavo elipsoida v ravnino obravnavajo kartografske projekcije. Cilj preslikave je transformacija trirazsežne površine zemeljskega elipsoida v njeno dvorazsežno sliko na projekcijski ravnini. Zaradi metričnih lastnosti ploskve elipsoida pa preslikava v ravnino ni možna brez deformacij – raztegov in krčenj (Peterca, 2001). Pri preslikovanju pride do deformacije kotov, dolžin in površine in tako govorimo o konformnih, ekvidistantnih in ekvivalentnih projekcijah. Preslikava iz elipsoida v ravnino je lahko direktna ali s posredovanjem konusnih in cilindričnih odvodnih ploskev, ki se potem razvijejo v ravnino brez deformacij. Po tem kriteriju se projekcije členijo na konusne, polikonusne, psevdokonusne, cilindrične, psevdocilindrične, azimutne, psevdoazimutne in pogojne (Peterca, 2001). Izbira kartografske projekcije je odvisna predvsem od namena karte (katere deformacije so dopustne) in od območja prikaza.

Kartografska izrazna sredstva

Kartografski znaki, napisi in grafikoni spadajo med kartografska izrazna sredstva. To so grafične kode, ki nam služijo za predstavitev podatkov na karti. Kartografske znake lahko glede na razsežnost prikazanega pojava delimo na točkovne, linijske in ploskovne.

Točkovni kartografski znaki

Točkovni kartografski znaki so uporabni za prikazovanje posameznih objektov in pojavov, ki jih na karti ni mogoče prikazati v ustreznem merilu. S svojo obliko in velikostjo zasedejo določen del prostora na karti, vendar se nanašajo na točno določeno mesto, saj označujejo objekte, katerih položaj je v naravi nedvoumno določen (npr. naselja, gradove, železniške postaje, idr.). Poleg informacij o položaju nam oblike točkovnih kartografskih znakov nudijo predstavo o vrstah objektov, njihova velikost pa nakazuje kvantitativne vrednosti (Fridl, 1999). Točkovne kartografske znake delimo na geometrične (geometrični liki, ki praviloma niso asociativni), nazorne (asociativni znaki) ter črkovno-številčne (znaki v obliki števil in črk).

Linijski kartografski znaki

Linijski kartografski znaki nam služijo za prikaz različnih linijskih objektov in pojavov (ceste, železnice, reke, idr.). Linijski objekti in pojavi so prikazani v eni razsežnosti ter v ustreznem merilu - to velja za dolžino linijskega objekta oz. pojava, medtem ko je širina objekta oz. pojava na karti vedno prikazana pretirano.

Ploskovni kartografski znaki

S ploskovnimi prostorskimi znaki prikazujemo objekte in pojave, ki zasedajo določene površine. To so lahko npr. različni socialno-ekonomski objekti in pojavi (administrativne enote, stavbe, idr.) ali naravni objekti in pojavi (morja, jezera, gozdovi, idr.). Objekt oz. pojav je prikazan v pravi obliki, v določenem merilu.









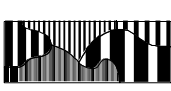






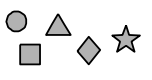

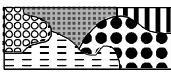
Napisi

Napisi so sestavni del vsake karte in na topografskih kartah pojasnjujejo številne objekte, ki jim pripadajo. Njihova osrednja vloga je povečati informativno vrednost karte, vendar od preostale vsebine ne smejo izstopati ter preusmerjati pozornosti uporabnika nase. Napisi, ki sodijo k posameznim prikazom objektov, so najpogosteje zemljepisna imena, čeprav lahko zaobjamejo tudi kakršnekoli druge tekstovne razlage (Fridl, 1999). Oblika napisa mora biti povezljiva z objektom na katerega se nanaša. Imena, ki se nanašajo na linijske pojave, morajo biti npr. izpisana vzporedno z linijskim pojavom, imena, ki se

nanašajo na površine, morajo zaobjeti čimvečji del referenčne površine, najboljša možna postavitev napisa k točkovnemu znaku je desno od znaka.

Grafične spremenljivke:

Kartografske znake lahko ustrezno oblikujemo s pomočjo grafičnih spremenljivk. S spreminjanjem barve, velikosti, svetlobne vrednosti, oblike, smeri ali vzorca, lahko vsakemu kartografskemu znaku določimo specifično obliko. Grafične spremenljivke opisuje slika 10.

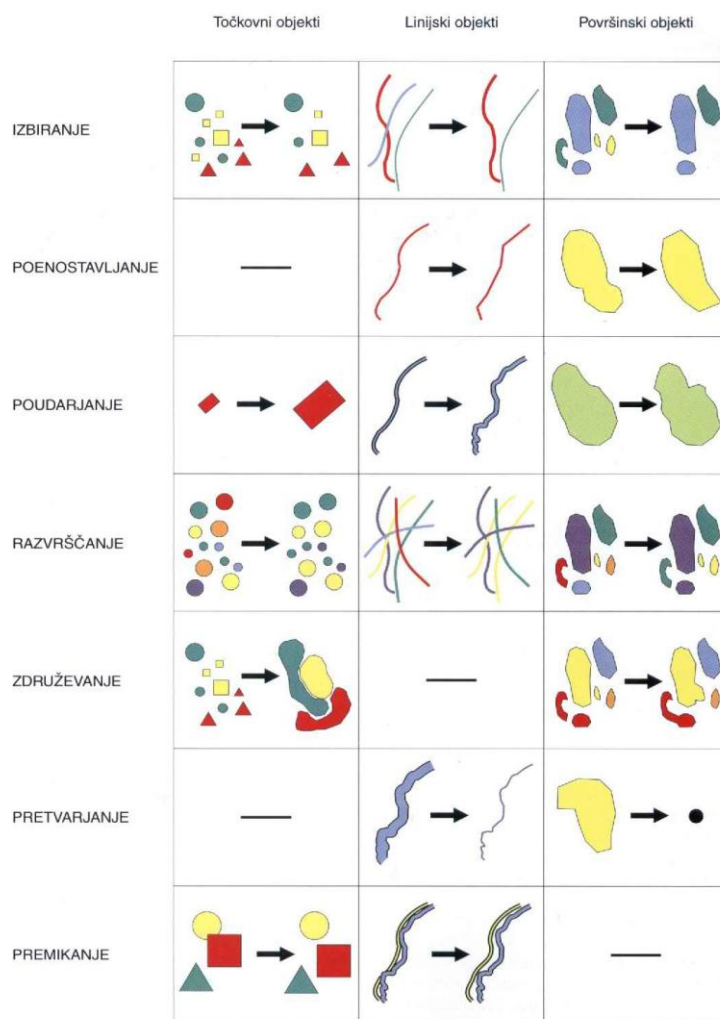
<i>točka</i>	<i>linija</i>	<i>območje</i>	
			<i>velikost</i>
			<i>svetlobna vrednost</i>
			<i>vzorec</i>
			<i>barva</i>
			<i>smer</i>
			<i>oblika</i>

Slika 10: Grafične spremenljivke (vir: Internet 6).

Kartografska generalizacija:

Karta lahko z manjšanjem merila (ob ohranitvi količine prikazanih podatkov) hitro postane prenasočena z vsebino in s tem nepregledna ter nerazumljiva. Količino prikazanih pojavov ter kompleksnost prikaza je zato potrebno ustrezno prilagajati merilu. Manj pomembne pojave je potrebno izpuščati oz. poenostavljati, pomembnejše pojave na drugi strani poudarjati. Vse to počnemo s postopki kartografske generalizacije. Prikazani so na sliki 11. Kartografska generalizacija se je z uporabo računalniške tehnologije precej poenostavila. Nekatere postopke kartografske generalizacije lahko izvedemo v zelo kratkem času, z

majhnimi stroški in nad veliko količino podatkov. Avtomatizirani so predvsem postopki geometričnega poenostavljanja. Kljub vsemu pa programska oprema, ki bi omogočala popolnoma avtomatizirano generalizacijo, ne obstaja in verjetno tudi nikoli ne bo (Radovan, 2003). Kartografska generalizacija je namreč preveč kompleksna, da bi jo bilo mogoče popolnoma avtomatizirati in izkušnje in znanje kartografa so pri tem delu še vedno odločilne.



Slika 11: Postopki kartografske generalizacije (vir: Fridl, 1999).

5.6 Tipi aplikacij za spletno kartografijo

Razlikujemo med različnimi tipi aplikacij za spletno kartografijo. Nekatere aplikacije so namenjene predvsem prikazovanju statičnih oz. dinamičnih kart – pri teh je poudarek

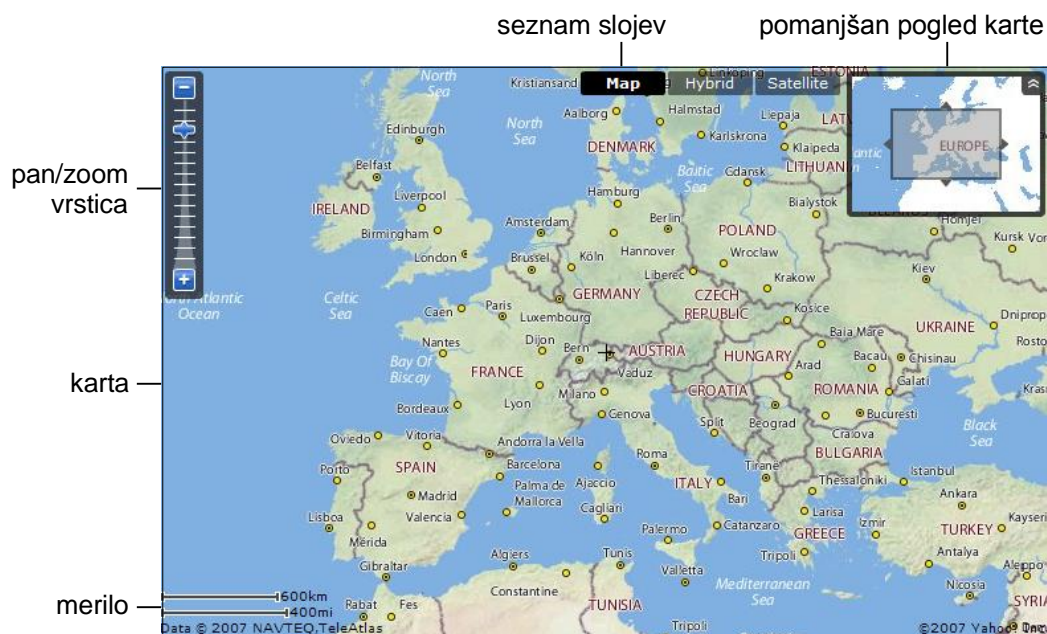
predvsem na dobri kartografski upodobitvi. Druge, ki se npr. bolj posvečajo različnim GIS analizam imajo ponavadi slabšo kartografsko upodobitev, itd. Tip aplikacije za strežbo spletnih kart izbiramo glede na namen uporabe karte. Aplikacije lahko v splošnem razdelimo v naslednje kategorije (Hächler, 2003 po Gartner 1999):

- Prostorski podatkovni strežnik ponuja dostop in iskanje po bazi prostorskih podatkov, omogoča tudi prenos podatkov za lokalno procesiranje na strani odjemalca. Ni interaktiven, ni povezljivosti s podatkovno bazo, kartografska upodobitev je slaba. Primer takšnega strežnika najdemo na spletni strani *U.S. Geological Survey*: <http://www.usgs.gov>.
- Kartografski strežnik na zahtevo odjemalca streže statične ali interaktivne karte.
 - Statični kartografski strežnik streže karte v obliki slikovnih datotek, ki so bile že predhodno pripravljene (takšne, ki smo jih že prej npr. izvozili iz namiznega GIS orodja).
 - Upodobitveni kartografski strežnik streže karte, ki se generirajo sproti, glede na trenutno zahtevo odjemalca. Funkcionalnost je skoncentrirana na kartografsko upodobitev, dodatnih GIS funkcionalnosti nima.
 - Interaktivni kartografski strežnik omogoča odjemalcu spremembo nekaterih parametrov na karti, ponuja različne atributne in prostorske poizvedbe.
- Spletni kartografski informacijski sistem – poudarek je na različnih tematskih in prostorskih poizvedbah, ki jih izvajamo interaktivno preko tekstovnega polja. Primer takšnih sistemov so različni načrtovalci poti, ki omogočajo izračun najhitrejših/najkrajših poti od točke A do točke B.
- Spletni GIS je interaktiven, posveča se predvsem GIS analizam in ne toliko sami kartografski upodobitvi.
- Strežnik z GIS funkcionalnostmi omogoča oddaljen dostop do funkcij GIS strežnika. Odjemalec pošlje zahtevo in podatke, ki se sprocesirajo na strežniški strani, rezultate (ki niso nujno karta) strežnik nato pošlje nazaj odjemalcu.

5.7 Uporabniški vmesnik spletne karte

V splošnem naj bi uporabniški vmesnik spletne kartografske aplikacije sestavljale naslednje osnovne sestavine (Schütze, 2007):

- Karta: glavna sestavina vsake aplikacije, vsebuje enega ali več rastrskih ali vektorskih podatkovnih slojev.
- Pomanjšan pregled karte (*map overview*): v manjšem navigacijskem oknu prikazuje glavno karto v pomanjšanem merilu, vidi se celotno območje prikaza ter položaj trenutnega pogleda na glavni karti, uporabnik se tako na karti lažje orientira.
- *Pan/zoom* vrstica: vključuje puščice za navigacijo po karti ter drsni trak, ki nam omogoča spreminjanje merila karte (funkcija je lahko omogočena tudi na drugačen način).
- Lista podatkovnih slojev (*layer overview*): seznam vseh podatkovnih slojev z opcijo vklapljanja in izklapljanja posameznih slojev.
- Orodna vrstica: vsebuje gumbe za aktivacijo različnih GIS funkcij (navigacija, analiza, poizvedovanja, tiskanje, idr.).
- Merilo: grafično ali numerično prikazuje merilo trenutnega pogleda.
- Legenda: razlaga barve in simbole, ki so bili uporabljeni pri oblikovanju karte (pogosto se uporablja v kombinaciji z listo slojev).



Slika 12: Nekaj tipičnih sestavin spletne karte na primeru »Yahoo Maps«.

Navedene osnovne sestavine karte so lahko na zaslonu prikazane vse hkrati, kar je pogosto problematično zaradi zelo omejenega formata karte ali pa se določene vsebine prikazujejo na zahtevo.

6 »STYLED LAYER DESCRIPTOR« (SLD)

6.1 Uvod

Prostorski podatki, ki jih hranimo v bazi podatkov vsebujejo lokacijske in opisne podatke. V kartografskem smislu pa so ti podatki do neke mere še vedno surovi in za končnega uporabnika karte omejeno uporabni. Podatke želimo ustrezno predstaviti, v obliki karte. Za ustrezno kartografsko predstavitev podatkov pa potrebujemo določena orodja, ki nam to omogočajo.

SLD (*Styled Layer Descriptor*) je specifikacija, ki jo je razvilo združenje OGC (*Open Geospatial Consortium*). Definira XML (*Extensible Markup Language*) jezik, ki kartografu omogoča, da prostorske podatke ustrezno oblikuje in jih na koncu predstavi uporabniku kot tematsko karto. Podatkovne sloje oblikujemo tako, da določamo barve, debeline in oblike linij, šrafure, idr. S tem dobimo stilizirane sloje (*Styled Layers*), ki si jih lahko predstavljamo kot sloje z določenimi tematskimi vsebinami (npr. hidrologija, ceste, objekti). S prekrivanjem stiliziranih slojev v določenem zaporedju v končni fazi dobimo tematsko karto.

Specifikacija SLD določa slovnico za oblikovanje prostorskih podatkov. Navodila o tem, kako naj bodo posamezni sloji oblikovani zapišemo v SLD datoteke – kodirane besedilne datoteke, ki so razumljive tako ljudem kot tudi ustrezni programski opremi. SLD lahko tako zapišemo z navadnim urejevalnikom besedila in za to ne potrebujemo drage programske opreme. Za vsak posamezen podatkovni sloj, ki ga želimo oblikovati, izdelamo SLD datoteko, ki jo nato posredujemo kartografskemu strežniku. Ta na osnovi navodil, katera SLD datoteka se uporablja v kombinaciji s katerim podatkovnim slojem, izdela ustrezno oblikovane sloje. To je najbolj osnovni način uporabe SLD in ga opisujemo v nadaljevanju.

6.2 Zgradba SLD datoteke

Preprost SLD je sestavljen iz glave ter iz niza pravil (*rules*). V glavi so napisane nekatere formalne opredelitve, kot so npr. verzija SLD specifikacije, uporabljena kodna tabela, idr.

V pravilih pa definiramo ustrezne oblikovalske sloge. Za vsako skupino pojavov, ki bo imela enak slog, je potrebno izdelati svoje pravilo. Vsako pravilo sestavlja:

- filter,
- simbolizator (*symbolizer*),
- razpon merila (*MinScaleDenominator*, *MaxScaleDenominator*).

S filtrom najprej opredelimo, na katere pojave podatkovnega sloja se bo nanašal določen oblikovalski slog (npr. želimo, da se slog nanaša na vse avtoceste in hitre ceste iz podatkovnega sloja cestnega omrežja). S simbolizatorjem nato določimo oblikovalski slog (barva, debelina linije, idr.), z razponom merila pa definiramo, za katera merila naj omenjeno pravilo velja. Slika 13 prikazuje zgradbo preprostega SLD-ja.

```
<?xml version="1.0" encoding="UTF-8"?>
.
.
.

<Rule>
  <Filter>
    ... avtocesta, hitra cesta ...
  </Filter>

  <LineSymbolizer>
    ... rumena barva, debelina 2 px ...
  </LineSymbolizer>

  <MinScaleDenominator>... 1:10000...</MinScaleDenominator>
  <MaxScaleDenominator>... 1:5000 ...</MaxScaleDenominator>
</Rule>

<Rule>
  <Filter>
    ... mestna cesta ...
  </Filter>

  <LineSymbolizer>
    ... siva barva, debelina 1 px ...
  </LineSymbolizer>

  <MinScaleDenominator>... 1:10000...</MinScaleDenominator>
  <MaxScaleDenominator>... 1:5000...</MaxScaleDenominator>
</Rule>
```

Slika 13: Prikaz zgradbe preprostega SLD-ja.

6.3 Filter

Znotraj elementa *<Filter>* na podlagi opisnih atributov določimo selekcijo pojavov, za katere bo veljalo določeno pravilo. V primeru, da filtra ne definiramo, to pomeni, da bo pravilo veljalo za vse podatke določenega podatkovnega sloja.

Filter je ponavadi sestavljen iz kombinacije izrazov. Tri različne skupine operatorjev, ki se uporabljajo pri filtriranju so:

- primerjalni operatorji (*PropertyIsEqualTo* (=), *PropertyIsGreaterThan* (>), idr.),
- prostorski operatorji (*Overlaps* - prekriva, *Touches* - se dotika, idr.),
- logični operatorji (*and*, *or*, *not*)

S primerjalnimi operatorji, kot pove že ime, primerjamo med sabo dve vrednosti – testiramo, ali vrednost atributa zadostuje postavljenemu pogoju. Pri prostorskih operatorjih je podobno, le da tu primerjamo prostorski odnos med dvema pojavoma (npr. ali je določen geometrijski pojav znotraj referenčnega območja). Logični operatorji se uporabljajo za povezovanje različnih primerjalnih oz. prostorskih operatorjev v en izraz.

Recimo, da imamo podatkovni sloj mest, kjer ima vsak podatek numerični atribut »*population*«, ki označuje število prebivalcev ter atribut »*capital*«, ki označuje ali je to mesto glavno mesto države (1 – je glavno mesto, 0 – ni glavno mesto). Želimo izbrati le glavna mesta z več kot 500.000 prebivalci. Zapišemo filter:

```
<Filter>
  <And>
    <PropertyIsEqualTo>
      <PropertyName>capital</PropertyName>
      <Literal>1</Literal>
    </PropertyIsEqualTo>

    <PropertyIsGreaterThan>
      <PropertyName>population</PropertyName>
      <Literal>500000</Literal>
    </PropertyIsGreaterThan>
  </And>
</Filter>
```

Slika 14: »*Styled Layer Descriptor*« - primer uporabe filtra.

6.4 Simbolizatorji (»*Symbolizers*«)

S simbolizatorji definiramo oblikovalske sloge oz. način, kako se bodo določeni pojavi prikazovali na karti. Določamo barve, debeline in oblike linij, šrafure, idr. Glede na tip pojava, ki ga oblikujemo, ločimo različne vrste simbolizatorjev:

- linijski simbolizator <*LineStyleSymbolizer*> – za oblikovanje linijskih pojavov,
- poligonski simbolizator <*PolygonSymbolizer*> – za oblikovanje ploskovnih pojavov,
- točkovni simbolizator <*PointSymbolizer*> – za oblikovanje točkovnih pojavov,
- tekstovni simbolizator <*TextSymbolizer*> – za oblikovanje teksta,
- rastrski simbolizator <*RasterSymbolizer*> – za oblikovanje rastrskih podatkov


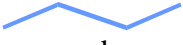





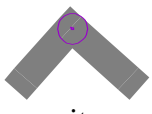

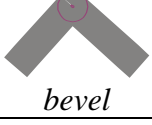


Linijski simbolizator (»*LineStyleSymbolizer*«)

Z linijskim simbolizatorjem določamo obliko linijskih pojavov. Osnovne oblike, ki jih lahko dodelimo neki liniji, opredelimo znotraj elementa <*Stroke*>. Opredelimo barvo linije, prosojnost, debelino, obliko zaključkov linije, obliko spojev linije, črtkan vzorec. Slika 15 prikazuje strukturo zapisa linijskega simbolizatorja, tabela 1 pa konkretne primere uporabe elementa <*Stroke*>.

```
<LineStyleSymbolizer>  
  <Stroke>  
    <stroke>#FFFFFF</stroke>  
    <stroke-opacity>1.0</stroke-opacity>  
    <stroke-weight>3.0</stroke-weight>  
    <stroke-linejoin>mitre</stroke-linejoin>  
    <stroke-linecap>butt</stroke-linecap>  
    <stroke-dasharray>0.0</stroke-dasharray>  
    <stroke-dashoffset>0.0</stroke-dashoffset>  
  </Stroke>  
</LineStyleSymbolizer>
```

Slika 15: Struktura zapisa linijskega simbolizatorja.

Tabela 1: Primeri uporabe elementa <Stroke>.

Oblikovni element	Primer SLD sintakse	Grafični primer
Barva (<i>stroke</i>)	<pre><sld:CssParameter name="stroke"> *<ogc:Literal>#FF6600</ogc:Literal> </sld:CssParameter></pre> <pre><sld:CssParameter name="stroke"> <ogc:Literal>#6699FF</ogc:Literal> </sld:CssParameter></pre>	 <p>oranžna (#FF6600)</p>  <p>modra (#6699FF)</p>
Debelina (<i>stroke-width</i>)	<pre><sld:CssParameter name="stroke-width"> <ogc:Literal>4.0</ogc:Literal> </sld:CssParameter></pre> <pre><sld:CssParameter name="stroke-width"> <ogc:Literal>2.0</ogc:Literal> </sld:CssParameter></pre>	 <p>4 px</p>  <p>2 px</p>
Oblika zaključka linije (<i>stroke-linecap</i>)	<pre><sld:CssParameter name="stroke-linecap"> <ogc:Literal>butt</ogc:Literal> </sld:CssParameter></pre> <pre><sld:CssParameter name="stroke-linecap"> <ogc:Literal>round</ogc:Literal> </sld:CssParameter></pre> <pre><sld:CssParameter name="stroke-linecap"> <ogc:Literal>square</ogc:Literal> </sld:CssParameter></pre>	 <p><i>butt</i></p>  <p><i>round</i></p>  <p><i>square</i></p>
Oblika spojev linije (<i>stroke-linejoin</i>)	<pre><sld:CssParameter name="stroke-linejoin"> <ogc:Literal>mitre</ogc:Literal> </sld:CssParameter></pre> <pre><sld:CssParameter name="stroke-linejoin"> <ogc:Literal>round</ogc:Literal> </sld:CssParameter></pre> <pre><sld:CssParameter name="stroke-linejoin"> <ogc:Literal>bevel</ogc:Literal> </sld:CssParameter></pre>	 <p><i>mitre</i></p>  <p><i>round</i></p>  <p><i>bevel</i></p>
Prosojnost linije (<i>stroke-opacity</i>)	<pre><sld:CssParameter name="stroke-opacity"> <ogc:Literal>0.8</ogc:Literal> </sld:CssParameter></pre> <pre><sld:CssParameter name="stroke-opacity"> <ogc:Literal>0.2</ogc:Literal> </sld:CssParameter></pre>	 <p>20% prosojnost</p>  <p>80% prosojnost</p>

<p>Črtkast vzorec <i>(stroke-dasharray)</i> in <i>(stroke-dashoffset)</i></p>	<pre><sld:CssParameter name="stroke-dashoffset"> <ogc:Literal>0.0</ogc:Literal> </sld:CssParameter> <sld:CssParameter name="stroke-dasharray"> 2.0 2.0 </sld:CssParameter> <sld:CssParameter name="stroke-dashoffset"> <ogc:Literal>2.0</ogc:Literal> </sld:CssParameter> <sld:CssParameter name="stroke-dasharray"> 4.0 4.0 </sld:CssParameter></pre>	<p style="text-align: center;">- - - -</p> <p>2 px polna črta, 2 px praznine, ofset = 0 px</p> <p style="text-align: center;">- - - -</p> <p>4 px polna črta, 4 px praznine, ofset = 2 px</p>
--	---	---

*Pri določevanju barve moramo podati zapis v RGB načinu, pretvorjenem v heksadecimalni sistem.

Točkovni simbolizator (»*PointSymbolizer*«)

S točkovnim simbolizatorjem določamo obliko točkovnih pojavov. Osnovne oblike, ki jih lahko dodelimo točkovnemu pojavu, opredelimo znotraj elementa *<Graphic>*. Točkovni pojav lahko prikažemo na dva načina:

- z geometrijskim likom (kvadrat, krog, trikotnik, zvezda, križ, x) – definiramo ga znotraj elementa *<Mark>*,
- z grafično podobo (gif, png, svg, idr.) – to je »majhna slika« ki jo priključimo tako, da navedemo URL naslov, od koder se bo ta grafična podoba naložila – definiramo jo znotraj elementa *<ExternalGraphic>*.

Na sliki 16 je prikazana struktura zapisa točkovnega simbolizatorja. Prvi primer prikazuje zapis ob uporabi grafične podobe, drugi primer pa zapis ob uporabi geometrijskega lika.










```
<PointSymbolizer>
  <Graphic>
    <ExternalGraphic>
      <OnlineResource>...image01.gif</OnlineResource>
      <format>image/gif</format>
    </ExternalGraphic>
    <Opacity>1.0</Opacity>
    <Size>30</Size>
    <Rotation>0.0</Rotation>
  </Graphic>
</PointSymbolizer>

<PointSymbolizer>
  <Graphic>
    <Mark>
      <WellKnownName>circle</WellKnownName>
    </Mark>
    <Opacity>1.0</Opacity>
    <Size>30</Size>
    <Rotation>0.0</Rotation>
  </Graphic>
</PointSymbolizer>
```

Slika 16: Struktura zapisa točkovnega simbolizatorja.

Točkovne pojave lahko torej prikažemo v obliki grafične podobe ali geometrijskega lika. Poleg tega lahko točkovnemu pojavu določimo tudi velikost ter prosojnost, lahko ga zasučemo v poljubni smeri. V tabeli 2 so prikazani primeri uporabe elementa *<Graphic>*.

Tabela 2: Primeri uporabe elementa <Graphic>.

Oblikovni element	Primer SLD sintakse	Grafični primer
Način prikaza točkovega pojava	<pre> <sld:ExternalGraphic> <sld:OnlineResource xlink:type="simple" xlink:href="...image/airport.gif"/> <sld:Format>image/gif</sld:Format> </sld:ExternalGraphic> <sld:Mark> <sld:WellKnownName> star </sld:WellKnownName> * <sld:Fill> <sld:CssParameter name="fill"> <ogc:Literal>#808080</ogc:Literal> </sld:CssParameter> <sld:CssParameter name="fill-opacity"> <ogc:Literal>1.0</ogc:Literal> </sld:CssParameter> </sld:Fill> </sld:Mark> </pre>	 <p>grafična podoba (<i>ExternalGraphic</i>)</p>  <p>geometrijski lik (<i>Mark</i>)</p>
Velikost (<i>Size</i>)	<pre> <sld:Size> <ogc:Literal>12.0</ogc:Literal> </sld:Size> <sld:Size> <ogc:Literal>6.0</ogc:Literal> </sld:Size> </pre>	 <p>12 px</p>  <p>6 px</p>
Zasuk (<i>Rotation</i>)	<pre> <sld:Rotation> <ogc:Literal>0.0</ogc:Literal> </sld:Rotation> <sld:Rotation> <ogc:Literal>180.0</ogc:Literal> </sld:Rotation> </pre>	 <p>brez zasuca</p>  <p>zasuk za 180°</p>
Prosojnost (<i>Opacity</i>)	<pre> <sld:Opacity> <ogc:Literal>0.8</ogc:Literal> </sld:Opacity> <sld:Opacity> <ogc:Literal>0.2</ogc:Literal> </sld:Opacity> </pre>	 <p>20% prosojnost</p>  <p>80% prosojnost</p>

* Znotraj samega elementa *<Mark>* lahko uporabimo element za zapolnitev ploskve *<Fill>*, ki ga opisujemo v nadaljevanju.

Poligonski simbolizator (»*PolygonSymbolizer*«)

S poligonskim simbolizatorjem (*PolygonSymbolizer*) določamo obliko ploskovnih pojavov. Obliko, ki jo lahko dodelimo ploskovnemu pojavu, opredelimo znotraj dveh elementov:

- *<Fill>* – zapolnitev ploskve,
- *<Stroke>* – obroba ploskve.

Element *<Stroke>* je bil opisan že zgoraj (tabela 1), zato v nadaljevanju opisujemo le element za zapolnitev ploskve *<Fill>*. Ploskev lahko zapolnimo na dva načina:

- zapolnitev z barvo – definiramo jo znotraj (pod)elementa *<Fill>*,
- zapolnitev z grafičnim vzorcem – definiramo ga znotraj elementa *<GraphicFill>*.

Na sliki 17 je prikazana struktura zapisa poligonskega simbolizatorja. Prvi primer prikazuje zapis pri zapolnitvi z barvo, drugi primer pa zapis pri zapolnitvi z grafičnim vzorcem. Pri zapolnitvi z grafičnim vzorcem ploskev zapolnimo s ponavljajočim grafičnim vzorcem, ki se ustvari na podlagi majhne slike, shranjeni v npr. *.gif datoteki. Takšen grafični vzorec opredelimo znotraj elementa *<Graphic>*, ki je že opisan zgoraj (tabela 2) v poglavju o točkovnem simbolizatorju.

```


<PolygonSymbolizer>
  <Fill>
    <Fill>#808080</Fill>
    <FillOpacity>1.0</FillOpacity>
  </Fill>
  <Stroke>
  ...
</Stroke>
</PolygonSymbolizer>




<PolygonSymbolizer>
  <Fill>
    <GraphicFill>
      <Graphic>
        <ExternalGraphic>
          <OnlineResource>...image01.png</OnlineResource>
          <format>image/png</format>
        </ExternalGraphic>
        <Opacity>1.0</Opacity>
        <Size>30</Size>
        <Rotation>0.0</Rotation>
      </Graphic>
    </GraphicFill>
    <FillOpacity>1.0</FillOpacity>
  </Fill>
  <Stroke>
  ...
</Stroke>
</PolygonSymbolizer>
    
```

Slika 17: Struktura zapisa poligonskega simbolizatorja.

Ploskev lahko zapolnimo z barvo ali grafičnim vzorcem, poleg tega ji lahko določamo tudi prosojnost. V tabeli 3 so prikazani primeri uporabe elementa *<Fill>*.

Tabela 3: Primeri uporabe elementa *<Fill>*.

Oblikovni element	Primer SLD sintakse	Grafični primer
Način zapolnitve ploskovnega pojava	<pre> <sld:CssParameter name="fill"> <ogc:Literal>#6699FF</ogc:Literal> </sld:CssParameter> <GraphicFill> <Graphic> <ExternalGraphic> <OnlineResource xlink:type="simple" xlink:href=".../circles.png"/> </pre>	 <p>zapolnitev z barvo (<i>Fill</i>)</p>

	<pre><Format>image/png</Format> </ExternalGraphic> <Opacity>1.0</Opacity> <Size>10</Size> <Rotation>0.0</Rotation> </Graphic> </GraphicFill> </Fill></pre>	 <p>zapolnitev z grafičnim vzorcem (<i>GraphicFill</i>)</p>
Prosojnost (<i>Opacity</i>)	<pre><sld:CssParameter name="fill-opacity"> <ogc:Literal>0.6</ogc:Literal> </sld:CssParameter> <sld:CssParameter name="fill-opacity"> <ogc:Literal>0.4</ogc:Literal> </sld:CssParameter></pre>	 <p>40% prosojnost</p>  <p>60% prosojnost</p>

Tekstovni simbolizator (»*TextSymbolizer*«)

S tekstovnim simbolizatorjem določamo obliko teksta oz. napisov na karti. Obliko, ki jo lahko dodelimo tekstu, opredelimo znotraj naslednjih elementov:

- *<Label>* – napis,
- ** – pisava,
- *<LabelPlacement>* – položaj napisa,
- *<Halo>* – koroniranje,
- *<Fill>* – zapolnitev.

Na sliki 18 je prikazana struktura zapisa tekstovnega simbolizatorja.

```
<TextSymbolizer>
  <Label>poi_name</Label>

  <Font>
    <font-family>Arial</font-family>
    <font-style>italic</font-style>
    <font-weight>bold</font-weight>
    <font-size>12</font-size>
  </Font>

  <Halo>
    <radius>3</radius>
    <fill>
      <fill>#FFFFFF</fill>
      <opacity>1.0</opacity>
    </fill>
  </Halo>

  <LabelPlacement>
    <PointPlacement>
      <AnchorPoint>
        <AnchorPointX>0.0</AnchorPointX>
        <AnchorPointY>0.0</AnchorPointY>
      </AnchorPoint>
      <Displacement>
        <DisplacementX>2.0</DisplacementX>
        <DisplacementY>1.0</DisplacementY>
      </Displacement>
      <Rotation>0.0</Rotation>
    </PointPlacement>
    <LinePlacement>
      <PerpendicularOffset>1.0</PerpendicularOffset>
    </LinePlacement>
  </LabelPlacement>

  <fill>
    <fill>#000000</fill>
    <opacity>1.0</opacity>
  </fill>
</TextSymbolizer>
```

Slika 18: Struktura zapisa tekstovnega simbolizatorja.

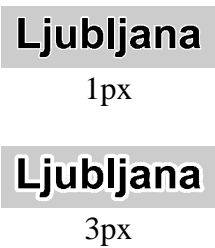
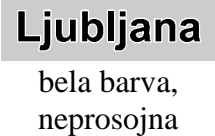
Pri oblikovanju teksta moramo najprej opredeliti vsebino napisov na karti (npr. poleg točke, ki označuje položaj mesta želimo izpisati tudi ime mesta). Znotraj elementa `<Label>` zato najprej navedemo ime polja atributne tabele, od koder se bo črpala vsebina napisov na karti. Znotraj elementa `` nato določimo obliko pisave – t.j. družino, stil, debelino in velikost pisave. Tabela 4 prikazuje primere uporabe elementa ``.

Tabela 4: Primeri uporabe elementa .

Oblikovni element	Primer SLD sintakse	Grafični primer
Pisava – družina (<i>Font-Family</i>)	<sld:CssParameter name="font-family"> <ogc:Literal>Arial</ogc:Literal> </sld:CssParameter>	Ljubljana <i>Arial</i>
	<sld:CssParameter name="font-family"> <ogc:Literal>Times New Roman</ogc:Literal> </sld:CssParameter>	Ljubljana <i>Times New Roman</i>
Pisava – stil (<i>Font-Style</i>)	<sld:CssParameter name="font-style"> <ogc:Literal>normal</ogc:Literal> </sld:CssParameter>	Ljubljana <i>normal</i>
	<sld:CssParameter name="font-style"> <ogc:Literal>italic</ogc:Literal> </sld:CssParameter>	Ljubljana <i>italic</i>
	<sld:CssParameter name="font-style"> <ogc:Literal>oblique</ogc:Literal> </sld:CssParameter>	Ljubljana <i>oblique</i>
Pisava – debelina (<i>Font-Weight</i>)	<sld:CssParameter name="font-weight"> <ogc:Literal>normal</ogc:Literal> </sld:CssParameter>	Ljubljana <i>normal</i>
	<sld:CssParameter name="font-weight"> <ogc:Literal>bold</ogc:Literal> </sld:CssParameter>	Ljubljana bold
Pisava – velikost (<i>Font-Size</i>)	<sld:CssParameter name="font-size"> <ogc:Literal>14</ogc:Literal> </sld:CssParameter>	Ljubljana 14 pik
	<sld:CssParameter name="font-size"> <ogc:Literal>20</ogc:Literal> </sld:CssParameter>	Ljubljana 20 pik

Za izboljšanje čitljivosti napisov na karti se napis lahko tudi koronira – doda se (ponavadi bela) obroba okrog napisa, ki ustvari dodaten kontrast med ozadjem in napisom in tako pripomore k boljši čitljivosti. Znotraj elementa <Halo> lahko spreminjamo velikost radija (t.j. debelino obrobe) in način zapolnitve. Tabela 5 prikazuje primere uporabe elementa <Halo>.

Tabela 5: Primeri uporabe elementa <Halo>.

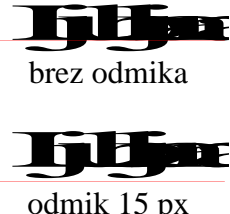
Oblikovni element	Primer SLD sintakse	Grafični primer
Radij (Radius)	<pre><sld:Radius> <ogc:Literal>1</ogc:Literal> </sld:Radius> <sld:Radius> <ogc:Literal>3</ogc:Literal> </sld:Radius></pre>	 <p>Ljubljana 1px</p> <p>Ljubljana 3px</p>
Zapolnitev (Fill)	<pre><sld:Fill> <sld:CssParameter name="fill"> <ogc:Literal>#FFFFFF</ogc:Literal> </sld:CssParameter> <sld:CssParameter name="fill-opacity"> <ogc:Literal>1.0</ogc:Literal> </sld:CssParameter> </sld:Fill></pre>	 <p>Ljubljana bela barva, neprosojna</p>

Zadnji element, ki ga opisujemo je element <LabelPlacement>. Ta nam omogoča, da postavimo napis v poljuben položaj glede na pojav, na katerega se ta napis nanaša. Npr. napis, ki označuje ime mesta, lahko leži levo, desno, zgoraj, spodaj od točke, ki označuje položaj mesta, lahko je glede na referenčno točko zarotiran. Element <LabelPlacement> vsebuje dva podelementa, znotraj katerih dejansko definiramo položaj napisa in sicer v odvisnosti od tega, ali je pojav, ki ga bomo poimenovali točka ali linija. Ta dva elementa sta:

- <LinePlacement> – omogoča nam postavitev napisa relativno glede na linijo,
- <PointPlacement> – omogoča nam postavitev napisa relativno glede na točko.

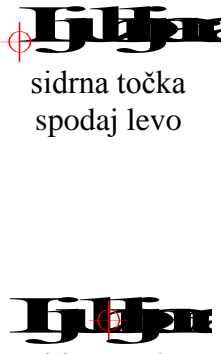
Znotraj elementa <LinePlacement> lahko določimo pravokotni odmik napisa od referenčne linije (Tabela 6).



Tabela 6: Primeri uporabe elementa <LinePlacement>.

Oblikovni element	Primer SLD sintakse	Grafični primer
Odmik (Perpendicular-Offset)	<pre><sld:PerpendicularOffset> <ogc:Literal>0.0</ogc:Literal> </sld:PerpendicularOffset> <sld:PerpendicularOffset> <ogc:Literal>15.0</ogc:Literal> </sld:PerpendicularOffset></pre>	 <p>Ljubljana brez odmika</p> <p>Ljubljana odmik 15 px</p>

Znotraj elementa *<PointPlacement>* lahko določimo sidrno točko, premik in zasuk napisa. Sidrna točka je točka znotraj napisa, ki se uporablja za sidranje točke na referenčno geometrijsko točko. S premikom določimo relativni odmik od referenčne točke po x in y osi. Tabela 7 prikazuje primere postavitve napisa relativno glede na točko.

Tabela 7: Primeri uporabe elementa *<PointPlacement>*.

Oblikovni element	Primer SLD sintakse	Grafični primer
Sidrna točka (<i>Anchor Point</i>)	<pre> <sld:AnchorPoint> <sld:AnchorPointX> <ogc:Literal>0.0</ogc:Literal> </sld:AnchorPointX> <sld:AnchorPointY> <ogc:Literal>0.0</ogc:Literal> </sld:AnchorPointY> </sld:AnchorPoint> <sld:AnchorPoint> <sld:AnchorPointX> <ogc:Literal>0.5</ogc:Literal> </sld:AnchorPointX> <sld:AnchorPointY> <ogc:Literal>0.5</ogc:Literal> </sld:AnchorPointY> </sld:AnchorPoint> </pre>	 <p>sidrna točka spodaj levo</p> <p>sidrna točka na sredini napisa</p>
Premik (<i>Displacement</i>)	<pre> <sld:Displacement> <sld:DisplacementX> <ogc:Literal>0.0</ogc:Literal> </sld:DisplacementX> <sld:DisplacementY> <ogc:Literal>0.0</ogc:Literal> </sld:DisplacementY> </sld:Displacement> <sld:Displacement> <sld:DisplacementX> <ogc:Literal>10.0</ogc:Literal> </sld:DisplacementX> <sld:DisplacementY> <ogc:Literal>-5.0</ogc:Literal> </sld:DisplacementY> </sld:Displacement> </pre>	 <p>brez premika</p> <p>premik: 10px po x, 5 px po y</p>

Zasuk (<i>Rotation</i>)	<pre><sld:Rotation> <ogc:Literal>0.0</ogc:Literal> </sld:Rotation> <sld:Rotation> <ogc:Literal>5.0</ogc:Literal> </sld:Rotation></pre>	 brez zasuk  zasuk za 5°
------------------------------	---	---

7 PREDSTAVITEV PROSTORSKE PODATKOVNE BAZE »NAVSTREETS«

7.1 Uvod

Spletno karto Evrope bomo oblikovali na podlagi podatkov prostorske podatkovne baze *Navstreets*, ki jo zagotavlja *Navteq* – eden največjih dobaviteljev prostorskih podatkov na svetu. Baza *Navstreets* vsebuje podatke, ki so namenjeni predvsem za izdelavo cestnonavigacijskih kart in sistemov. Njihovi podatki so vgrajeni v številne navigacijske naprave, njihovi odjemalci so med drugimi *BMW*, *Ford*, *Siemens*, *Garmin*, *Alpine*, *Blaupunkt*, idr. Prostorsko podatkovno bazo *Navstreets* sestavlja skupno 52 različnih podatkovnih slojev, ki zagotavljajo osnovo za popolno navigacijo v cestnonavigacijskih sistemih. Med podatkovnimi sloji so takšni, ki so namenjeni izključno kartografskemu prikazu (npr. podatkovni sloj železnic), nekateri vsebujejo izključno podatke, ki so pomembni za namene navigacije (npr. tabela prepovedi zavijanj), nekaj pa je takšnih, ki so pomembni tako za kartografski prikaz, kakor tudi za namene navigacije (npr. podatkovni sloj cest). V diplomskem delu se osredotočamo le na tiste dele baze *Navstreets*, ki so pomembni za kartografski prikaz. V nadaljevanju opisujemo 33 takšnih slojev, med katerimi se bomo pri opisu atributnih tabel zopet posvetili le opisu tistih atributov, ki so pomembni za kartografski prikaz. Podatki baze *Navstreets* so sicer dostopni v različnih formatih. Podatki, ki jih navajamo v diplomskem delu so zapisani v *ESRI Shape* formatu, podani so v geografskih koordinatah na elipsoidu *WGS-84*. Seznam podatkovnih slojev baze *Navstreets* opisujemo v tabeli 8:

Tabela 8: Seznam podatkovnih slojev baze »Navstreets«.

Ime podatkovnega sloja	Tip podatkovnega sloja	Opis podatkovnega sloja (ENG)	Opis podatkovnega sloja (SLO)
<i>Streets</i>	Linijski	<i>Streets</i>	Ceste
<i>MajHwys</i>	Linijski	<i>Major Highways</i>	Pomembne ceste 1. reda
<i>SecHwys</i>	Linijski	<i>Secondary Highways</i>	Pomembne ceste 2. reda
<i>RailRds</i>	Linijski	<i>Railroads</i>	Železnice
<i>AdminBndy1</i>	Ploskovni	<i>Aministrative Area Boundaries 1</i>	Administrativne enote 1. reda
<i>AdminBndy2</i>	Ploskovni	<i>Aministrative Area Boundaries 2</i>	Administrativne enote 2. reda
<i>AdminBndy3</i>	Ploskovni	<i>Aministrative Area Boundaries 3</i>	Administrativne enote 3. reda

<i>AdminBndy4</i>	Ploskovni	<i>Administrative Area Boundaries 4</i>	Administrativne enote 4. reda
<i>AdminBndy5</i>	Ploskovni	<i>Administrative Area Boundaries 5</i>	Administrativne enote 5. reda
<i>CartoCountry</i>	Linijski	<i>Cartographic Country</i>	Državne meje
<i>WaterPoly</i>	Ploskovni	<i>Waterway Polygons</i>	Ploskovni sloj vod
<i>WaterSeg</i>	Linijski	<i>Waterway Segments</i>	Linijski sloj vod
<i>Landmark</i>	Ploskovni	<i>Building/Landmark Features (A,B)</i>	Objekti/zgradbe
<i>LandUseA</i>	Ploskovni	<i>Landuse Features A</i>	Raba prostora A
<i>LandUseB</i>	Ploskovni	<i>Landuse Features B</i>	Raba prostora B
<i>Islands</i>	Ploskovni	<i>Islands</i>	Otoki
<i>NamedPlc</i>	Točkovni	<i>Named Places</i>	Imena mest in krajev
<i>Hospital</i>	Točkovni	<i>Hospitals</i>	Bolnišnice
<i>ParkRec</i>	Točkovni	<i>Parks and Recreation</i>	Parki in območja za rekreacijo
<i>TransHubs</i>	Točkovni	<i>Transportation Hubs</i>	Javna prevozna sredstva
<i>Travdest</i>	Točkovni	<i>Travel Destinations</i>	Potovalne destinacije
<i>Shopping</i>	Točkovni	<i>Shopping</i>	Nakupi
<i>Restrnrs</i>	Točkovni	<i>Restaurants</i>	Restavracije
<i>Entertn</i>	Točkovni	<i>Entertainment</i>	Zabava
<i>AutoSvc</i>	Točkovni	<i>Auto Maintenance, Servis and Petrol</i>	Avto-servis, vzdrževanje, bencinske črpalke
<i>FinInsts</i>	Točkovni	<i>Financial Institutions</i>	Finančne institucije
<i>Business</i>	Točkovni	<i>Business Facilities</i>	Poslovne zgradbe
<i>CommSvc</i>	Točkovni	<i>Community Service Centres</i>	Družbeni centri
<i>EduInsts</i>	Točkovni	<i>Educational Institutions</i>	Izobraževalne ustanove
<i>Parking</i>	Točkovni	<i>Parking</i>	Parkirišča
<i>BordCross</i>	Točkovni	<i>Border Crossing</i>	Mejni prehodi
<i>PostOffice</i>	Točkovni	<i>Post Office</i>	Poštna poslovalnice
<i>MiscCategories</i>	Točkovni	<i>Miscellaneous Categories</i>	Razno

7.2 Opredelitev območja, ki ga zajema baza »Navstreets« – Evropa

Kljub temu, da smo zgoraj zapisali, da izdelujemo karto Evrope, pa je to le grob opis območja, ki ga bo karta zajemala. Baza *Navstreets* za področje Evrope vsebuje prostorske podatke za večino evropskih držav, a pojem Evrope je zelo raztegljiv in zato velja na tem mestu malo natančneje opisati območje, ki ga pokriva baza.

Baza *Navstreets* ne zajema območja Islandije ter nekaterih medkontinentalnih držav, ki se raztezajo čez območje Evrope in Azije – Turčija, Armenija, Azerbajdžan, Gruzija. Prav tako v bazi ni zajeto območje Cipra, ki geografsko sicer pripada Aziji, a ga zgodovinsko in

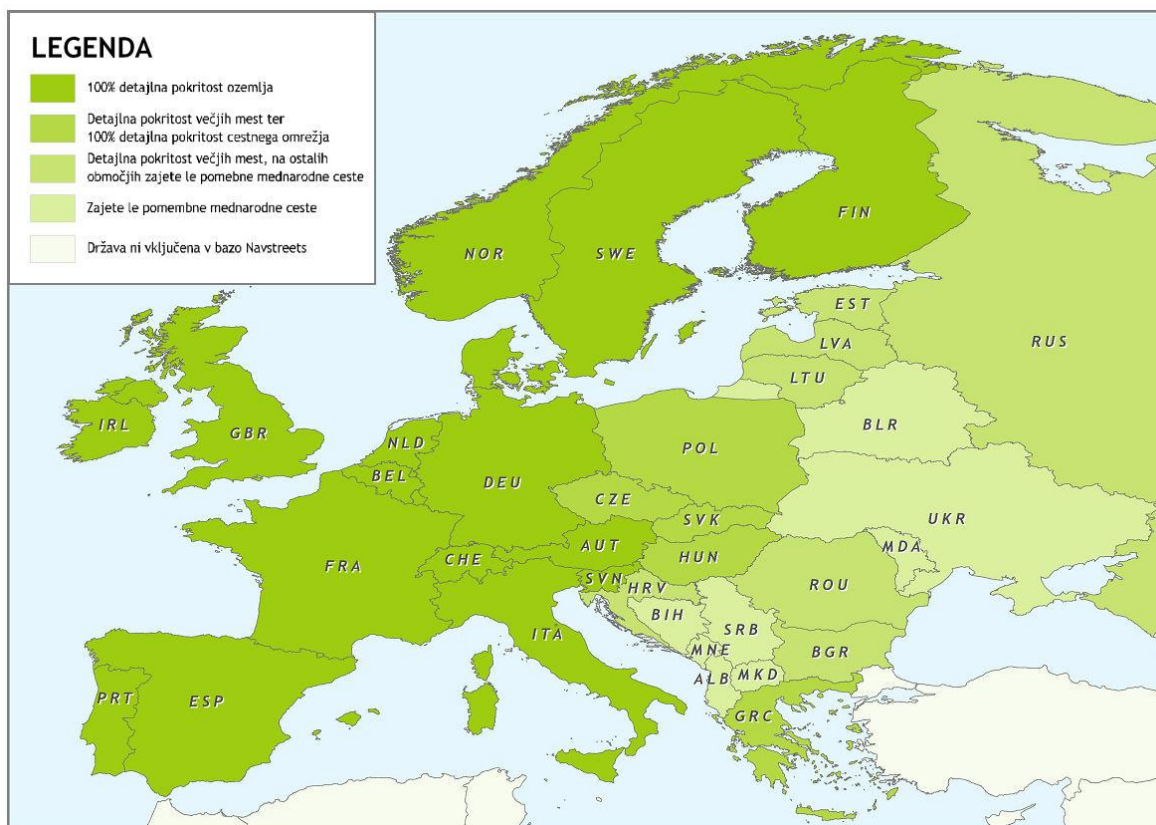
kulturno pojmuje kot del Evrope. Na drugi strani so v bazo vključena nekatera območja, ki v geografskem smislu ne spadajo v Evropo, vendar pa so avtonomni deli evropskih držav. Takšna primera sta portugalsko otočje Madeira ter španski Kanarski otoki (oboje se sicer nahajajo na afriški plošči). Območje Rusije pa je v bazi deloma pokrito tudi v azijskem delu, s tem, da je bil zajem podatkov v bazo na tem področju omejen predvsem na nekatere najpomembnejše prometne povezave.

7.3 Homogenost baze in ažuriranje podatkov v bazi

Iz zgornjega stavka lahko razberemo, da je baza *Navstreets* po svoji sestavi nehomogena. Kakovost in količina zajetih podatkov v bazi se zelo razlikuje od države do države (ter naprej tudi znotraj posamezne države). Odvisna je predvsem od razpoložljivosti kartografskih podlag v posamezni državi in na drugi strani od komercialne upravičenosti zajema podatkov v državi. *Navteq* praviloma odkupuje razpoložljive prostorske podatke, jih nato obdela, pretvori v ustrezen format in jih na koncu vključi v bazo *Navstreets*. Ponavadi se izvede tudi detajlni zajem podatkov, pri katerem se (s posebej opremljenim avtomobilom – GPS sledilna naprava, kamera, terenski računalnik, idr.) prevozijo odseki cest. S tem se zajamejo še podatki o omejitvah na cestah, preverijo se pozicije interesnih točk, idr. V bazi *Navstreets* so detajlno pokrite predvsem države zahodne, jugozahodne ter severne Evrope, medtem ko so države na vzhodu pokrite slabše. Glede na to, kako popoln je bil zajem podatkov v posamezni državi, jih v grobem lahko razdelimo v štiri skupine (slika 18a). V prvi skupini so države, ki so detajlno pokrite na celotnem ozemlju (popolnjeni so vsi podatkovni sloji, maksimalna količina atributov). V drugi skupini držav so tiste, ki imajo detajlno zajeto celotno cestno omrežje, medtem ko je popolnost ostalih podatkov omejena le na področja nekaterih večjih mest v državi. Tretjo skupino držav sestavljajo tiste, ki so detajlno pokrite le na področjih večjih mest, drugje pa je bil zajem podatkov omejen le na pomembnejše prometne povezave. V zadnji, četrti skupini so države pri katerih so bile v bazo zajete v glavnem le najpomembnejše mednarodne prometne povezave.

Podatki v bazi *Navstreets* se redno ažurirajo. *Navteq* zagotavlja nadgradnjo podatkov štirikrat letno. V državah, ki so detajlno pokrite na celotnem območju, se v bazi ažurirajo

npr. novi oz. spremenjeni odseki cest, prometna signalizacija, pozicije interesnih točk. V ostalih državah, ki še niso detajlno pokrite pa se izboljšuje tudi sama pokritost.



Slika 18a: Homogenost podatkov v bazi »Navstreets«.

7.4 Specifikacija podatkovnih slojev baze »Navstreets«

V nadaljevanju bomo opisali podatkovne sloje baze *Navstreets*, ki so pomembni za kartografski prikaz. Zaradi velike količine atributov posameznih slojev, se bomo tudi pri opisu pomena posameznih atributov omejili le na tiste, ki so pomembni za kartografski prikaz.

Podatkovni sloj »Streets«



Slika 19: Podatkovni sloj »Streets« (grafika).

Podatkovni sloj *Streets* je najpomembnejši sloj v bazi *Navstreets*, saj vsebuje podatke o cestnem omrežju. Vsak cestni odsek je v bazi predstavljen z usmerjeno linijo – vektorjem, ki so mu dodani ustrezni atributi (kategorija ceste, omejitev hitrosti, omejitev prevoznosti, idr.). Podatkovni sloj vsebuje podatke, ki služijo namenom navigacije kot tudi za kartografski prikaz. Pomen posameznih atributov opisuje tabela 9.

Tabela 9: Opis atributne tabele podatkovnega sloja »Streets«.

Ime polja/atributa	Tip in dolžina polja	Opis atributa
<i>Link_ID</i>	<i>number(10,0)</i>	Enolični identifikator ceste
<i>St_Name</i>	<i>char(80)</i>	Ime ceste/ulice
<i>St_LangCd</i>	<i>char(3)</i>	Koda, ki opisuje v katerem jeziku je zapisano ime ceste – možne vrednosti so: SLV: slovenska jezikovna koda, ...
<i>Addr_Type</i>	<i>char(1)</i>	Tip imena ceste – možne vrednosti so: B: osnovno ime C: ime mesta D: ime okrožja O: staro ime T: komercialno ime

<i>Func_Class</i>	<i>char(1)</i>	Kategorija ceste – možne vrednosti so: 1: ceste z malo ali brez omejitev hitrosti, omogočajo velik pretok prometa in maksimalno potovalno hitrost med in skozi večja metropolitanska območja 2: ceste z malo ali brez omejitev hitrosti, omogoča velik pretok prometa in veliko potovalno hitrost, priključujejo se na ceste 1. kategorije 3: ceste, ki povezujejo ceste 2. kategorije, omogočajo velik pretok prometa vendar pri manjši stopnji mobilnosti kot ceste 2. kategorije 4: ceste, ki omogočajo velik pretok prometa pri zmernih hitrostih med naselji in sosekami 5: vse ostale ceste NA: nerazvrščene ceste
<i>AR_Auto</i>	<i>char(1)</i>	Omejitev za avtomobile: Y: cesta nima prepovedi za avtomobilski promet N: prepoved za avtomobilski promet
<i>AR_Bus</i>	<i>char(1)</i>	Omejitev za avtobuse: Y: cesta nima prepovedi za avtobusni promet N: prepoved za avtobusni promet
<i>AR_Taxis</i>	<i>char(1)</i>	Omejitev za taxi: Y: cesta nima prepovedi za taxi-je N: prepoved za taxi-je
<i>AR_Carpool</i>	<i>char(1)</i>	Omejitev za <i>carpool</i> : Y: cesta nima <i>carpool</i> * omejitve N: cesta ima <i>carpool</i> omejitev *" <i>carpool lane</i> " v zahodnih državah (predvsem ZDA) označuje prometni pas, ki je namenjen le tistim vozilom, v katerih se vozi več kot ena oseba
<i>AR_Pedest</i>	<i>char(1)</i>	Omejitev za pešce: Y: cesta nima prepovedi za pešce N: cesta ima prepoved za pešce
<i>AR_Truck</i>	<i>char(1)</i>	Omejitev za tovorni promet: Y: cesta nima prepovedi za tovorni promet N: cesta ima prepoved za tovorni promet
<i>AR_Traff</i>	<i>char(1)</i>	Omejitev za tranzitni promet: Y: cesta nima prepovedi za tranzitni promet N: cesta ima prepoved za tranzitni promet
<i>AR_Deliv</i>	<i>char(1)</i>	Omejitev za dostavo: Y: cesta nima prepovedi za dostavna vozila N: cesta ima prepoved za dostavna vozila
<i>AR_EmerVeh</i>	<i>char(1)</i>	Omejitev za reševalna vozila: Y: cesta nima prepovedi za reševalna vozila N: cesta ima prepoved za reševalna vozila

<i>Paved</i>	<i>char(1)</i>	Ali je cesta asfaltirana: Y: cesta je asfaltirana N: cesta ni asfaltirana
<i>Bridge</i>	<i>char(1)</i>	Ali je cesta je del mostu: Y: cesta je del mostu N: cesta ni del mostu
<i>Tunnel</i>	<i>char(1)</i>	Ali je cesta del predora: Y: cesta je del predora N: cesta ni del predora
<i>Ferry_Type</i>	<i>char(1)</i>	Način potovanja: H: cestna pot B: trajektna pot R: železniška pot
<i>Route_type</i>	<i>char(1)</i>	Atribut <i>Route_type</i> nakazuje, da ime avtoceste vsebuje številko avtoceste ter nato identificira tip avtocestne številke (E označba, A označba,...). Tipi označb se razlikujejo od države do države, možne vrednosti za Slovenijo so: 1: evropska cesta (E označba) 2: avtocesta (A označba) 3: hitra cesta (H označba)

Podatkovna sloja »MajHwys« in »SecHwys«



Slika 20: Podatkovna sloja »MajHwys« in »SecHwys« (grafika).

Podatkovna sloja *MajHwys* in *SecHwys* vsebujeta pomembnejše povezovalne ceste (avtoceste, hitre ceste, regionalne ceste, pomembne mestne ceste). Vse te ceste so sicer že zajete v podatkovnem sloju *Streets*, vendar pa sta sloja zaradi praktičnih razlogov dostopna tudi v takšni obliki. Na ta način lahko v manjših merilih enostavneje prikazujemo cestno omrežje (brez dodatnega obremenjevanja baze z izdelavo različnih poizvedb). Sloja služita predvsem za kartografski prikaz, zato je tudi atributna tabela temu ustrezno enostavna.

Tabela 10: Opis atributne tabele podatkovnih slojev »MajHwys« in »SecHwys«.

Ime polja/atributa	Tip in dolžina polja	Opis atributa
<i>Link_ID</i>	<i>Number(10,0)</i>	Enolični identifikator ceste
<i>Highway_Nm</i>	<i>char(80)</i>	Ime ceste
<i>Lang_Code</i>	<i>char(3)</i>	Jezikovna koda za ime ceste
<i>Func_class</i>	<i>char(1)</i>	Kategorija ceste
<i>Route_Type</i>	<i>char(1)</i>	... glej zgoraj (podatkovni sloj <i>Streets</i>)
<i>Ferry_Type</i>	<i>char(1)</i>	Način potovanja: H: cestna pot B: trajektna pot R: železniška pot

Podatkovni sloj »Railrds«



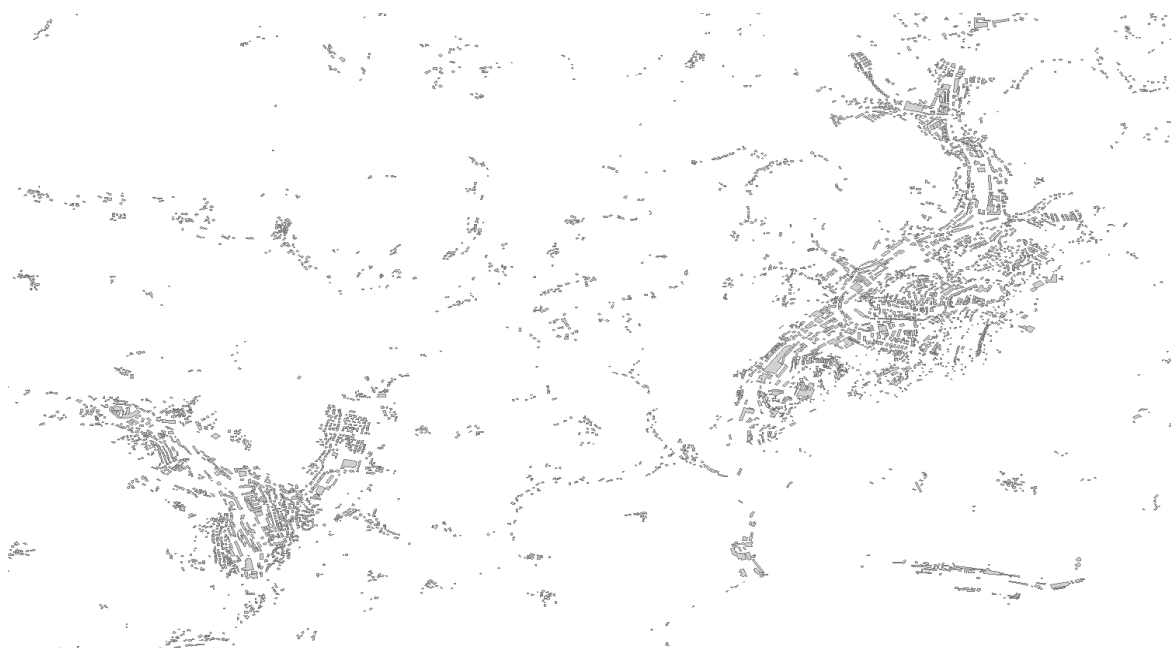
Slika 21: Podatkovni sloj »Railrds« (grafika).

Podatkovni sloj *Railrds* vsebuje podatke o železniškem omrežju. Vsak železniški odsek je zajet z linijo, ki predstavlja os proge. V splošnem so železnice lahko eno, dvo ali več tirne, vendar so zajete le z eno (sredinsko) osjo. Dodani so atributi o imenu proge ter o mostovih in predorih na trasi. Podatki o železniških postajah so v posebnem točkovnem podatkovnem sloju, ki ga opisujemo v nadaljevanju, trase avtovlakov pa so zajete v podatkovnem sloju *Streets*. Sloj *Railrds* služi predvsem kartografski predstavitvi za lažjo orientacijo v prostoru.

Tabela 11: Opis atributne tabele podatkovnega sloja »Railrds«.

Ime polja/atributa	Tip in dolžina polja	Opis atributa
<i>Link_ID</i>	<i>number(10,0)</i>	Enolični identifikator železnice
<i>Railway_Nm</i>	<i>char(35)</i>	Ime železnice
<i>Lang_Code</i>	<i>char(3)</i>	Jezikovna koda za ime železnice
<i>Bridge</i>	<i>char(1)</i>	Ali je železnica del mostu: Y: železnica je del mostu N: železnica ni del mostu
<i>Tunnel</i>	<i>char(1)</i>	Ali je železnica del predora: Y: železnica je del predora N: železnica ni del predora

Podatkovni sloj »Landmark«



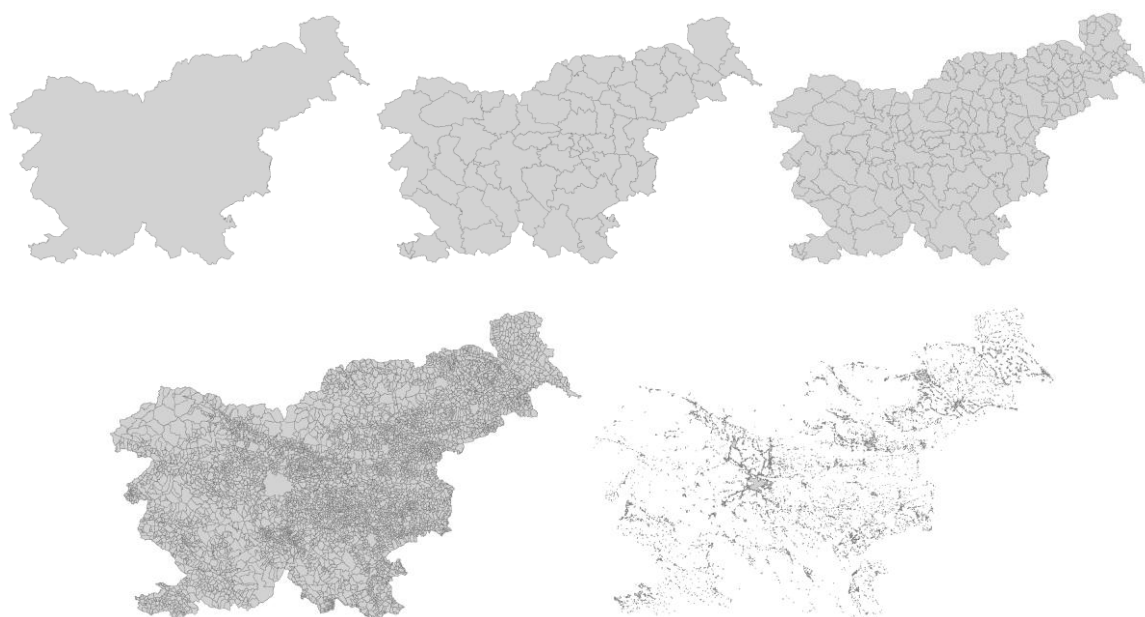
Slika 22: Podatkovni sloj »Landmark« (grafika).

V podatkovnem sloji *Landmark* so zajeti objekti oz. stavbe. Vsaka stavba je zajeta s svojo ploskvijo, stavbe so klasificirane glede na namembnost (stanovanjske stavbe, poslovne stavbe, javni objekti, idr.). Sloj služi predvsem kartografski predstavitvi.

Tabela 12: Opis atributne tabele podatkovnega sloja »Landmark«.

Ime polja/atributa	Tip in dolžina polja	Opis atributa
<i>Polygon_ID</i>	<i>number(10,0)</i>	Enolični identifikator zgradbe
<i>Polygon_Nm</i>	<i>char(35)</i>	Ime zgradbe
<i>Nm_LangCd</i>	<i>char(3)</i>	Jezikovna koda za ime zgradbe
<i>Feat_Type</i>	<i>char(30)</i>	Oznaka za tip zgradbe – možne vrednosti so: <i>BUSINESS/COMMERCE</i> : poslovni/industrijski objekt <i>CULTURAL</i> : kulturni objekt <i>EDUCATION</i> : izobraževalni objekt <i>GOVERNMENT</i> : vladni objekt <i>HISTORICAL</i> : zgodovinski objekt <i>MEDICAL</i> : zdravstveni objekt <i>PARK/LEISURE</i> : objekt namenjen zabavi <i>RESIDENTIAL</i> : stanovanjska stavba <i>SPORTS</i> : športni objekt <i>TOURIST</i> : turistični objekt <i>TRANSPORTATION</i> : transportni objekt

Podatkovni sloji »AdminBndy1« - »AdminBndy5«



Slika 23: Podatkovni sloji »AdminBndy1« – »AdminBndy5« (grafika).

Podatkovni sloji *AdminBndy1* – *AdminBndy5* vsebujejo podatke o administrativnih enotah. Ti podatki so uradni podatki (za območje Slovenije jih npr. zagotavlja Geodetska uprava RS). Izjema je podatkovni sloj *AdminBndy5* – ta ni iz uradnih evidenc, predstavlja pa zgrajena območja ali t.i. *built-up area*. *AdminBndy5* je ponavadi zgrajen na podlagi generaliziranega sloja objektov. Vsaka administrativna enota je predstavljena s ploskvijo, dodano ji je ime. Administrativne enote v Sloveniji so klasificirane na sledeč način:

- *AdminBndy1* (država),
- *AdminBndy2* (upravne enote),
- *AdminBndy3* (občine),
- *AdminBndy4* (naselja),
- *AdminBndy5* (zgrajena območja).

Atributna tabela za posamezne administrativne enote je podobna, opisana je v tabeli 13.

Tabela 13: Opis atributne tabele podatkovnih slojev »AdminBndy«.

Ime polja/atributa	Tip in dolžina polja	Opis atributa
<i>Area_ID</i>	<i>number(10,0)</i>	Enolični identifikator administrativne enote
<i>Polygon_Nm</i>	<i>char(35)</i>	Ime administrativne enote
<i>Nm_Lang_Code</i>	<i>char(3)</i>	Jezikovna koda za ime administrativne enote

Podatkovni sloj »CartoCountry«



Slika 24: Podatkovni sloj »CartoCountry« (grafika).

V podatkovnem sloju *CartoCountry* so zajete državne meje. *CartoCountry* je prav tako pridobljen iz uradnih evidenc. Državne meje so zajete z linijo, sloj je namenjen predvsem kartografskemu prikazu.

Tabela 14: Opis atributne tabele podatkovnega sloja »*CartoCountry*«.

Ime polja/atributa	Tip in dolžina polja	Opis atributa
<i>Link_ID</i>	<i>number(10,0)</i>	Enolični identifikator meje

Podatkovna sloja »*LandUseA*« in »*LandUseB*«



Slika 25: Podatkovna sloja »*LandUseA*« in »*LandUseB*« (grafika).

Podatkovna sloja *LandUseA* in *LandUseB* zajemata podatke o rabi prostora. Oba sloja sta ploskovna. V sloju *LandUseA* so zajeti so podatki o letališčih, pokopališčih, industrijskih conah, gozdovih, parkih, nakupovalnih centrih, športnih kompleksih, idr. Sloj *LandUseB* pa zajema podatke o rabi prostora manjših območij, ki ležijo znotraj večjih območij rabe. Gre za letališke steze, peš cone, igrišča za golf.

Tabela 15: Opis atributne tabele podatkovnih slojev »LandUseA« in »LandUseB«.

Ime polja/atributa	Tip in dolžina polja	Opis atributa
<i>Polygon_ID</i>	<i>number(10,0)</i>	Enolični identifikator ploskve
<i>Polygon_Nm</i>	<i>char(35)</i>	Ime pojava
<i>Nm_LangCd</i>	<i>char(3)</i>	Jezikovna koda za ime
<i>Feat_Type</i>	<i>char(30)</i>	<p>Oznaka za tip rabe – možne vrednosti so:</p> <p><u>za podatkovni sloj <i>LandUseA</i>:</u></p> <ul style="list-style-type: none"> - <i>AIRPORT</i>: letališče - <i>CEMETERY</i>: pokopališče - <i>HOSPITAL</i>: bolnišnica - <i>INDUSTRIAL COMPLEX</i>: industrijski kompleks - <i>PARK (CITY/COUNTY)</i>: mestni/regionalni park - <i>PARK (STATE)</i>: nacionalni park - <i>PARK/MONUMENT (NATIONAL)</i>: spominski park državnega pomena - <i>PARK IN WATER</i>: naravni vodni park - <i>PARKING LOT</i>: parkirne površine - <i>SHOPPING CENTRE</i>: trgovski center - <i>SPORTS COMPLEX</i>: športni kompleks - <i>UNDEFINED TRAFFIC AREA</i>: nedefinirane prometne površine - <i>UNIVERSITY/COLLEGE</i>: univerzitetno območje - <i>WOODLAND</i>: gozd <p><u>za podatkovni sloj <i>LandUseB</i>:</u></p> <ul style="list-style-type: none"> - <i>AIRCRAFT ROADS</i>: letališka steza - <i>GOLF COURSE</i>: igrišče za golf - <i>PEDESTRIAN ZONE</i>: cona za pešce

Podatkovni sloj »Oceans«



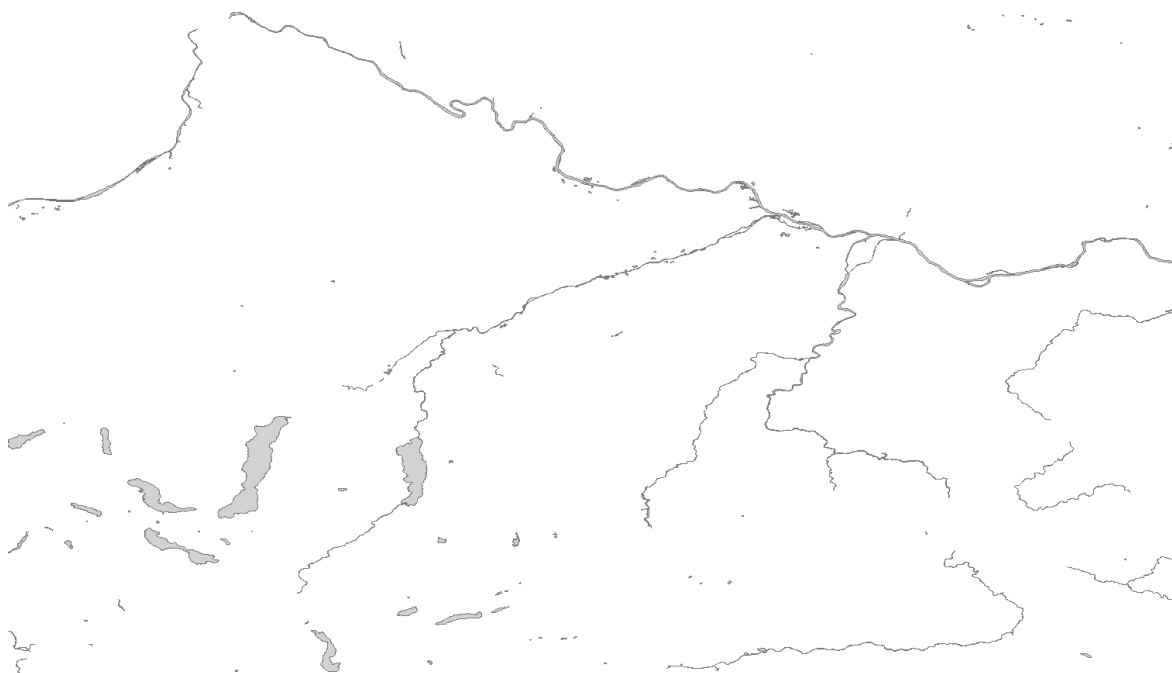
Slika 26: Podatkovni sloj »Oceans« (grafika).

Podatkovni sloj *Oceans* vsebuje podatke o morjih in oceanih. V bazi so ti predstavljeni s ploskvami, dodan jim je atribut, ki opisuje ime morja/oceana.

Tabela 16: Opis atributne tabele podatkovnega sloja »Oceans«.

Ime polja/atributa	Tip in dolžina polja	Opis atributa
<i>Polygon_ID</i>	<i>number(10,0)</i>	Enolični identifikator morja
<i>Polygon_Nm</i>	<i>char(35)</i>	Ime morja
<i>Nm_Lang_Code</i>	<i>char(3)</i>	Jezikovna koda za ime morja

Podatkovni sloj »WaterPoly«



Slika 27: Podatkovni sloj »WaterPoly« (grafika).

V sloju *WaterPoly* so s ploskvami zajete vse večje reke, jezera in nekatere druge vrste vodnih površin. Sloj je namenjen predvsem kartografskemu prikazu.

Tabela 17: Opis atributne tabele podatkovnega sloja »WaterPoly«.

Ime polja/atributa	Tip in dolžina polja	Opis atributa
<i>Polygon_ID</i>	<i>number(10,0)</i>	Enolični identifikator vode
<i>Polygon_Nm</i>	<i>char(35)</i>	Ime vode
<i>Nm_LangCd</i>	<i>char(3)</i>	Jezikovna koda za ime vode
<i>Feat_Type</i>	<i>char(30)</i>	Tip vode: <i>RIVER</i> : reka <i>LAKE</i> : jezero <i>BAY/HARBOUR</i> : obala/pristanišče <i>CANAL/WATER CHANNEL</i> : vodni kanal

Podatkovni sloj »WaterSeg«



Slika 28: Podatkovni sloj »WaterSeg« (grafika).

Podatkovni sloj *WaterSeg* vsebuje vse ostale pomembne tekoče vode, ki niso zajete v sloju *WaterPoly*. Vodotoki so predstavljeni z linijo, dodani so jim atributi o imenu vode in tipu vodotoka.

Tabela 18: Opis atributne tabele podatkovnega sloja »WaterSeg«.

Ime polja/atributa	Tip in dolžina polja	Opis atributa
<i>Link_ID</i>	<i>number(10,0)</i>	Enolični identifikator vode
<i>Polygon_Nm</i>	<i>char(35)</i>	Ime vode
<i>Nm_LangCd</i>	<i>char(3)</i>	Jezikovna koda za ime vode
<i>Feat_Type</i>	<i>char(30)</i>	Tip vode, možne vrednosti so: RIVER: reka CANAL/WATER CHANNEL: potok/vodni kanal

Podatkovni sloj »Islands«



Slika 29: Podatkovni sloj »Islands« (grafika).

Podatkovni sloj *Islands* vsebuje podatke o otokih – ti so predstavljeni s ploskvami, dodan jim je atribut z imenom otoka. Tudi ta sloj služi predvsem kartografski predstavitvi.

Tabela 19: Opis atributne tabele podatkovnega sloja »Islands«.

Ime polja/atributa	Tip in dolžina polja	Opis atributa
<i>Polygon_ID</i>	<i>number(10,0)</i>	Enolični identifikator otoka
<i>Polygon_Nm</i>	<i>char(35)</i>	Ime otoka
<i>Nm_LangCd</i>	<i>char(3)</i>	Jezikovna koda za ime otoka

Podatkovni sloj »NamedPlc«



Slika 30: Podatkovni sloj »NamedPlc« (grafika).

V podatkovnem sloju *NamedPlc* so s točkami zajeti kraji in mesta. Točka predstavlja center mesta/kraja. Mestom je poleg imena dodan atribut še o številu prebivalstva, atribut, ki opisuje, ali je mesto glavno mesto katere izmed administrativnih enot. Podatkovni sloj služi kartografski predstavitvi, kot tudi namenom navigacije (iskanje lokacije – glede na ime mesta).

Tabela 20: Opis atributne tabele podatkovnega sloja »NamedPlc«.

Ime polja/atributa	Tip in dolžina polja	Opis atributa
<i>Poi_ID</i>	<i>number(10,0)</i>	Enolični identifikator mesta
<i>Poi_Name</i>	<i>char(35)</i>	Ime mesta
<i>Nm_LangCd</i>	<i>char(3)</i>	Jezikovna koda za ime mesta
<i>Poi_NmTyp</i>	<i>char(1)</i>	Tip imena mesta, možne vrednosti so: B: osnovno ime E: eksonim (tujka) S: sinonim (sopomenka) U: neimenovan
<i>Population</i>	<i>number(10,0)</i>	Število prebivalstva
<i>Capital</i>	<i>char(1)</i>	Atribut, ki opisuje ali je mesto glavno mesto katere izmed administrativnih enot – možne vrednosti so od 1 – 5 (1: glavno mesto države, 2: glavno mesto administrativne enote 2. reda, itd.)

Skupina podatkovnih slojev »POI«



Slika 31: Podatkovni sloji »POI« (grafika).

Skupina *POI* je sestavljena iz več različnih podatkovnih slojev, ki pa imajo podobne atributne tabele in jih zato opisujemo skupaj. V slojih *POI* so zajete različne interesne točke (bencinske črpalke, parkirišča, avtobusne postaje, restavracije, idr.) Vsaka interesna točka je zajeta s točko, dodani so ji atributi o tipu interesne točke, imenu, naslovu, telefonski številki, idr. Sloji interesnih točk se uporabljajo za namene navigacije (iskanje lokacije glede na ime interesne točke), prav tako pa pripomorejo k lažji orientaciji na karti.

V skupino *Points of Interest* spadajo naslednji podatkovni sloji:

- *Hospital (Hospitals)*,
- *ParkRec (Park and Recreation)*,
- *TranHubs (Transportation Hubs)*,
- *TravDest (Travel Destinations)*,
- *Shopping (Shopping)*,
- *Restrnrs (Restaurants)*,
- *Entertn (Entertainment)*,
- *AutoSvc (Auto Maintenance, Service and Petrol)*,
- *FinInsts (Financial Institutions)*,

- *Business (Business Facilities)*,
- *CommSvc (Community Service Centres)*,
- *EduInsts (Educational Institutions)*,
- *Parking (Parking)*,
- *BordCross (Border Crossing)*,
- *PostOffice (Post Office)*,
- *MiscCategories (Miscellaneous Categories)*

Tabela 21: Opis atributne tabele podatkovnih slojev »POI«.

Ime polja/atributa	Tip in dolžina polja	Opis atributa
<i>Poi_ID</i>	<i>number(10,0)</i>	Enolični identifikator interesne točke
<i>Fac_Type</i>	<i>number(5,0)</i>	<p>Koda za tip interesne točke – možne vrednosti so:</p> <p><u>podatkovni sloj <i>Hospital</i>:</u> 8060: bolnišnica 9583: zdravstvene storitve</p> <p><u>podatkovni sloj <i>ParkRec</i>:</u> 7996: zabavišni park 7992: igrišče za golf 8410: muzej 7940: športni kompleks 4493: marina 4580: športno letališče 7933: igrišče za bowling 7947: park, površine za rekreacijo 7985: casino 7997: športni center 7998: drsališče</p> <p><u>podatkovni sloj <i>TranHubs</i>:</u> 4581: mednarodno letališče 4170: avtobusna postaja 4013: železniška postaja 4482: trajektni terminal 4100: železniška postaja za regionalni vlak</p> <p><u>podatkovni sloj <i>TravDest</i>:</u> 7011: hotel 7999: turistična atrakcija 7389: turistične informacije 7990: sejem, razstavni prostor</p>

		<p>2084: vinoteka 7012: smučišče 7897: počivališče 5999: zgodovinski spomenik 7510: rent-a-car 7013: gostišče</p> <p><u>podatkovni sloj <i>Shopping</i>:</u> 6512: trgovina – splošno 5400: prehrambena trgovina 9995: knjigarna 9565: lekarna</p> <p><u>podatkovni sloj <i>Restrnst</i>:</u> 5800: restavracija 9996: kavarna</p> <p><u>podatkovni sloj <i>Entertn</i>:</u> 5813: nočni lokal 7929: odrska umetnost 7832: kino 9532: bar, pub</p> <p><u>podatkovni sloj <i>AutoSvc</i>:</u> 5540: bencinska črpalka 7538: avtomobilski servis in vzdrževanje 5511: prodajalec avtomobilov 8699: avto-klub 5571: prodajalec motornih koles 5512: prodajalec rabljenih avtomobilov</p> <p><u>podatkovni sloj <i>FinInsts</i>:</u> 6000: banka 3578: bankomat</p> <p><u>podatkovni sloj <i>Business</i></u> 5000: poslovna stavba 9991: industrijski objekt, industrijska cona</p> <p><u>podatkovni sloj <i>CommSvc</i>:</u> 7994: družbeni center 9121: mestna/občinska hiša 9211: sodišče 9221: policijska postaja 9992: religiozni prostor/objekt 9993: veleposlaništvo 9994: okrožni zbor</p>
--	--	---

		<p><u>podatkovni sloj <i>EduInsts</i>:</u> 8200: izobraževalna ustanova (višja izobrazba) 8211: šola 8231: knjižnica</p> <p><u>podatkovni sloj <i>Parking</i>:</u> 7520: parkirišče 7521: parkirna hiša 7522: parkiraj in odpelji (<i>Park & Ride</i>)</p> <p><u>podatkovni sloj <i>BordCross</i>:</u> 9999: mejni prehod</p> <p><u>podatkovni sloj <i>PostOffice</i>:</u> 9530: pošta</p> <p><u>podatkovni sloj <i>MiscCategories</i>:</u> 9591: pokopališče *1001: toplice *1002: kamp *1003: postajališče za avtodom *1004: gasilski dom *1005: vladne ustanove</p> <p>*teh kategorij v originalni navstreets specifikaciji ni – specifikacijo smo razširili zato, da smo lahko dodali nekatere interesne točke, ki so specifične za področje Slovenije.</p>
<i>Poi_Name</i>	<i>char(35)</i>	Ime interesne točke
<i>Nm_LangCd</i>	<i>char(3)</i>	Jezikovna koda za ime interesne točke
<i>Poi_NmTyp</i>	<i>char(1)</i>	Tip imena interesne točke B: osnovno ime E: eksonim (tujka) S: sinonim (sopomenka) U: neimenovano
<i>Poi_st_num</i>	<i>char(10)</i>	Hišna številka interesne točke
<i>St_name</i>	<i>char(80)</i>	Naslov (ime ulice) interesne točke
<i>St_Lang_Code</i>	<i>char(3)</i>	Jezikovna koda za ime ulice
<i>Ph_Number</i>	<i>char(15)</i>	Telefonska številka interesne točke
<i>Bld_Type</i>	<i>char(30)</i>	To polje se uporablja le v tabeli podatkovnega sloja <i>CommSvc</i> – nanaša se na <i>Fac_Type</i> = 9992 ter označuje tip religioznega objekta – možne vrednosti so: <i>MOSQUE</i> : mošeja <i>CHURCH</i> : cerkev <i>TEMPLE</i> : tempelj <i>SYNAGOGUE</i> : sinagoga

8 VZPOSTAVITEV SPLETNE KARTE

8.1 Opis odprtokodnih aplikacij, ki so potrebne za vzpostavitev spletne karte

Za vzpostavitev spletne karte potrebujemo naslednjo programsko opremo:

- zanesljiv operacijski sistem na katerem bomo poganjali aplikacije,
- učinkovit sistem za upravljanje z bazo podatkov – uporabljali ga bomo za hrambo naših prostorskih podatkov, sistem mora omogočati učinkovito in hitro poizvedovanje po bazi podatkov,
- kartografski strežnik – ta bo na osnovi prostorskih podatkov iz podatkovne baze izdeloval karte ter jih v obliki slikovnih datotek pošiljal spletnemu strežniku,
- pregledovalnik kart – interaktiven in uporabniško prijazen pregledovalnik,
- zanesljiv in zmogljiv spletni strežnik za strežbo spletnih strani in dostavo kart do spletnega uporabnika.

»Linux«

Linux je med odprtokodno opremo najbrž najbolj razpoznavno ime – gre za priljubljen odprtokodni operacijski sistem, ki ga v prvi vrsti zaznamuje predvsem visoka:

- stabilnost in zanesljivost (sistem deluje konsistentno, brez prekinitev, zelo redko ga je potrebno ponovno zaganjati, praktično le ob morebitnih posodobitvah),
- zmogljivost,
- varnost (visoka stopnja varnosti - za *Linux* praktično ni virusov).

Linux je zaradi svojih dobrih lastnosti med drugim idealno okolje za poganjanje različnih strežnikov kot so npr. spletni strežniki. Obstaja veliko (več sto) različnih distribucij *Linuxa*, ki zadostijo različnim potrebam uporabnikov. Mi smo uporabili distribucijo *Debian*.

»Apache Tomcat«

Apache Tomcat je odprtokodni spletni in aplikacijski strežnik. Njegova modularna zgradba, podpora sodobnim standardom, zanesljivo delovanje, dobra odzivnost ter majhna poraba sistemskih zmogljivosti so razlogi, da je *Apache* že vse od leta 1996 najpopularnejši spletni strežnik. V novembru 2007 je *Apache Tomcat* stregel 50,76% od vseh spletnih

strani (vir: Internet 2). *Apache* se uporablja za strežbo statičnih in dinamičnih spletnih strani in deluje na različnih operacijskih sistemih, med drugim tudi na *Linuxu*.

»PostgreSQL« in »PostGIS«

PostgreSQL (oz. na kratko *Postgres*) je zelo zmogljiv odprtokodni sistem za upravljanje z bazo podatkov oz. t.i. ORDBMS (*Object-relational database management system*). Pogosto se označuje tudi kot najbolj napreden takšen odprtokodni sistem. *Postgres* ponuja podobno funkcionalnost in vsebino, kot jo najdemo pri drugih komercialnih sistemih, za razliko od njih pa ponuja tudi veliko možnosti za razširitev sistema. Deluje na različnih platformah, med drugim tudi na *Windowsih*, *UNIX-u* in *Linuxu*.

PostGIS je prostorska razširitev sistema *Postgres* in sistemu omogoča hranjenje prostorskih podatkov ter upravljanje z njimi. *PostGIS* izpolnjuje OGC specifikacijo izvedbe enostavnih pojavov (*OpenGIS Simple Features Implementation Specification*). Ponuja mnogo različnih funkcij, npr. sposobnost izvajanja geometrijskih testov (preseki, razlike, buferji, idr.), enostavne analitične funkcije (površina, obseg, dolžina), omogoča prostorsko indeksiranje podatkov, ponuja orodja za uvoz ter izvoz prostorskih podatkov, idr.

»Geoserver«

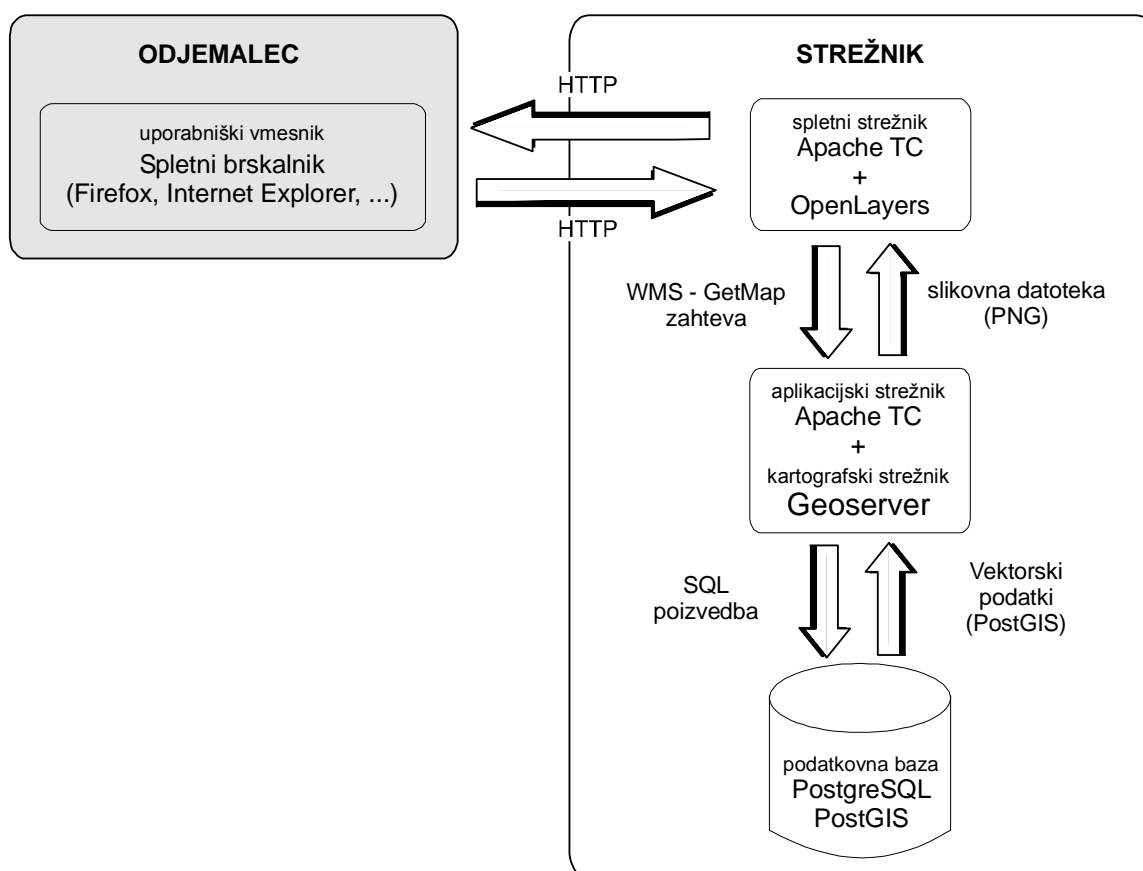
Geoserver je odprtokodni kartografski strežnik, ki omogoča strežbo prostorskih podatkov preko standardnih internetnih protokolov ter v popolnosti izpolnjuje OGC standarde. Zgrajen je na osnovi *Geotoolsov*. Podpira WMS, WFS ter WCS servise. Podpira vhodne podatkovne formate: *Oracle Spatial*, *ArcSDE*, *PostGIS*, *MySQL*, *Shape* in izhodne formate PNG, SVG, GML, PDF, SHP, idr. *Geoserver* je preprost za uporabo, poleg tega ima dobro dokumentirano uporabniško dokumentacijo in zagotovljeno podporo v obliki uporabniških forumov. Zaradi mnogih dobrih lastnosti je *Geoserver* v zadnjih letih eden najpopularnejših odprtokodnih kartografskih strežnikov.

»OpenLayers«

OpenLayers je odrtokodna *JavaScript* knjižnica, ki nam ponuja ustrezne funkcije za postavitev karte v spletnem brskalniku. Z uporabniškega vidika je to enostavno spletni pregledovalnik za karte. Z njim lahko na enostaven način na spletno stran postavimo dinamično spletno karto. *OpenLayers* med drugim podpira WMS in WFS OGC standarda. Omogoča poljubno prilagoditev pregledovalnika kart – na karto lahko dodajamo različne uporabniške kontrole: pomanjšan prikaz karte, *pan*, *zoom* orodje, merilo, meni za vklopljanje in izklopljanje posameznih podatkovnih slojev, idr.

8.2 Programska arhitektura sistema za strežbo kart

Programska arhitektura opredeljuje zgradbo sistema za strežbo kart na nivoju programske opreme. Opredeljuje način komunikacije med posameznimi aplikacijami oz. vse operacije, ki so potrebne od trenutka, ko odjemalec preko internetnega omrežja pošlje zahtevo po karti spletnemu strežniku, do trenutka, ko odjemalec prejme želeno karto in se le ta prikaže v njegovem spletnem brskalniku.



Slika 32: Shema programske arhitekture sistema za strežbo kart.

Uporabnik v svojem spletnem brskalniku poda zahtevo po karti, ki se preko HTTP protokola prenese do spletnega strežnika. Spletni strežnik sprejme uporabniško zahtevo in jo posreduje naprej kartografskemu strežniku v obliki WMS zahteve (*GetMap*). V zahtevi *GetMap* so opredeljeni: merilo karte, območje prikaza, kartografska projekcija, format vrnjene slikovne datoteke, idr. Kartografski strežnik na osnovi te zahteve izdela SQL poizvedbo, ki jo posreduje podatkovni bazi (oz. sistemu za upravljanje z bazo podatkov). Baza vrne želene podatke v vektorski obliki – v *PostGIS* formatu. *Geoserver* na osnovi vrnutih podatkov izvede transformacijo v zahtevano kartografsko projekcijo in nato na osnovi podatkov o oblikovalskih stilih, ki so zapisani v SLD datotekah (in nekaterih drugih parametrov), sestavi sliko karte. Slikovno datoteko vrne spletnemu strežniku, ta pa jo na koncu posreduje uporabniku. Karto lahko uporabnik pregleduje v pregledovalniku kart (*OpenLayers*), ki je vgrajen v spletno stran.

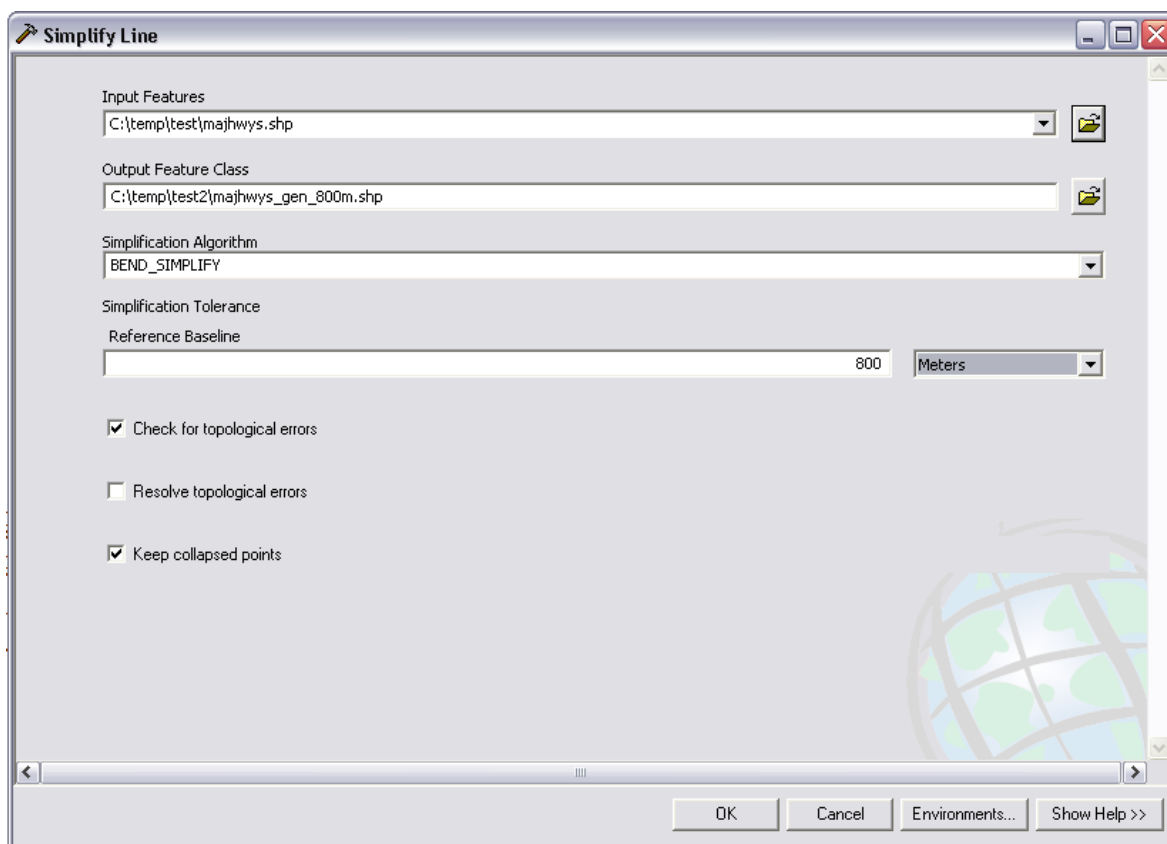
8.3 Priprava podatkov, oblikovanje in vzpostavitev spletne karte Evrope

Priprava podatkov

V prvi fazi vzpostavitve spletne karte je bilo potrebno ustrezno pripraviti prostorske podatke. Za prikaz v majhnih merilih je potrebno podatke predhodno generalizirati – v smislu poenostavljanja geometrije pojavov. Večina podatkov je bila v podatkovno bazo namreč zajeta na osnovi kartografskih podlag velikih meril in zato je kompleksnost prikaza posameznih pojavov v bazi (ne)ustrezno visoka. Kljub temu, da kartografski strežnik lahko v majhnih merilih prikaže tudi geometrijo negeneraliziranih pojavov, pa je preglednost prikaza v takšnem primeru slabša. Poleg tega je obremenitev strežnika zelo velika in posledično strežba kart s tem počasnejša. Zato smo predhodno generalizirali vse sloje, ki jih na karti prikazujemo do merila 1 : 500.000. Uporabili smo program (*Esri*) *ArcMap* ter orodji *SimplifyLine* ter *SimplifyPolygon* (slika 33). Za posamezne razpone meril smo določili različne stopnje generalizacije ter generalizirane podatke zapisali v nove podatkovne sloje. Konkretni parametri generalizacije za posamezne podatkovne sloje so navedeni v tabeli 22.

Tabela 22: Parametri generalizacije za posamezne podatkovne sloje.

Podatkovni sloj	Uporabljen algoritem gen.	Toleranca generalizacije	Razpon merila
<i>MajHwys</i>	<i>Bend Symplify</i>	4 km	1 : 34 milj. – 1 : 5 milj.
	<i>Bend Symplify</i>	800 m	1 : 5 milj. – 1 : 500.000
<i>SecHwys</i>	<i>Bend Symplify</i>	800 m	1 : 1 milj. – 1 : 500.000
<i>RailRds</i>	<i>Bend Symplify</i>	800 m	1 : 1 milj. – 1 : 500.000
<i>CartoCountry</i>	<i>Bend Symplify</i>	4 km	1 : 34 milj. – 1 : 5 milj.
	<i>Bend Symplify</i>	800 m	1 : 5 milj. – 1 : 500.000
<i>AdminBndy1</i>	<i>Bend Symplify</i>	4 km	1 : 34 milj. – 1 : 5 milj.
	<i>Bend Symplify</i>	800 m	1 : 5 milj. – 1 : 500.000
<i>AdminBndy5</i>	<i>Bend Symplify</i>	800m/500.000 m ² minimalne površine	1 : 5 milj. – 1 : 500.000
<i>Oceans</i>	<i>Bend Symplify</i>	4 km	1 : 34 milj. – 1 : 5 milj.
	<i>Bend Symplify</i>	800 m	1 : 5 milj. – 1 : 500.000
<i>Islands</i>	<i>Bend Symplify</i>	4 km/10 km ² minimalne površine	1 : 34 milj. – 1 : 5 milj.
	<i>Bend Symplify</i>	800m/500.000 m ² minimalne površine	1 : 5 milj. – 1 : 500.000

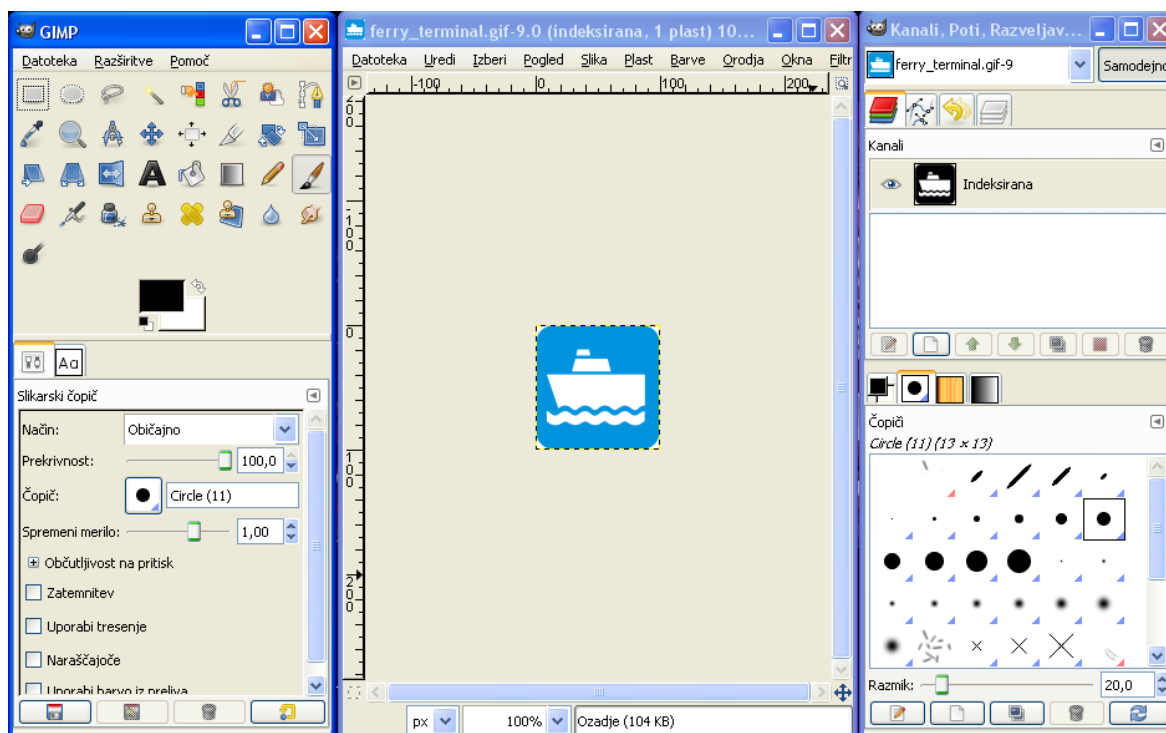


Slika 33: Generalizacija geometrije v programskem okolju »ArcMap«.

Oblikovanje kartografskih znakov

Naslednja faza projekta je bila oblikovanje kartografskih znakov. Pri tem smo se naslanjali na teorijo kartografskega oblikovanja – teorijo kartografskih izraznih sredstev, grafičnih spremenljivk in kartografske generalizacije. Za posamezne podatkovne sloje in za posamezne razpone meril smo opredelili, kateri pojavi se bodo prikazovali in kako se bodo prikazovali. Izdelali smo knjižnico kartografskih znakov (priloga B) ter zapisali SLD datoteke, v katerih smo formalno opredelili oblikovalska pravila (v jeziku, ki je razumljiv kartografskemu strežniku). Primer takšne SLD datoteke – za podatkovni sloj "*Streets*" – je priložen v prilogi Č.

Na karti prikazujemo lokacije interesnih točk, zato je bilo potrebno za posamezne tipe interesnih točk izdelati ustrezne točkovne kartografske znake. Kartografske znake smo oblikovali asociativno, glede na značilnosti prikazane interesne točke – znaki so zbrani in prikazani v knjižnici kartografskih znakov (priloga C). Pri oblikovanju smo si zopet pomagali z odprtokodno programsko opremo. Kartografske znake smo oblikovali v programskem okolju *Gimp*, ki je nekakšna različica bolj znanega *PhotoShop* (torej program za obdelavo fotografij in grafike) in deluje tako na *Linuxu*, kot tudi na *Windowsih*. Znake smo izdelali s predelavo nekaterih *Esrijevih* simbolnih pisav in simbolnih pisav, ki so dostopne brezplačno na spletu.



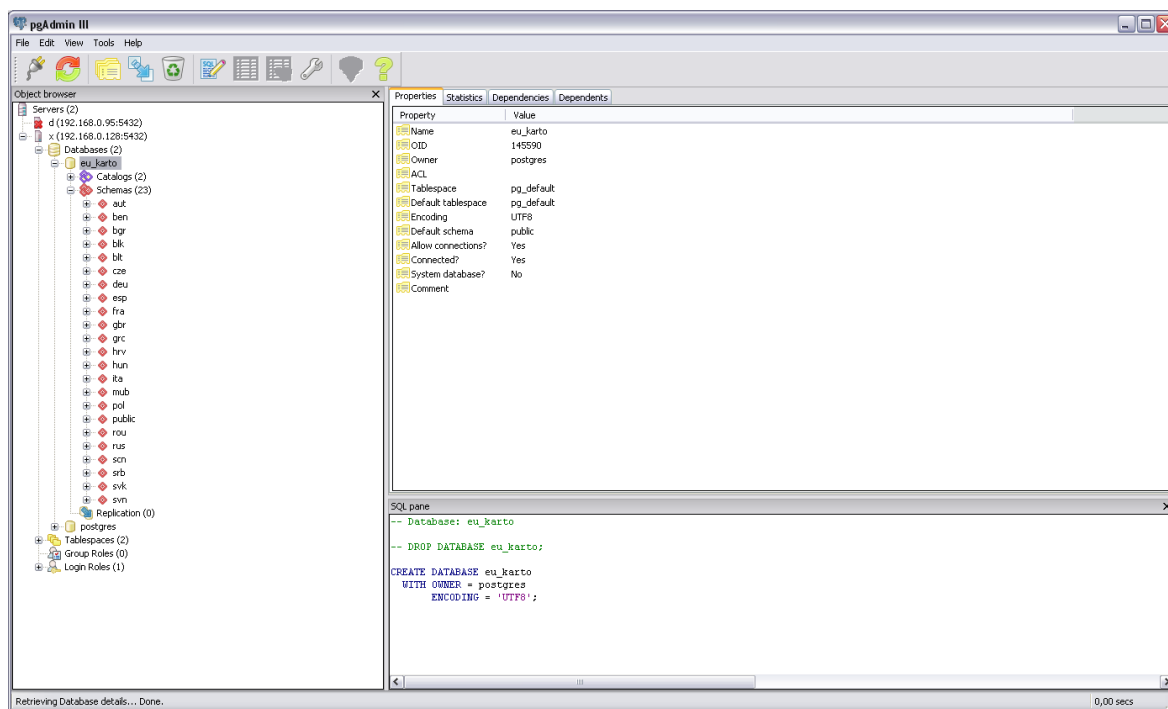
Slika 34: Oblikovanje točkovnih kartografskih znakov v programskem okolju »Gimp«.

8.4 Uvoz »shape« datotek v bazo »PostgreSQL/PostGIS« in indeksiranje podatkov

Prostorske podatke, zapisane v *shape* datotekah, je bilo potrebno uvoziti v sistem za upravljanje z bazo podatkov in jih optimirati za potrebe kasnejših poizvedovanj, z dodelitvijo t.i. indeksov. Uvoz v *Postgres* bazo in indeksiranje smo izvajali preko *Postgres* konzole ter s pomočjo administracijskega programa *PgAdminIII* (slika 35). *PgAdminIII* je odprtokodna administracijska in razvojna platforma za *Postgres*. Uporabniški vmesnik podpira vse *Postgres* funkcije in omogoča enostavno administracijo. Z njim lahko pišemo enostavne SQL poizvedbe oz. na drugi strani razvijamo zelo kompleksne podatkovne baze. Deluje na različnih operacijskih sistemih, med drugim tudi na operacijskem sistemu *Windows*.

Datoteke s prostorskimi podatki smo najprej kopirali na *Postgres* strežnik. Podatki so bili zapisani v *ESRI Shape* formatu – v različnih podatkovnih slojih, za vsako državo posebej. Imena posameznih slojev so se znotraj vsake države ponavljala. Za vsako državo je bilo zato potrebno v podatkovni bazi najprej izdelati lastno shemo, ki v podatkovni bazi deluje kot nekakšen imenski prostor. Vsaka takšna shema vsebuje imenovane objekte (tabele,

podatkovne tipe, funkcije in operacije), katerih imena se lahko podvajajo s tistimi objekti, ki se nahajajo v drugih shemah. Do objektov v shemi kasneje dostopamo tako, da opredelimo ime objekta skupaj s predpono, ki vsebuje ime sheme. Sheme podatkovne baze smo opredelili z enostavnim SQL stavkom, ki ga vpišemo v *PgAdminov SQL* urejevalnik.



Slika 35: PgAdminIII - administracijski program baze Postgres.

Ko smo imeli sheme za posamezne države izdelane, smo začeli z uvozom podatkov v podatkovno bazo. Uvoz podatkov smo izvajali preko *Postgres* konzole za vsak podatkovni sloj posamezne države posebej - z enostavnim ukazom:

```
shp2pgsql -c <ime_shape_datoteke> <ime_sheme_pb> <ime_podatkovne_baze>
```

Ko so bili vsi podatki zapisani v *Postgres* bazi, je bilo potrebno prostorske podatke še ustrezno indeksirati. Indeks je podatkovna struktura, ki izboljša hitrost nad operacijami v tabeli. Indeksiramo vsa tista polja v posameznih podatkovnih tabelah, na podlagi katerih bomo kasneje (pri samem izrisu podatkov) izdelovali različne poizvedbe. Zaradi hitrejšega prikazovanja geometrije pa indeksiramo tudi samo geometrijo pojavov. Indekse v bazo

zapišemo z enostavnim SQL ukazom, ki ga zapišemo v *PgAdminov* SQL urejevalnik. Posamezen indeks opredelimo z enoličnim imenom indeksa, tabelo in stolpcem tabele, na katerega se nanaša indeks ter tipom indeksa (*btree*, *gist*, idr.). Primer opredelitve indeksa za podatkovni sloj *Streets* (Slovenija) ter stolpec "*st_name*" opisuje naslednji SQL stavek:

```
CREATE INDEX "Index_slv_streets_st_name" ON slv.streets USING btree (st_name);
```

8.5 Nastavitve na »Geoserverju«

Potem ko smo uvozili podatke v *Postgres* bazo, je bilo potrebno opraviti ustrezne nastavitve na *Geoserverju*. Najprej smo zagnali *Geoserver* in se prijavili. Za vsako državo smo nato posebej definirali svoj imenski prostor (*Namespace*) z ukazom *Config -> Data -> Namespace*. Imenski prostor smo definirali s tričrkovno kratico države, pred katero je bila predpona "*eu_*" (za Slovenijo npr. *eu_svn*). Za vsako državo posebej smo nato definirali še podatkovno skladišče (*Data Store*). Definiranje podatkovnega skladišča pomeni določitev nekaterih parametrov, ki jih *Geoserver* potrebuje za dostop do baze, kjer so shranjeni naši podatki. Vpisali smo podatkovni tip podatkov (*PostGIS*), imenski prostor, IP številko in port strežnika, kjer se nahaja podatkovna baza, ime sheme podatkovne baze, ime podatkovne baze ter ime uporabnika podatkovne baze. Definiranje podatkovnega skladišča za primer Slovenije prikazuje slika 36.

The screenshot shows the GeoServer web interface. On the left, there is a sidebar with a 'Data' section containing a table of data stores:

Data:	
GeoServer	mar 17, 2:53 PM
Configuration	mar 17, 2:53 PM
XML	mar 17, 2:53 PM

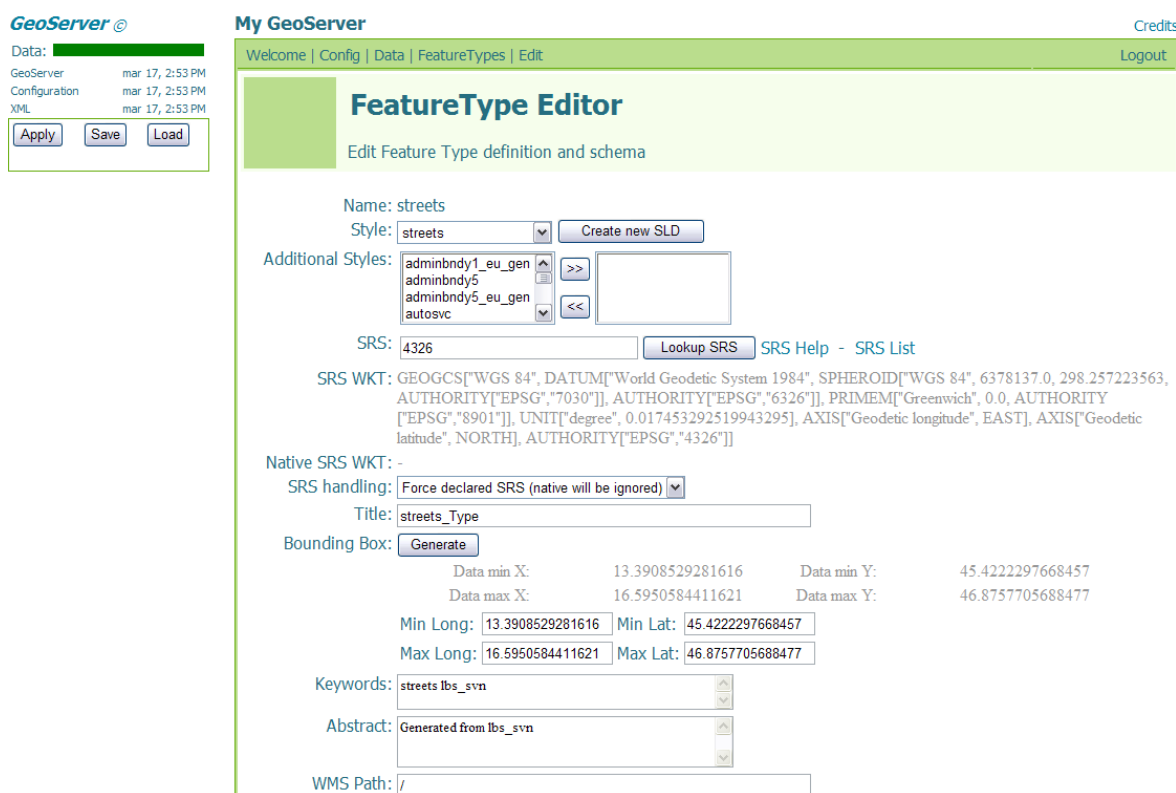
Below the table are buttons for 'Apply', 'Save', and 'Load'. The main content area is titled 'My GeoServer' and 'Feature Data Set Editor'. It contains the following configuration fields:

- Feature Data Set ID: lbs_svn
- Enabled:
- Namespace: eu_svn (dropdown)
- Description: (text input)
- * host: 192.168.0.128
- * port: 5432
- schema: svn
- * database: eu_karto
- * user: postgres
- passwd: (password input)
- wkb enabled: true (dropdown)
- loose bbox: true (dropdown)
- estimated extent: false (dropdown)

At the bottom, there are 'Submit' and 'Reset' buttons, and a note: '* = required field'.

Slika 36: Definiranje podatkovnega skladišča na »Geoserverju«, za primer Slovenije.

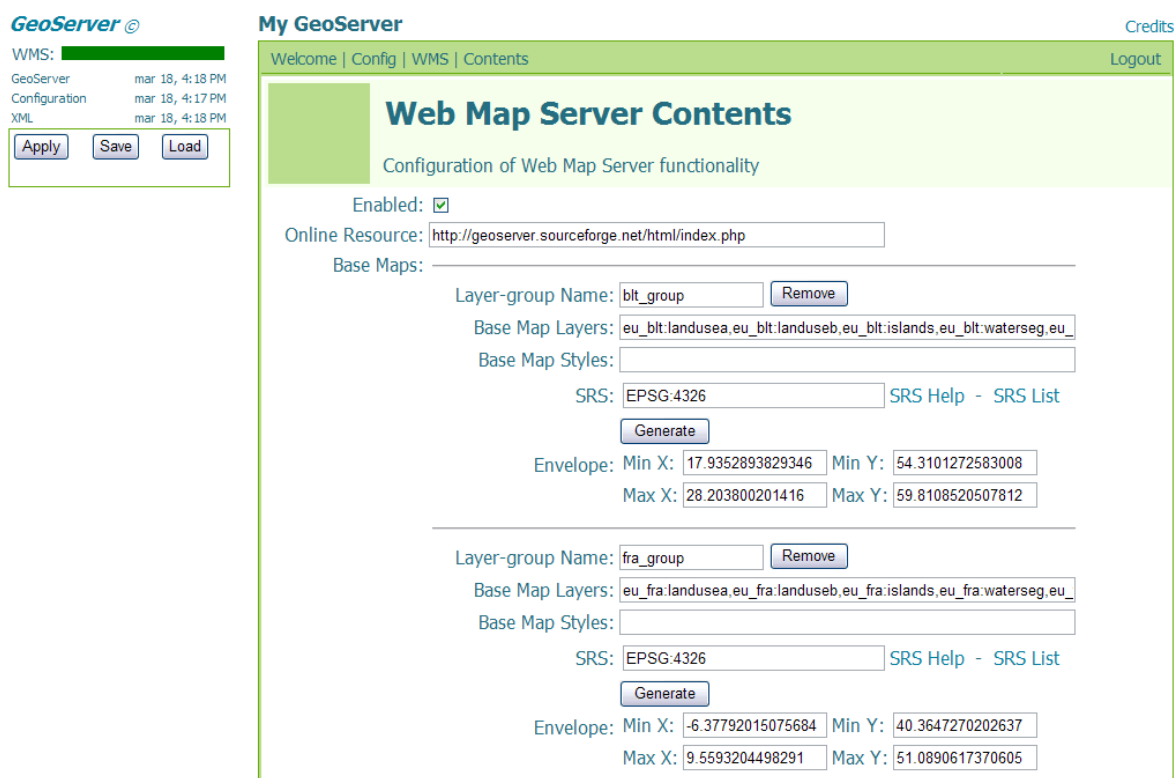
Naprej je potrebno uvoziti oblikovalske sloge, ki smo jih za posamezne podatkovne sloje že predhodno zapisali v SLD datoteke. Vsakemu oblikovalskemu slugu smo za uporabo na *Geoserverju* dodelili ime (*Style ID*), ki je (zaradi preglednosti) enako kot ime podatkovnega sloja, na katerega se nanaša. V oknu *Style Editor* smo nato SLD datoteke naložili na *Geoserver*. V naslednji fazi pa je bilo potrebno posamezne sloge še povezati z ustreznimi podatkovnimi sloji. To smo storili v oknu *Feature Type Editor*. Tu smo definirali še koordinatno osnovo (*SRS – Spatial Reference System*) naših podatkov in določili t.i. *Bounding Box* – območje prikaza oz. mejne koordinate podatkov (največja/najmanjša geografska širina/dolžina). Z ukazom *Submit* smo potrdili nastavitve, z gumboma *Apply* in *Save* pa smo nastavitve shranili na *Geoserver*. Nastavitve v oknu *Feature Type Editor* prikazuje slika 37.



Slika 37: Nastavitve v »Geoserverjevem« oknu »FeatureTypeEditor«.

Zadnja nastavitvev na *Geoserverju* je bila definiranje podatkovnih skupin v oknu *Web Map Server Contents* (slika 38). Definiranje podatkovnih skupin pomeni, da sloje, ki so si sorodni (npr. po prostorski lokaciji, tipu podatkovnega sloja) združimo v neko celoto.

Razdelitev v podatkovne skupine je bila potrebna zaradi velike količine podatkovnih slojev na *Geoserverju*. Z definiranjem podatkovnih skupin smo tako izboljšali preglednost nad podatki, ki jih streže *Geoserver*. Do podatkov bomo kasneje dostopali lažje in bolj organizirano. Podatkovne sloje smo združevali v skupine glede na prostorsko lokacijo. Za vsako državo posebej smo definirali skupino, ki smo jo poimenovali s tričrkovno kratico države. Našteli smo vse podatkovne sloje posamezne države in jih združili v skupino. Pri tem smo upoštevali tudi vrstni red posameznih podatkovnih slojev. Najprej smo našteli sloje, ki se bodo tudi na karti prikazovali prvi in tako naprej. *Geoserver* smo tako z zadnjo fazo nastavitve pripravili za strežbo naših prostorskih podatkov.



Slika 38: Definiranje podatkovnih skupin na »Geoserverju«.

8.6 Opredelitev matematičnih elementov karte

V nadaljevanju opisujemo še matematične elemente navigacijske karte Evrope.

Kartografska projekcija

Pri izdelavi navigacijske karte je pomembno, da je uporabljena kartografska projekcija konformna. V našem primeru smo se odločili za uporabo (pokončne) *Mercatorjeve* projekcije, pri kateri ne prihaja do deformacij kotov. Vzporedniki in meridiani se preslikajo kot linije in se med seboj sekajo pod pravim kotom. Vsi meridiani so obrnjeni v smeri sever-jug, vzporedniki pa v smeri vzhod-zahod. Te dobre lastnosti so zelo pomembne pri izdelavi navigacijske karte saj je podajanje navodil za navigacijo na ta način precej poenostavljeno. Dobre lastnosti *Mercatorjeve* projekcije so v tem primeru pomembnejše od dobro poznane slabosti projekcije – t.j. velikih deformacij površin, ki nastanejo, ko se približujemo poloma. *Mercatorjeva* projekcija naj bi bila uporabna nekje do geografske širine 70° in za področje Evrope je to še ravno zadovoljiv razpon.

Koordinatni sistem karte

- geodetski datum: WGS-84,
- kartografska projekcija: (pokončna) *Mercatorjeva* projekcija,
- geografska dolžina srednjega meridiana: $\lambda_0 = 0$ (Greenwich),
- geografska širina izhodiščne paralele: $\phi_0 = 0$ (Ekvator),
- navidezni pomik proti severu ("*false easting*"): 0 m,
- navidezni pomik proti vzhodu ("*false northing*"): 0 m,
- x os je usmerjena proti vzhodu, y os je usmerjena proti severu

Merila karte

Navigacijska karta Evrope bo na voljo v naslednjih (v naprej določenih) merilih:

1 : 34.000.000, 1 : 12.000.000, 1 : 3.500.000, 1 : 1.200.000, 1 : 600.000, 1 : 180.000,
1 : 70.000, 1 : 25.000, 1 : 12.000, 1 : 5000, 1 : 2500, 1 : 1000

Območja prikaza (koordinate roba karte)

Spodnji levi rob:

x = -2.400.000 m

y = 3.100.000 m

Zgornji desni rob:

$x = 8.000.000$ m

$y = 11.500.000$ m

8.7 Izdelava spletne strani

Naše karte želimo v končni fazi predstaviti spletnemu uporabniku in zato je potrebno na koncu izdelati ustrezno spletno stran, kjer bo naša spletna karta prikazana. Vsebino in obliko spletne strani opredelimo znotraj HTML dokumenta. V HTML dokument vpnemo *OpenLayers Java* skripto, znotraj katere naprej opredelimo obliko in vsebino karte ter obliko in vsebino pregledovalnika karte. *OpenLayers* funkcije (ki omogočajo postavitev pregledovalnika za karte na spletno stran) kličemo iz zunanje *Java* skripte. V HTML dokumentu smo zato najprej navedli pot do te datoteke. Nato smo opredelili tudi pot (URL) do kartografskega strežnika, od koder bomo črpali naše oblikovane podatkovne sloje.

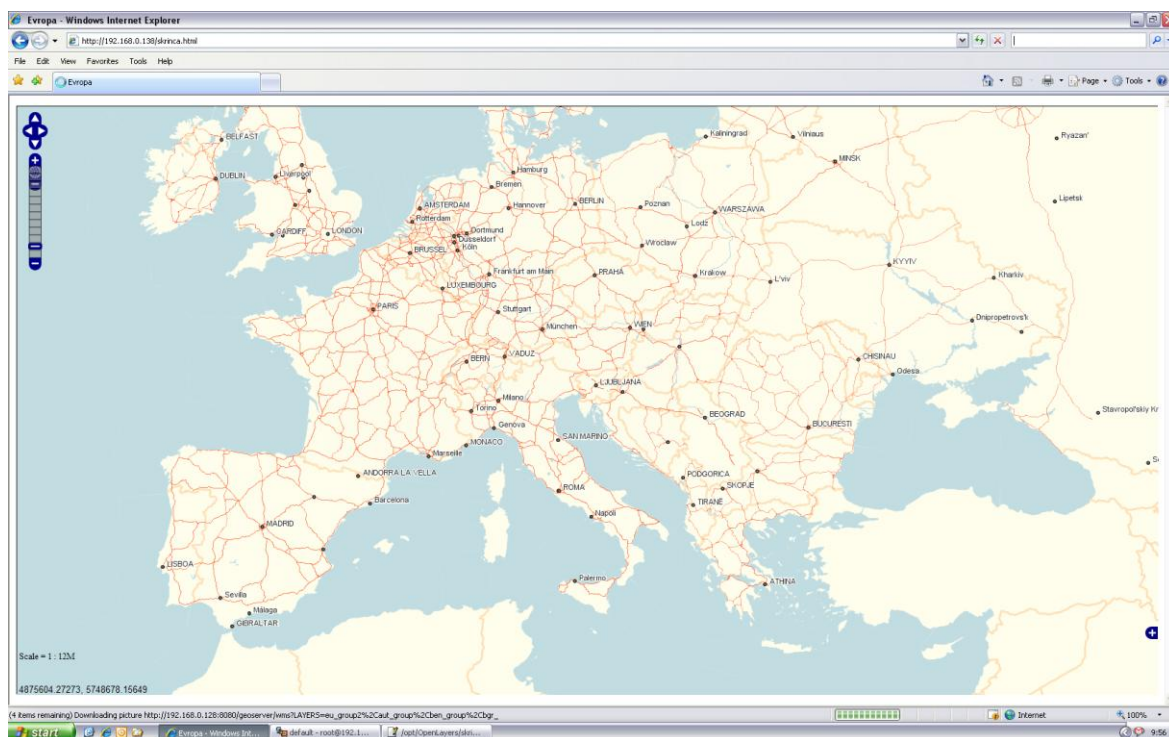
Obliko in vsebino karte smo nato znotraj *Java* skripte opredelili z naslednjimi elementi:

- območje prikaza (koordinate roba karte),
- prostorski referenčni sistem,
- merske enote na karti,
- seznam podatkovnih slojev/podatkovnih skupin, ki jih želimo prikazati,
- format vrnjene slikovne datoteke.

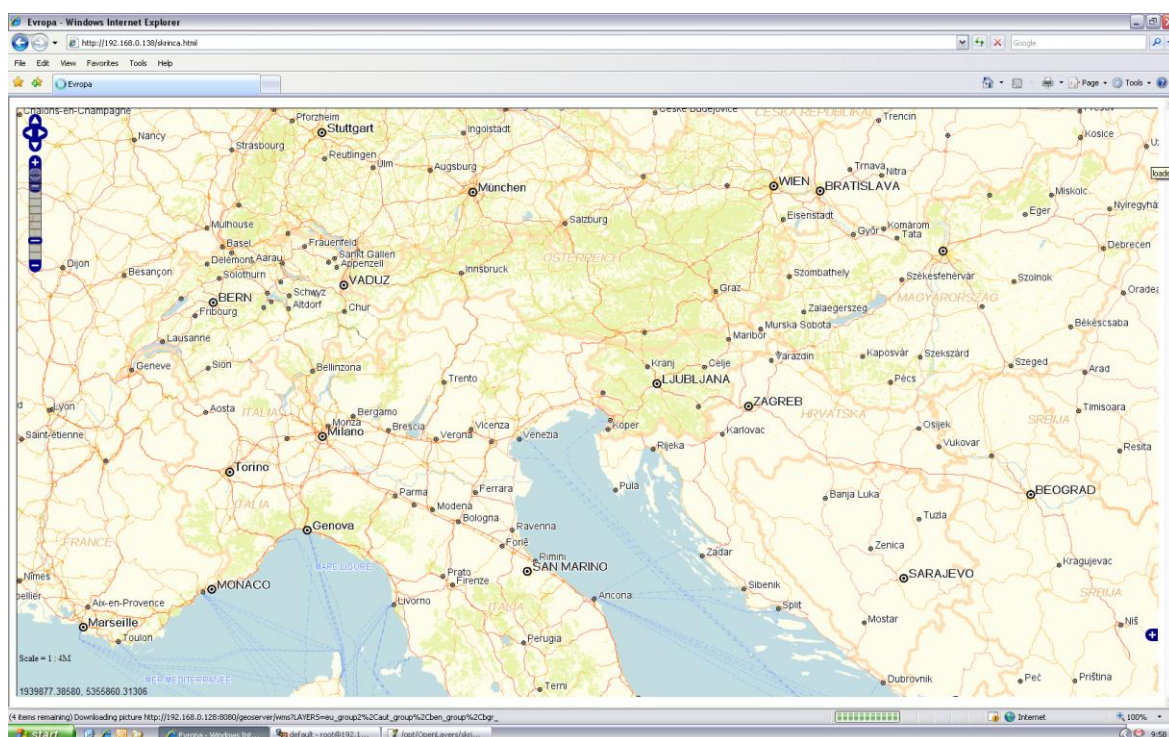
Obliko in vsebino pregledovalnika karte pa smo opredelili z naslednjimi elementi:

- oblika (velikost, debelina, barva) okvirja karte,
- kontrole, ki jih želimo na karti (*pan/zoom* vrstica, pomanjšan pogled karte, merilo karte, koordinate položaja miškinega kurzorja).

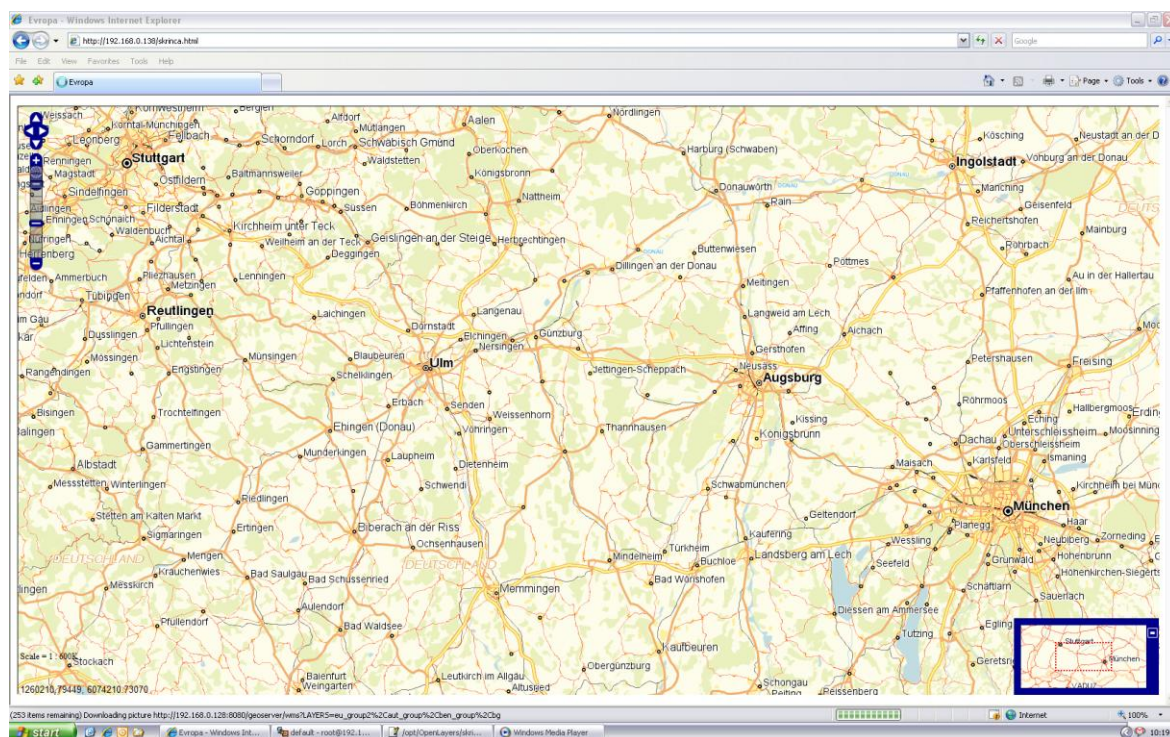
Zgradba enostavnega HTML dokumenta je prikazana in opisana v prilogi A. Slike 39 – 44 pa v različnih merilih prikazujejo končni izgled spletne karte.



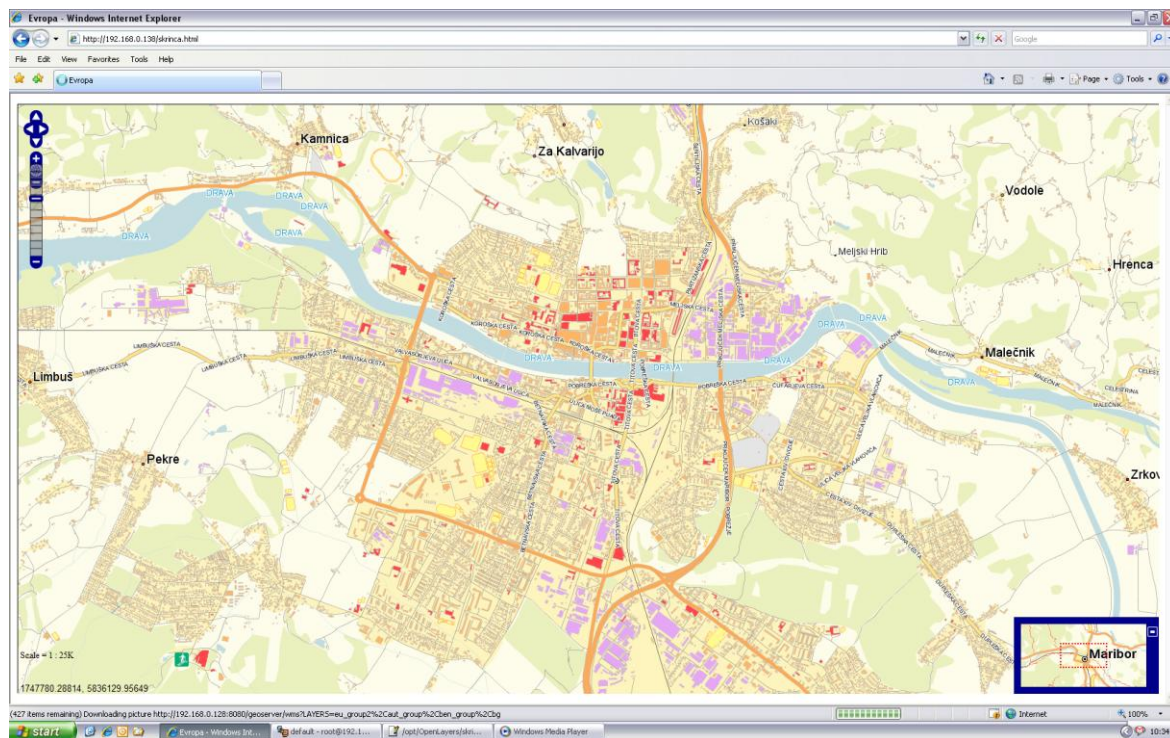
Slika 39: Spletna karta Evrope: primer 1.



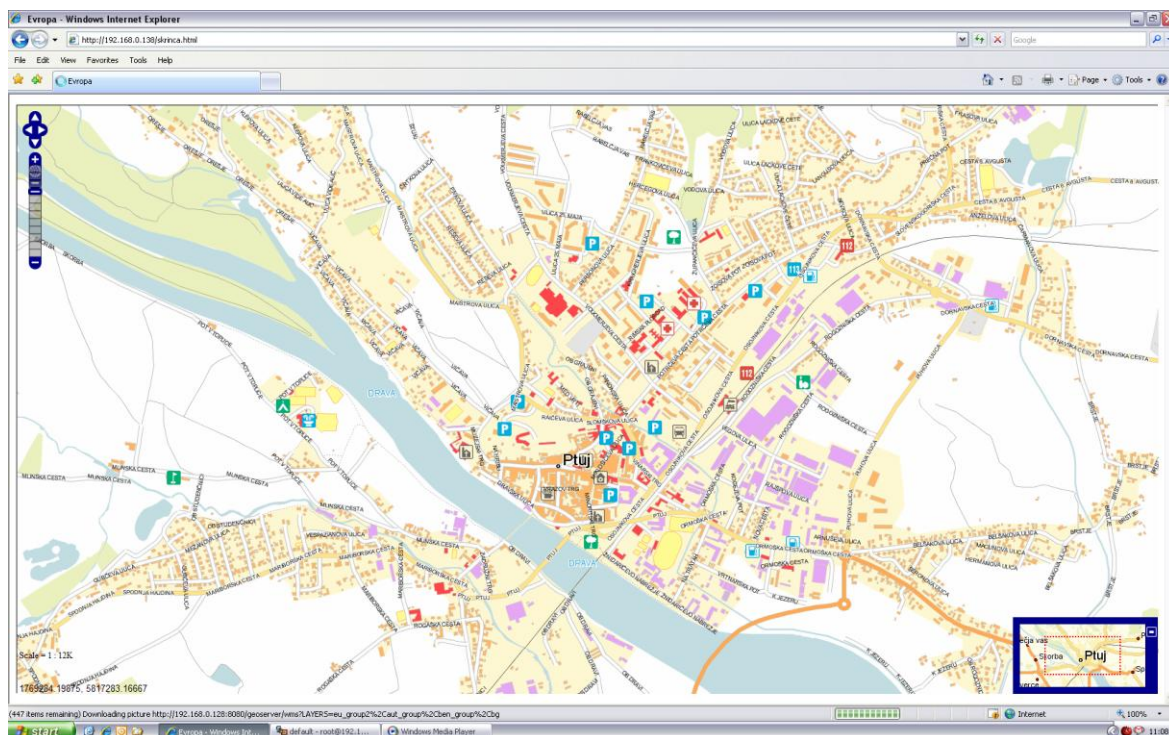
Slika 40: Spletna karta Evrope: primer 2.



Slika 41: Spletna karta Evrope: primer 3.



Slika 43: Spletna karta Evrope: primer 4.



Slika 44: Spletna karta Evrope: primer 5.

9 ZAKLJUČEK

Spletna karta Evrope bo v končni fazi delovala na spletnem portalu www.yellowlbs.com in bo v popolnosti zaživela predvidoma jeseni 2008. Portal bo uporabnikom omogočal brezplačno poizvedovanje po lokaciji (glede na naslov, interesne točke, mesta) in izračune najhitrejših/najkrajših poti med različnimi lokacijami v Evropi. Registriranim naročnikom sistema Sledenje (sistem, ki omogoča sledenje vozil na podlagi GPS naprave vgrajene v vozilo) pa bo portal omogočal tudi spremljanje vozil na karti Evrope in še nekatere druge funkcionalnosti v zvezi s sledenjem vozila.

Portal *YellowLbs* je skupni projekt podjetij Sledenje in Monolit. V slednjem sem zaposlen tudi sam in del diplomske naloge je bil tudi del mojih rednih zaposlitev v podjetju. Podjetje Monolit med drugim razpolaga z vektorskima podatkovnima bazama *StreetConnect* in *Slo Topo*. Podatki iz teh dveh baz so integrirani v proizvode priznanih proizvajalcev, ki se ukvarjajo s cestnonavigacijskimi sistemi in kartografijo – *Navteq, Garmin, Mio-Tech, Avmap, WayFinder, Magellan*. V karti Evrope smo za področje Slovenije tako uporabili podatke baze *StreetConnect*, ki smo jih pred tem pretvorili v *Navstreets* format. Karta pa se bo v prihodnosti še nadgrajevala z dodatno vsebino baze *Slo Topo*. Predviden je prikaz topografskih vsebin kot so plastnice, vrhovi, planinske in pohodniške poti, kolesarske, konjeniške poti, vinske ceste. Podatki v bazi se bodo predvidoma posodabljali štirikrat letno, tako da bo uporabnikom vedno na voljo relativno ažurna karta.

Portal *YellowLbs* bo med drugim služil predstavitvi baze *StreetConnect* ter baze *Slo Topo*. Zaradi tega je pomembno, da se te karte strežejo preko standardnega WMS protokola, saj bodo tako zanimivejše precej širšemu krogu potencialnih naročnikov kart kot bi to bilo pri nestandardni rešitvi (ki je bila v uporabi do sedaj). Prav tako bomo lahko v prihodnosti na teh kartah enostavno prikazovali podatke, ki jih ponujajo katerikoli oddaljeni javni oz. komercialni WMS strežniki. Enostavno bomo lahko združevali podatke iz porazdeljenih strežnikov, ki na primer ponujajo informacije o lokacijah prometnih nesreč, stanju na cestah, različne tematske sloje, ortofoto in satelitske posnetke (npr. iz javnega WMS strežnika *WorldWind*).

Z uporabo odprtokodne programske opreme za spletno kartografijo smo prihranili. Stroškov povezanih z nakupom programske opreme namreč nismo imeli. Zaradi odprtosti programske kode imamo možnost prilagoditve programske opreme našim specifičnim potrebam, ki se lahko pojavijo v prihodnosti. S sodelovanjem v diskusijah na spletnih forumih pa imamo tudi možnost sooblikovanja programske opreme, ki jo uporabljamo. Nenazadnje tudi prehod na drugo programsko opremo v prihodnosti ne bo tako težaven, kot bi bil pri uporabi lastniške programske opreme ali nestandardnih rešitev.

VIRI

Strokovne publikacije:

Burrough, Peter A. 1998. Principles of geographical information systems. Oxford, Oxford University Press, str. 12.

Fridl J. 1999. Metodologija tematske kartografije nacionalnega atlasa Slovenije. Ljubljana, Geografski inštitut Antona Melika ZRC SAZU, str. 98, 91-92, 113.

Gartner G., Cartwright W., Peterson M.P. 1999. Multimedia Cartography. Springer, Berlin, Heidelberg, New York, London.

Hächler T. 2003. Online Visualization of Spatial Data, A prototype of an open source internet map server backend spatial database for the Swiss national park. Department of Geography, University of Zurich, str. 12-13.

<http://www.webgis.ch/downloads/WebGIS.pdf>

Kvamme K., Oštir-Sedej K., Stančič Z., Šumrada R. 1997. Geografski informacijski sistemi. Ljubljana, Znanstvenoraziskovalni center SAZU, str. 207, 290-291.

Mac Gillavry E. 2000. Cartographic aspects of WebGIS-software. Department of Cartography, Utrecht University, str. 12.

Nasr, M.R. 2007. OpenSource Software: The Use of open source GIS software and its impact on organizations. Middlesex University, School of Computing Science, str. 2, 57.

http://opensource.mit.edu/papers/OS_GIS.pdf

Navteq. Navstreets Street Data. Version 3.6.0.

Open Geospatial Consortium inc. 2005. OpenGIS Filter Encoding Implementation Specification, Version 1.1.0.

<http://www.opengeospatial.org/standards/filter>

Open Geospatial Consortium inc. 2005. OpenGIS Web Feature Service Implementation Specification, Version 1.1.0.

<http://www.opengeospatial.org/standards/wfs>

Open Geospatial Consortium inc. 2006. Web Coverage Service (WCS) Implementation Specification, Version 1.1.0. 2006.

<http://www.opengeospatial.org/standards/wcs>

Open Geospatial Consortium inc. 2006. OpenGIS Web Map Server Implementation Specification, Version 1.3.0.

<http://www.opengeospatial.org/standards/wms>

OpenGIS Consortium inc. 2002. Styled Layer Descriptor Implementation Specification, Version 1.0.0.

<http://www.opengeospatial.org/standards/sld>

OpenGIS Consortium inc. 2003. The Importance of Going Open. 2003. An Open GIS Consortium White Paper.

https://www.seegrid.csiro.au/twiki/pub/Xmml/DataModels/20030923_openWP.pdf

Peterca M. 2001. Matematična kartografija, Kartografske projekcije. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo, str. 31-32.

Peterson M. P. 2005. Maps and the internet. Amsterdam, London, Elsevier, cop, str. 1, 3.

Radovan D. 2003. Kartografija II - zapiski iz predavanj, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo.

Saccon R. 2003. Possibilities and Limits of Open Source Software. Swiss Business School, str. 14, 31.

<http://s3.amazonaws.com/rsaccon/rsaccons-MBA-thesis.pdf>

Saidin H. 2007. All for open standards.

<http://www.openmalaysiablog.com/2007/10/all-for-open-st.html> (18.12.2007)

Schütze E. 2007. Current state of technology and potential of smart map browsing in web browsers. Bremen University of Applied Sciences, str. 17.

http://www.smartmapbrowsing.org/thesis_EmanuelSchuetze.pdf

Šumrada R. 2005a. Tehnologija GIS. Ljubljana, Univerza v Ljubljani, Fakulteta za gradbeništvo in geodezijo, str. 9, 86, 110-111, 138, 207, 211, 214.

Šumrada R. 2005b. Osrednja in porazdeljena uporaba GIS tehnologije. GIS tehnologija - skripta, PDF format, verzija: 2005-01-03.

Trinidad G, Cole I., Chan W.Y. 2000. Developing Internet-based GIS Applications. Proceedings of the Philippine Computing Science Congress 2000, str. 184-186.

<http://citeseer.ist.psu.edu/cache/papers/cs/23106/http:zSzzSzccslinux.dlsu.edu.phzSzczspzSzdocszSzproceedingszSzposterszSzdeveloping.pdf/trinidad00developing.pdf>

Spletne strani:

Internet 1: <http://www.geog.ubc.ca/courses/klink/gis.notes/ncgia/u10.html> (13.1.2008)

Internet 2: <http://news.netcraft.com> (10.12.2007)

Internet 3: <http://www.linux.com/articles/53407> (8.12.2007)

Internet 4:

http://lyceum.massgis.state.ma.us/wiki/doku.php?id=wms:simple_example:home

(9.1.2008)

Internet 5: <http://sourceforge.net> (10.12.2007)

Internet 6: <http://www.kartografie.nl/webcartography/webbook/> (12.5.2008)

<http://geoserver.org>

<http://geotools.codehaus.org>

<http://maps.yahoo.com>

<http://openlayers.org>

<http://postgis.refractions.net>

<http://www.answers.com>

<http://www.navteq.com>

<http://www.opengeospatial.org>

<http://www.opensource.org>

<http://www.pocketgpsworld.com>

<http://www.wikipedia.org>

PRILOGA – A:

Zgradba enostavnega HTML dokumenta z vpeto OpenLayers Java skripto, ki omogoča postavitev dinamične karte na spletno stran.

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN">
<html>
<head>
  <!--definiranje velikosti in oblike okvirja karte-->
  <style type="text/css">
    #map
    {
      width: 100%;
      height: 100%;
      border: 1px solid black;
    }
  </style>

  <!--definiranje poti do zunanje OpenLayers java skripte-->
  <script src="/libs/OpenLayers24_dev/lib/OpenLayers.js"></script>
  <title>Evropa</title>
</head>
<body>
  <div id="map"></div>
  <script type="text/javascript">

    //deklaracija nekaterih spremenljivk
    var mercatorMap,mercatorMapOptions,mercatorLayer;

    //definiranje poti do kartografskega strežnika
    var serverUrl = ["http://192.168.0.128:8080/geoserver/wms"];

    // definiranje območja prikaza, kartografske projekcije, merskih enot, definiranje meril
    var maxMercatorExtent = new OpenLayers.Bounds(-2400000,3100000,8000000,11500000);
    var mercatorPrj = 'EPSG:41001';
    var mercatorUnits = 'm';
    var mercatorScales =
    [32000000,12000000,3500000,1200000,600000,180000,70000,25000,12000,5000,2500,1000];

    mercatorMapOptions = {maxExtent:maxMercatorExtent,scales:mercatorScales,
    units:mercatorUnits, projection:mercatorPrj}

    //definiranje podatkovnih slojev/skupin, ki jih želimo prikazati na karti
    var layersList = "aut_group,ben_group,bgr_group,blk_group,blt_group,cze_group,deu_group,
    esp_group,fra_group,gbr_group,grc_group,hrv_group,hun_group,ita_group,
    mub_group,pol_group,rou_group,rus_group,scn_group,srb_group,svk_group,
    svn_group";

    //sestavljanje karte
    mercatorMap = new OpenLayers.Map ('map',mercatorMapOptions);
    mercatorLayer = new OpenLayers.Layer.WMS
    (
    "EU",
    serverUrl,
```

```
{layers: layersList, format: "image/png8"}
);




mercatorMap.addLayers([mercatorLayer]);

//definiranje kontrol, ki jih želimo prikazati na karti (pan/zoom orodje, numerično merilo,
//koordinatne pozicije miške, pomanjšani pogled karte)
mercatorMap.addControl(new OpenLayers.Control.PanZoomBar());
mercatorMap.addControl(new OpenLayers.Control.Scale($('scale')));
mercatorMap.addControl(new OpenLayers.Control.MousePosition());
mercatorMap.addControl(new OpenLayers.Control.OverviewMap({mapOptions:
mercatorMapOptions}));
mercatorMap.zoomToMaxExtent();
</script>
</body>
</html>
```


PRILOGA - B:

Knjižnica kartografskih znakov




AdminBndy1





Razpon merila	barva polnila	barva linije/obrobe	debelina linije/obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
FeatType = country						
1 : 34 milj. - 1 : 15 milj.	R: 255 G: 253 B: 223					
1 : 15 milj. – 1 : 5 milj.	R: 255 G: 253 B: 223			R: 234 G: 180 B: 106	Arial 13 oblique	
1 : 5 milj. - 1 : 1 milj.	R: 255 G: 253 B: 223			R: 234 G: 180 B: 106	Arial 16 oblique	

AdminBndy5



Razpon merila	barva polnila	barva linije/obrobe	debelina linije/obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
1 : 5 milj. – 1 : 1 milj.	R: 255 G: 241 B: 159					
1 : 1 milj. – 1 : 1.000	R: 255 G: 244 B: 187					

Cartocountry


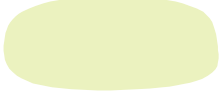
Razpon merila	barva polnila	barva linije/obrobe	debelina linije/obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
1 : 34 milj. – 1 : 5 milj.		R: 255 G: 225 B: 185	2 px			
1 : 5 milj. - 1 : 2 milj.		R: 255 G: 225 B: 185 R: 255 G: 210 B: 151	4 px 1 px			
1 : 2 milj. - 1 : 1 milj.		R: 255 G: 225 B: 185 R: 255 G: 210 B: 151	6 px 2 px			




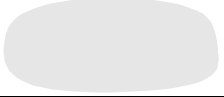
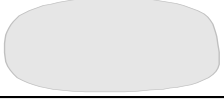


1 : 1 milj. – 1 : 250.000		R: 255 G: 225 B: 185	9 px			
		R: 255 G: 210 B: 151	4 px, vzorec: 3,5,12,5			
1 : 250.000 – 1 : 100.000		R: 255 G: 225 B: 185	12 px			
		R: 255 G: 210 B: 151	6 px, vzorec: 3,5,12,5			
1 : 100.000 – 1 : 50.000		R: 255 G: 225 B: 185	16 px			
		R: 255 G: 210 B: 151	8 px, vzorec: 3,5,12,5			
1 : 50.000 - 1 : 1.000		R: 255 G: 225 B: 185	20 px			
		R: 255 G: 210 B: 151	12 px, vzorec: 3,5,12,5			

Islands



Razpon merila	barva polnila	barva linije/obrobe	debelina linije/obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
1 : 34 milj. – 1 : 500.000	R: 250 G: 253 B: 242					
1 : 500.000 - 1 : 1.000	R: 250 G: 253 B: 242			R: 192 G: 192 B: 192	Arial 12 normal	

LandUseA




Razpon merila	barva polnila	barva linije/obrobe	debelina linije/obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
feat_type = WOODLAND						
1 : 5 milj. – 1 : 1 milj.	R: 233 G: 243 B: 171					
1 : 1 milj. – 1 : 1.000	R: 235 G: 242 B: 191					

feat_type = PARK IN WATER						
1 : 5 milj. – 1 : 1.000	R: 194 G: 221 B: 226					
feat_type = AIRPORT						
1 : 250.000 – 1 : 50.000	R: 255 G: 244 B: 187					
1 : 50.000 – 1 : 1.000	R: 255 G: 244 B: 187	R: 254 G: 200 B: 0	1 px			
feat_type = CEMETERY						
1 : 250.000 – 1 : 50.000	R: 230 G: 230 B: 230					
1 : 50.000 – 1 : 1.000	R: 230 G: 230 B: 230	R: 192 G: 192 B: 192	1 px			
feat_type = SPORTS COMPLEX						
1 : 250.000 – 1 : 50.000	R: 255 G: 252 B: 147					
1 : 50.000 – 1 : 1.000	R: 255 G: 252 B: 147	R: 255 G: 145 B: 34	1 px			



LandUseB

Razpon merila	barva polnila	barva linije/obrobe	debelina linije/obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
feat_type = GOLF COURSE						
1 : 250.000 – 1 : 1.000	R: 210 G: 236 B: 196					
feat_type = AIRCRAFT ROADS						
1 : 250.000 – 1 : 1.000	R: 200 G: 200 B: 200					






Landmark

Razpon merila	barva polnila	barva linije/obrobe	debelina linije/obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
feat_type = RESIDENTIAL or HISTORICAL or PARK/LEISURE or SPORTS or TOURIST or TRANSPORTATION						
1 : 25.000 – 1 : 1.000	R: 255 G: 183 B: 102	R: 192 G: 192 B: 192	1 px			
feat_type = GOVERNMENT or CULTURAL or EDUCATION or MEDICAL						
1 : 25.000 – 1 : 1.000	R: 255 G: 70 B: 74	R: 192 G: 192 B: 192	1 px			
feat_type = BUSINESS/COMMERCE						
1 : 25.000 – 1 : 1.000	R: 231 G: 168 B: 247	R: 192 G: 192 B: 192	1 px			

Waterpoly

Razpon merila	barva polnila	barva linije/obrobe	debelina linije/obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
1 : 1 milj. – 1 : 100.000	R: 194 G: 221 B: 226					
1 : 100.000 – 1 : 1.000	R: 194 G: 221 B: 226			R: 36 G: 160 B: 208	Arial 12 normal Halo: 1 px	

Waterseg



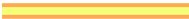





Razpon merila	barva polnila	barva linije/obrobe	debelina linije/obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
feat_type = RIVER						
1 : 250.000 – 1 : 50.000		R: 194 G: 221 B: 226	1 px			
1 : 50.000 – 1 : 25.000		R: 194 G: 221 B: 226	2 px			
1 : 25.000 – 1 : 10.000		R: 194 G: 221 B: 226	3 px			
1 : 10.000 – 1 : 5.000		R: 194 G: 221 B: 226	6 px	R: 36 G: 160 B: 208	Arial 8 normal Halo: 1 px	
1 : 5.000 – 1 : 1.000		R: 194 G: 221 B: 226	12 px	R: 36 G: 160 B: 208	Arial 12 normal Halo: 1px	












feat_type = WATER CHANNEL						
1 : 50.000 – 1 : 25.000		R: 194 G: 221 B: 226	1 px			
1 : 25.000 – 1 : 10.000		R: 194 G: 221 B: 226	2 px			
1 : 10.000 – 1 : 5.000		R: 194 G: 221 B: 226	3 px	R: 36 G: 160 B: 208	Arial 7 normal Halo: 1 px	
1 : 5.000 – 1 : 1.000		R: 194 G: 221 B: 226	6 px	R: 36 G: 160 B: 208	Arial 8 normal Halo: 1 px	

Oceans

Razpon merila	barva polnila	barva linije/obrobe	debelina linije/obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
1 : 34 milj. – 1 : 10 milj.	R: 194 G: 221 B: 226					
1 : 10 milj. – 1 : 5 milj.	R: 194 G: 221 B: 226			R: 91 G: 136 B: 255	Arial 10 normal Halo: 1 px	
1 : 5 milj. – 1 : 1 milj.	R: 194 G: 221 B: 226			R: 91 G: 136 B: 255	Arial 12 normal Halo: 1 px	
1 : 1 milj. – 1 : 250.000	R: 194 G: 221 B: 226			R: 91 G: 136 B: 255	Arial 16 normal Halo: 1 px	
1 : 250.000 – 1 : 50.000	R: 194 G: 221 B: 226			R: 91 G: 136 B: 255	Arial 22 normal Halo: 1 px	
1 : 50.000 – 1 : 1.000	R: 194 G: 221 B: 226					

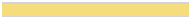
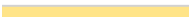
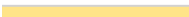













Majhwys

Razpon merila	barva polnila	barva linije/ obrobe	debelina linije/ obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
Ferrytype = H and Func_class = 1						
1 : 34 milj. – 1 : 5 milj.		R: 255 G: 100 B: 55	1 px			
1 : 5 milj. – 1 : 1 milj.		R: 255 G: 89 B: 60	1 px			
1 : 1 milj. – 1 : 250.000		R: 255 G: 169 B: 83 R: 247 G: 255 B: 138	4 px 2 px			
1 : 250.000 – 1 : 50.000		R: 255 G: 169 B: 83 R: 247 G: 255 B: 138	6 px 4 px			
1 : 50.000 – 1 : 10.000		R: 255 G: 169 B: 83 R: 247 G: 255 B: 138	8 px 6 px			
1 : 10.000 – 1 : 5.000		R: 255 G: 169 B: 83 R: 247 G: 255 B: 138	10 px 8 px			
1 : 5.000 – 1 : 1.000		R: 255 G: 169 B: 83 R: 247 G: 255 B: 138	14 px 12 px			
Ferrytype = H and Func_class = 2						
1 : 5 milj. – 1 : 1 milj.		R: 255 G: 145 B: 0	1 px			

1 : 1 milj. – 1 : 250.000		R: 210 G: 210 B: 210	3 px			
		R: 255 G: 172 B: 89	2 px			
1 : 250.000 – 1 : 50.000		R: 210 G: 210 B: 210	5 px			
		R: 255 G: 169 B: 83	4 px			
1 : 50.000 – 1 : 10.000		R: 210 G: 210 B: 210	7 px			
		R: 255 G: 169 B: 83	6 px			
1 : 10.000 – 1 : 5.000		R: 210 G: 210 B: 210	9 px			
		R: 255 G: 169 B: 83	7 px			
1 : 5.000 – 1 : 1.000		R: 210 G: 210 B: 210	12 px			
		R: 255 G: 169 B: 83	10 px			
Ferrytype = R and Func_class = 2						
1 : 5 milj. - 1 : 100.000		R: 0 G: 0 B: 0	1 px vzorec: 1,3,5,3			
1 : 100.000 – 1 : 10.000		R: 0 G: 0 B: 0	2 px vzorec: 1,3,5,3			
1 : 10.000 – 1 : 1.000		R: 0 G: 0 B: 0	4 px vzorec: 1,3,5,3			
Ferrytype = B and Func_class = 2						
1 : 5 milj. - 1 : 100.000		R: 91 G: 136 B: 255	1 px vzorec: 1,3,5,3			
1 : 100.000 – 1 : 10.000		R: 91 G: 136 B: 255	2 px vzorec: 1,3,5,3			
1 : 10.000 – 1 : 1.000		R: 91 G: 136 B: 255	4 px vzorec: 1,3,5,3			



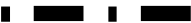
Sechwys

Razpon merila	barva polnila	barva linije/ obrobe	debelina linije/ obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
Ferrytype = H and Func_class = 3						
1 : 1 milj. – 1 : 250.000		R: 255 G: 100 B: 55	1 px			
1 : 250.000 - 1 : 100.000		R: 255 G: 216 B: 79	2 px			
1 : 100.000 - 1 : 50.000		R: 210 G: 210 B: 210	4 px			
		R: 255 G: 228 B: 136	3 px			
1 : 50.000 – 1 : 25.000		R: 210 G: 210 B: 210	5 px			
		R: 255 G: 228 B: 136	4 px			
1 : 25.000 – 1 : 10.000		R: 210 G: 210 B: 210	6 px	R: 0 G: 0 B: 0	Arial 8 normal	
		R: 255 G: 228 B: 136	5 px			
1 : 10.000 – 1 : 5.000		R: 210 G: 210 B: 210	8 px	R: 0 G: 0 B: 0	Arial 10 normal	
		R: 255 G: 228 B: 136	6 px			
1 : 5.000 - 1 : 1.000		R: 210 G: 210 B: 210	12 px	R: 0 G: 0 B: 0	Arial 12 normal	
		R: 255 G: 228 B: 136	10 px			
Ferrytype = H and Func_class = 4						
1 : 500.000 - 1 : 100.000		R: 192 G: 192 B: 192	1 px			

1 : 100.000 - 1 : 50.000		R: 210 G: 210 B: 210	4 px			
		R: 255 G: 228 B: 136	3 px			
1 : 50.000 - 1 : 25.000		R: 210 G: 210 B: 210	5 px			
		R: 255 G: 228 B: 136	4 px			
1 : 25.000 – 1 : 10.000		R: 210 G: 210 B: 210	6 px	R: 0 G: 0 B: 0	Arial 8 normal	
		R: 255 G: 228 B: 136	5 px			
1 : 10.000 – 1 : 5.000		R: 210 G: 210 B: 210	8 px	R: 0 G: 0 B: 0	Arial 10 normal	
		R: 255 G: 228 B: 136	6 px			
1 : 5.000 - 1 : 1.000		R: 210 G: 210 B: 210	12 px	R: 0 G: 0 B: 0	Arial 12 normal	
		R: 255 G: 228 B: 136	10 px			
Ferrytype = B						
1 : 5 milj. - 1 : 100.000		R: 91 G: 136 B: 255	1 px vzorec: 1,3,5,3			
1 : 100.000 – 1 : 10.000		R: 91 G: 136 B: 255	2 px vzorec: 1,3,5,3			
1 : 10.000 – 1 : 1.000		R: 91 G: 136 B: 255	4 px vzorec: 1,3,5,3			
Ferrytype = R						
1 : 5 milj. - 1 : 100.000		R: 0 G: 0 B: 0	1 px vzorec: 1,3,5,3			
1 : 100.000 – 1 : 10.000		R: 0 G: 0 B: 0	2 px vzorec: 1,3,5,3			
1 : 10.000 – 1 : 1.000		R: 0 G: 0 B: 0	4 px vzorec: 1,3,5,3			




Streets






Razpon merila	barva polnila	barva linije/obrobe	debelina linije/obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
Ferrytype = H and Func_class = 5 and Paved = Y						
1 : 100.000 - 1 : 50.000		R: 192 G: 192 B: 192	1 px			
1 : 50.000 - 1 : 25.000		R: 127 G: 127 B: 127	1 px			
Ferrytype = H and Func_class = 5 and Paved = Y and *filter1						
1 : 25.000 - 1 : 10.000		R: 192 G: 192 B: 192 R: 255 G: 255 B: 255	6 px 4 px	R: 0 G: 0 B: 0	Arial 8 normal	
1 : 10.000 - 1 : 5.000		R: 192 G: 192 B: 192 R: 255 G: 255 B: 255	8 px 6 px	R: 0 G: 0 B: 0	Arial 10 normal	
1 : 5.000 - 1 : 1.000		R: 192 G: 192 B: 192 R: 255 G: 255 B: 255	12 px 10 px	R: 0 G: 0 B: 0	Arial 12 normal	
Ferrytype = H and Func_class = 5 and Paved = N and *filter1						
1 : 25.000 - 1 : 5.000		R: 127 G: 127 B: 127	1 px			
1 : 5.000 - 1 : 1.000		R: 192 G: 192 B: 192	2 px			
Ferrytype = H and Func_class = 5 and *filter2						
1 : 25.000 - 1 : 5.000		R: 127 G: 127 B: 127	1 px vzorec: 2,2			
1 : 5000 - 1 : 1.000		R: 192 G: 192 B: 192	2 px vzorec: 2,2			
Ferrytype = B						
1 : 100.000 - 1 : 10.000		R: 91 G: 136 B: 255	2 px vzorec: 1,3,5,3			

1 : 10.000 – 1 : 1.000		R: 91 G: 136 B: 255	4 px vzorec: 1,3,5,3			
Ferrytype = R						
1 : 100.000 – 1 : 10.000		R: 0 G: 0 B: 0	2 px vzorec: 1,3,5,3			
1 : 10.000 – 1 : 1.000		R: 0 G: 0 B: 0	4 px vzorec: 1,3,5,3			






* filter1 = (ar_auto= Y) or (ar_bus= Y) or (ar_taxis= Y) or (ar_trucks= Y) or (ar_traff= Y) or (ar_deliv= Y) or (ar_emerveh= Y)
 * filter2 = (ar_auto = N) and (ar_bus = N) and (ar_taxis = N) and (ar_trucks = N) and (ar_traff = N) and (ar_deliv = N) and (ar_emerveh = N)








NamedPlc








Razpon merila	barva polnila	barva linije/obrobe	debelina linije/obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
capital = 1						
1 : 34 milj. – 1 : 15 milj.		R: 50 G: 50 B: 50 R: 255 G: 255 B: 255	5 px 1 px	R: 50 G: 50 B: 50	Arial 11 normal	
1 : 15 milj. – 1 : 5 milj.		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255 R: 0 G: 0 B: 0	8 px 4 px 2 px	R: 0 G: 0 B: 0	Arial 12 normal	
1 : 5 milj. – 1 : 1 milj.		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255 R: 0 G: 0 B: 0	12 px 8 px 4 px	R: 0 G: 0 B: 0	Arial 16 bold	

1 : 1 milj. - 1 : 500.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255 R: 0 G: 0 B: 0	14 px 10 px 6 px			
1 : 500.000 – 1 : 100.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255 R: 0 G: 0 B: 0	14 px 10 px 6 px			
1 : 100.000 – 1 : 25.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255 R: 0 G: 0 B: 0	14 px 10 px 6 px			
capital ≠ 1 and population > 500.000						
1 : 34 milj. – 1 : 15 milj.		R: 50 G: 50 B: 50 R: 255 G: 255 B: 255	5 px 1 px	R: 50 G: 50 B: 50	Arial 11 normal	
1 : 15 milj. – 1 : 5 milj.		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255 R: 0 G: 0 B: 0	8 px 4 px 2 px		Arial 12 normal	

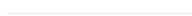





1 : 5 milj. – 1 : 1 milj.		R: 0 G: 0 B: 0	12 px			
		R: 255 G: 255 B: 255	8 px	R: 0 G: 0 B: 0	Arial 16 bold	
		R: 0 G: 0 B: 0	4 px			
1 : 1 milj. - 1 : 500.000		R: 0 G: 0 B: 0	14 px			
		R: 255 G: 255 B: 255	10 px	R: 0 G: 0 B: 0	Arial 18 bold	
		R: 0 G: 0 B: 0	6 px			
1 : 500.000 – 1 : 100.000		R: 0 G: 0 B: 0	14 px			
		R: 255 G: 255 B: 255	10 px	R: 0 G: 0 B: 0	Arial 22 bold	
		R: 0 G: 0 B: 0	6 px			
1 : 100.000 – 1 : 25.000		R: 0 G: 0 B: 0	14 px			
		R: 255 G: 255 B: 255	10 px	R: 0 G: 0 B: 0	Arial 28 bold	
		R: 0 G: 0 B: 0	6 px			
capital = 2 or (100.000 < population ≤ 500.000)						
1 : 5 milj. – 1 : 2 milj.		R: 0 G: 0 B: 0	6 px			
		R: 255 G: 255 B: 255	4 px	R: 0 G: 0 B: 0	Arial 14 normal	
		R: 0 G: 0 B: 0	2 px			

1 : 2 milj. – 1 : 1 milj.		R: 0 G: 0 B: 0	8 px			
		R: 255 G: 255 B: 255	6 px	R: 0 G: 0 B: 0	Arial 16 normal	
		R: 0 G: 0 B: 0	3 px			
1 : 1 milj. - 1 : 500.000		R: 0 G: 0 B: 0	8 px			
		R: 255 G: 255 B: 255	6 px	R: 0 G: 0 B: 0	Arial 18 bold	
		R: 0 G: 0 B: 0	3 px			
1 : 500.000 - 1 : 100.000		R: 0 G: 0 B: 0	8 px			
		R: 255 G: 255 B: 255	6 px	R: 0 G: 0 B: 0	Arial 20 bold	
		R: 0 G: 0 B: 0	3 px			
1 : 100.000 - 1 : 25.000		R: 0 G: 0 B: 0	8 px			
		R: 255 G: 255 B: 255	6 px	R: 0 G: 0 B: 0	Arial 26 bold	
		R: 0 G: 0 B: 0	3 px			
capital ≠ 1 and capital ≠ 2 and (25.000 < population ≤ 100.000)						
1 : 2 milj. – 1 : 1 milj.		R: 0 G: 0 B: 0	5 px			
		R: 255 G: 255 B: 255	2 px	R: 0 G: 0 B: 0	Arial 12 normal	

1 : 1 milj. - 1 : 500.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255	5 px 2 px	R: 0 G: 0 B: 0	Arial 15 normal	◦ 
1 : 500.000 - 1 : 100.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255	5 px 2 px	R: 0 G: 0 B: 0	Arial 18 normal	◦ 
1 : 100.000 - 1 : 25.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255	5 px 2 px	R: 0 G: 0 B: 0	Arial 24 normal	◦ 
1 : 25.000 – 1 : 10.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255	5 px 2 px	R: 0 G: 0 B: 0	Arial 24 bold	◦ 
capital = 3 or (25.000 < population ≤ 5000)						
1 : 1 milj. - 1 : 500.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255	5 px 2 px	R: 0 G: 0 B: 0	Arial 14 normal	◦ 
1 : 500.000 - 1 : 250.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255	5 px 2 px	R: 0 G: 0 B: 0	Arial 16 normal	◦ 
1 : 250.000 - 1 : 100.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255	5 px 2 px	R: 0 G: 0 B: 0	Arial 18 normal	◦ 

1 : 100.000 - 1 : 25.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255	5 px 2 px	R: 0 G: 0 B: 0	Arial 22 normal	◦ 
1 : 25.000 – 1 : 10.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255	5 px 2 px	R: 0 G: 0 B: 0	Arial 22 bold	◦ 
capital = 4 or (5000 < population ≤ 1000)						
1 : 250.000 – 1 : 100.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255	4 px 2 px	R: 0 G: 0 B: 0	Arial 11 normal	◦ 
1 : 100.000 – 1 : 25.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255	4 px 2 px	R: 0 G: 0 B: 0	Arial 14 normal	◦ 
1 : 25.000 – 1 : 10.000		R: 0 G: 0 B: 0 R: 255 G: 255 B: 255	4 px 2 px	R: 0 G: 0 B: 0	Arial 18 normal	◦ 
capital = 5 or capital = »« or population ≤ 1000						
1 : 100.000 – 1 : 25.000		R: 0 G: 0 B: 0	2 px	R: 0 G: 0 B: 0	Arial 12 normal	• 
1 : 25.000 – 1 : 10.000		R: 0 G: 0 B: 0	2 px	R: 0 G: 0 B: 0	Arial 12 normal	• 


RailRds

Razpon merila	barva polnila	barva linije/ obrobe	debelina linije/ obrobe	barva napisa	napis (družina, velikost, stil)	Grafični primer
Tunnel = N						
1 : 1 milj. – 1 : 25.000		R: 0 G: 0 B: 0	1 px			
1 : 25.000 – 1 : 10.000		R: 80 G: 80 B: 80	2 px			
1 : 10.000 – 1 : 1.000		R: 80 G: 80 B: 80	3 px			
Tunnel = Y						
1 : 1 milj. – 1 : 25.000		R: 0 G: 0 B: 0	1 px vzorec: 2,2			
1 : 25.000 – 1 : 10.000		R: 80 G: 80 B: 80	2 px vzorec: 2,2			
1 : 10.000 – 1 : 1.000		R: 80 G: 80 B: 80	3 px vzorec: 2,2			









PRILOGA C:


Knjižnica kartografskih znakov za podatkovne sloje POI

Hospitals





Razpon merila	Velikost (px)	Grafični primer
FeatType = 8060 or FeatType = 9583 (bolnišnica, zdravstveni dom)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	

ParkRec





Razpon merila	Velikost (px)	Grafični primer
FeatType = 7996 (zabavišni park)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 7992 (igrišče za golf)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 8410 (muzej)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 7940 or FeatType = 7997 (športni center, športni kompleks)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 4493 (marina)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 4580 (športno letališče)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 7933 (bowling center)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 7947 (park)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	







FeatType = 7998 (drsališče)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	

TranHubs




Razpon merila	Velikost (px)	Grafični primer
FeatType = 4581 (mednarodno letališče)		
1 : 250.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 4170 (avtobusna postaja)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 4013 or FeatType = 4100 (železniška postaja)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 4482 (trajekt)		
1 : 50.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	

TravDest



Razpon merila	Velikost (px)	Grafični primer
FeatType = 7011 (hotel)		
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 7999 (turistična atrakcija)		
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 7389 (turistične informacije)		
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 7990 (sejem)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	

FeatType = 2084 (vinoteka)		
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 7012 (smučišče)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 7897 (počivališče)		
1 : 50.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 5999 (zgodovinski spomenik)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 7510 (rent-a-car)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 7013 (gostišče/turistična kmetija)		
1 : 2.500 – 1 : 1.000	30	





Shopping

Razpon merila	Velikost (px)	Grafični primer
FeatType = 6512 or FeatType = 5400 (trgovina)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 9995 (knjigarna)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 9565 (lekarna)		
1 : 2.500 – 1 : 1.000	30	



Restrnts




Razpon merila	Velikost (px)	Grafični primer
FeatType = 5800 (restavracija)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 9996 (kavarna)		
1 : 2.500 – 1 : 1.000	30	

Entertn



Razpon merila	Velikost (px)	Grafični primer
FeatType = 5813 (nočni klub)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 7929 (odrske umetnoti)		
1 : 10.000 – 1 : 2.500	25	
1 : 2500 – 1 : 1000	30	
FeatType = 7832 (kino)		
1 : 10.000 – 1 : 2.500	25	
1 : 2500 – 1 : 1000	30	
FeatType = 9532 (bar)		
1 : 2.500 – 1 : 1.000	30	

AutoSvc



Razpon merila	Velikost (px)	Grafični primer
FeatType = 5540 (bencinska črpalka)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 7538 (avtomehanična delavnica)		
1 : 2.500 – 1 : 1.000	30	

FeatType = 5511 or FeatType = 5512 (prodajalec avtomobilov)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 8699 (avto klub)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 5571 (prodajalec motornih koles)		
1 : 2.500 – 1 : 1.000	30	











FinInsts

Razpon merila	Velikost (px)	Grafični primer
FeatType = 6000 (banka)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 3578 (bankomat)		
1 : 2.500 – 1 : 1.000	30	




Business

Razpon merila	Velikost (px)	Grafični primer
FeatType = 5000 (poslovna stavba)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 9991 (industrijski objekt)		
1 : 2.500 – 1 : 1.000	30	




CommSvc

Razpon merila	Velikost (px)	Grafični primer
FeatType = 7994 (komunalni center)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 9121 (mestna hiša)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 9211 (sodišče)		
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 9221 (policijska postaja)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 9992 and Bld_type = "MOSQUE" (mošeja)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 9992 and Bld_type = "CHURCH" (cerkev)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 9992 and Bld_type = "TEMPLE" (tempelj)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 9992 and Bld_type = "SYNAGOGUE" (sinagoga)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 9993 (veleposlaništvo)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 9994 (vladna ustanova)		
1 : 2.500 – 1 : 1.000	30	


EduInsts

Razpon merila	Velikost (px)	Grafični primer
FeatType = 8200 (visoka šola)		
1 : 10.000 – 1 : 2.500	25	
1 : 2500 – 1 : 1000	30	
FeatType = 8211 (šola)		
1 : 2.500 – 1 : 1.000	30	
FeatType = 8231 (knjižnica)		
1 : 2.500 – 1 : 1.000	30	


Parking

Razpon merila	Velikost (px)	Grafični primer
FeatType = 7520 (parkirišče)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 7521 (parkirna hiša)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 7522 (park & ride)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	






BordCross

Razpon merila	Velikost (px)	Grafični primer
FeatType = 9999 (mejni prehod)		
1 : 250.000 – 1 : 50.000	15	
1 : 50.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	

PostOffice

Razpon merila	Velikost (px)	Grafični primer
FeatType = 9530 (pošta)		
1 : 2.500 – 1 : 1.000	30	

MiscCategories

Razpon merila	Velikost (px)	Grafični primer
FeatType = 7591 (pokopališče)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 1001 (toplice)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 1002 (kamp)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 1003 (postajališče za avtodom)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	
FeatType = 1004 (gasilci)		
1 : 20.000 – 1 : 10.000	20	
1 : 10.000 – 1 : 2.500	25	
1 : 2.500 – 1 : 1.000	30	

PRILOGA - Č:

SLD datoteka za primer podatkovnega sloja »Streets«

```
<?xml version="1.0" encoding="UTF-8"?>
<sld:UserStyle xmlns:sld="http://www.opengis.net/sld"
xmlns:ogc="http://www.opengis.net/ogc" xmlns:gml="http://www.opengis.net/gml">
<sld:Name>Default Styler</sld:Name>
<sld:Title>Default Styler</sld:Title>
<sld:Abstract></sld:Abstract>

<sld:FeatureTypeStyle>
<sld:Name>name</sld:Name>
<sld:Title>title</sld:Title>
<sld:Abstract>abstract</sld:Abstract>
<sld:FeatureTypeName>streets</sld:FeatureTypeName>
<sld:SemanticTypeIdentifier>generic:geometry</sld:SemanticTypeIdentifier>

<sld:Rule>
  <sld:Name>name</sld:Name>
  <sld:Title>title</sld:Title>
  <sld:Abstract>Abstract</sld:Abstract>

  <ogc:Filter>
    <ogc:And>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>func_class</ogc:PropertyName>
        <ogc:Literal>5</ogc:Literal>
      </ogc:PropertyIsEqualTo>

      <ogc:Or>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_auto</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_bus</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_taxis</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_trucks</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_traff</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_deliv</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_emerveh</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:Or>

      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>paved</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
      </ogc:PropertyIsEqualTo>

      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ferry_type</ogc:PropertyName>
        <ogc:Literal>H</ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:And>
  </ogc:Filter>
</sld:Rule>
</sld:UserStyle>
```

```
</ogc:Filter>

<sld:MinScaleDenominator>50000.0</sld:MinScaleDenominator>
<sld:MaxScaleDenominator>100000.0</sld:MaxScaleDenominator>

<sld:LineStyleSymbolizer>
  <sld:Stroke>
    <sld:CssParameter name="stroke">
      <ogc:Literal>#C0C0C0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-linecap">
      <ogc:Literal>square</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-linejoin">
      <ogc:Literal>round</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-opacity">
      <ogc:Literal>1.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-width">
      <ogc:Literal>1.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-dashoffset">
      <ogc:Literal>0.0</ogc:Literal>
    </sld:CssParameter>
  </sld:Stroke>
</sld:LineStyleSymbolizer>
</sld:Rule>

<sld:Rule>

  <sld:Name>name</sld:Name>
  <sld:Title>title</sld:Title>
  <sld:Abstract>Abstract</sld:Abstract>

  <ogc:Filter>
    <ogc:And>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>func_class</ogc:PropertyName>
        <ogc:Literal>5</ogc:Literal>
      </ogc:PropertyIsEqualTo>

      <ogc:Or>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_auto</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_bus</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_taxis</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_trucks</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_traff</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_deliv</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_emerveh</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:Or>
    </ogc:And>
  </ogc:Filter>
</sld:Rule>
```

```
        </ogc:PropertyIsEqualTo>
    </ogc:Or>

    <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>paved</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
    </ogc:PropertyIsEqualTo>

    <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ferry_type</ogc:PropertyName>
        <ogc:Literal>H</ogc:Literal>
    </ogc:PropertyIsEqualTo>
</ogc:And>
</ogc:Filter>

<sld:MinScaleDenominator>25000.0</sld:MinScaleDenominator>
<sld:MaxScaleDenominator>50000.0</sld:MaxScaleDenominator>

<sld:LineSymbolizer>
    <sld:Stroke>
        <sld:CssParameter name="stroke">
            <ogc:Literal>#808080</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-linecap">
            <ogc:Literal>square</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-linejoin">
            <ogc:Literal>round</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-opacity">
            <ogc:Literal>1.0</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-width">
            <ogc:Literal>1.0</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-dashoffset">
            <ogc:Literal>0.0</ogc:Literal>
        </sld:CssParameter>
    </sld:Stroke>
</sld:LineSymbolizer>
</sld:Rule>

<sld:Rule>
    <sld:Name>name</sld:Name>
    <sld:Title>title</sld:Title>
    <sld:Abstract>Abstract</sld:Abstract>

    <ogc:Filter>
        <ogc:And>
            <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>func_class</ogc:PropertyName>
                <ogc:Literal>5</ogc:Literal>
            </ogc:PropertyIsEqualTo>

            <ogc:And>
                <ogc:PropertyIsEqualTo>
                    <ogc:PropertyName>ar_auto</ogc:PropertyName>
                    <ogc:Literal>N</ogc:Literal>
                </ogc:PropertyIsEqualTo>
                <ogc:PropertyIsEqualTo>
                    <ogc:PropertyName>ar_bus</ogc:PropertyName>
                    <ogc:Literal>N</ogc:Literal>
                </ogc:PropertyIsEqualTo>
                <ogc:PropertyIsEqualTo>
                    <ogc:PropertyName>ar_taxis</ogc:PropertyName>
                    <ogc:Literal>N</ogc:Literal>
                </ogc:PropertyIsEqualTo>
                <ogc:PropertyIsEqualTo>
                    <ogc:PropertyName>ar_trucks</ogc:PropertyName>
                    <ogc:Literal>N</ogc:Literal>
                </ogc:PropertyIsEqualTo>
            </ogc:And>
        </ogc:Filter>
    </sld:Rule>
```

```

        <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>ar_traff</ogc:PropertyName>
            <ogc:Literal>N</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>ar_deliv</ogc:PropertyName>
            <ogc:Literal>N</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
            <ogc:PropertyName>ar_emerveh</ogc:PropertyName>
            <ogc:Literal>N</ogc:Literal>
        </ogc:PropertyIsEqualTo>
    </ogc:And>

    <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ferry_type</ogc:PropertyName>
        <ogc:Literal>H</ogc:Literal>
    </ogc:PropertyIsEqualTo>
</ogc:And>
</ogc:Filter>

<sld:MinScaleDenominator>5000.0</sld:MinScaleDenominator>
<sld:MaxScaleDenominator>25000.0</sld:MaxScaleDenominator>

<sld:LineSymbolizer>
    <sld:Stroke>
        <sld:CssParameter name="stroke">
            <ogc:Literal>#808080</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-linecap">
            <ogc:Literal>butt</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-linejoin">
            <ogc:Literal>miter</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-opacity">
            <ogc:Literal>1.0</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-width">
            <ogc:Literal>1.0</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-dashoffset">
            <ogc:Literal>0.0</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-dasharray">2.0 2.0</sld:CssParameter>
    </sld:Stroke>
</sld:LineSymbolizer>
</sld:Rule>

<sld:Rule>
    <sld:Name>name</sld:Name>
    <sld:Title>title</sld:Title>
    <sld:Abstract>Abstract</sld:Abstract>

    <ogc:Filter>
        <ogc:And>
            <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>func_class</ogc:PropertyName>
                <ogc:Literal>5</ogc:Literal>
            </ogc:PropertyIsEqualTo>

            <ogc:Or>
                <ogc:PropertyIsEqualTo>
                    <ogc:PropertyName>ar_auto</ogc:PropertyName>
                    <ogc:Literal>Y</ogc:Literal>
                </ogc:PropertyIsEqualTo>
                <ogc:PropertyIsEqualTo>
                    <ogc:PropertyName>ar_bus</ogc:PropertyName>
                    <ogc:Literal>Y</ogc:Literal>
                </ogc:PropertyIsEqualTo>
            </ogc:Or>
        </ogc:And>
    </ogc:Filter>
</sld:Rule>
```

```

        <ogc:PropertyName>ar_taxis</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
    </ogc:PropertyIsEqualTo>
    <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ar_trucks</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
    </ogc:PropertyIsEqualTo>
    <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ar_traff</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
    </ogc:PropertyIsEqualTo>
    <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ar_deliv</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
    </ogc:PropertyIsEqualTo>
    <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ar_emerveh</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
    </ogc:PropertyIsEqualTo>
</ogc:Or>

    <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>paved</ogc:PropertyName>
        <ogc:Literal>N</ogc:Literal>
    </ogc:PropertyIsEqualTo>

    <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ferry_type</ogc:PropertyName>
        <ogc:Literal>H</ogc:Literal>
    </ogc:PropertyIsEqualTo>
</ogc:And>
</ogc:Filter>

<sld:MinScaleDenominator>5000.0</sld:MinScaleDenominator>
<sld:MaxScaleDenominator>25000.0</sld:MaxScaleDenominator>

<sld:LineSymbolizer>
    <sld:Stroke>
        <sld:CssParameter name="stroke">
            <ogc:Literal>#808080</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-linecap">
            <ogc:Literal>square</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-linejoin">
            <ogc:Literal>round</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-opacity">
            <ogc:Literal>1.0</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-width">
            <ogc:Literal>1.0</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-dashoffset">
            <ogc:Literal>0.0</ogc:Literal>
        </sld:CssParameter>
    </sld:Stroke>
</sld:LineSymbolizer>
</sld:Rule>

<sld:Rule>
    <sld:Name>name</sld:Name>
    <sld:Title>title</sld:Title>
    <sld:Abstract>Abstract</sld:Abstract>

    <ogc:Filter>
        <ogc:And>
            <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>func_class</ogc:PropertyName>
                <ogc:Literal>5</ogc:Literal>
            </ogc:PropertyIsEqualTo>

```

```
<ogc:Or>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>ar_auto</ogc:PropertyName>
    <ogc:Literal>Y</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>ar_bus</ogc:PropertyName>
    <ogc:Literal>Y</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>ar_taxis</ogc:PropertyName>
    <ogc:Literal>Y</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>ar_trucks</ogc:PropertyName>
    <ogc:Literal>Y</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>ar_traff</ogc:PropertyName>
    <ogc:Literal>Y</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>ar_deliv</ogc:PropertyName>
    <ogc:Literal>Y</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>ar_emerveh</ogc:PropertyName>
    <ogc:Literal>Y</ogc:Literal>
  </ogc:PropertyIsEqualTo>
</ogc:Or>

<ogc:PropertyIsEqualTo>
  <ogc:PropertyName>paved</ogc:PropertyName>
  <ogc:Literal>Y</ogc:Literal>
</ogc:PropertyIsEqualTo>

<ogc:PropertyIsEqualTo>
  <ogc:PropertyName>ferry_type</ogc:PropertyName>
  <ogc:Literal>H</ogc:Literal>
</ogc:PropertyIsEqualTo>
</ogc:And>
</ogc:Filter>

<sld:MinScaleDenominator>10000.0</sld:MinScaleDenominator>
<sld:MaxScaleDenominator>25000.0</sld:MaxScaleDenominator>

<sld:LineSymbolizer>
  <sld:Stroke>
    <sld:CssParameter name="stroke">
      <ogc:Literal>#C0C0C0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-linecap">
      <ogc:Literal>butt</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-linejoin">
      <ogc:Literal>round</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-opacity">
      <ogc:Literal>1.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-width">
      <ogc:Literal>6.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-dashoffset">
      <ogc:Literal>0.0</ogc:Literal>
    </sld:CssParameter>
  </sld:Stroke>
</sld:LineSymbolizer>

<sld:LineSymbolizer>
  <sld:Stroke>
    <sld:CssParameter name="stroke">
```

```
        <ogc:Literal>#FFFFFF</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="stroke-linecap">
        <ogc:Literal>square</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="stroke-linejoin">
        <ogc:Literal>round</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="stroke-opacity">
        <ogc:Literal>1.0</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="stroke-width">
        <ogc:Literal>4.0</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="stroke-dashoffset">
        <ogc:Literal>0.0</ogc:Literal>
      </sld:CssParameter>
    </sld:Stroke>
  </sld:LineSymbolizer>

  <sld:TextSymbolizer>
    <sld:Label>
      <ogc:PropertyName>st_name</ogc:PropertyName>
    </sld:Label>

    <sld:Font>
      <sld:CssParameter name="font-family">
        <ogc:Literal>Arial</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="font-size">
        <ogc:Literal>8.0</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="font-style">
        <ogc:Literal>normal</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="font-weight">
        <ogc:Literal>normal</ogc:Literal>
      </sld:CssParameter>
    </sld:Font>

    <sld:LabelPlacement>
      <sld:LinePlacement>
        <sld:PerpendicularOffset>
          <ogc:Literal>0.0</ogc:Literal>
        </sld:PerpendicularOffset>
      </sld:LinePlacement>
    </sld:LabelPlacement>

    <sld:Fill>
      <sld:CssParameter name="fill">
        <ogc:Literal>#000000</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="fill-opacity">
        <ogc:Literal>1.0</ogc:Literal>
      </sld:CssParameter>
    </sld:Fill>
  </sld:TextSymbolizer>
</sld:Rule>

<sld:Rule>
  <sld:Name>name</sld:Name>
  <sld:Title>title</sld:Title>
  <sld:Abstract>Abstract</sld:Abstract>

  <ogc:Filter>
    <ogc:And>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>func_class</ogc:PropertyName>
        <ogc:Literal>5</ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:And>
  </ogc:Filter>
</sld:Rule>
```



```
<ogc:And>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>ar_auto</ogc:PropertyName>
    <ogc:Literal>N</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>ar_bus</ogc:PropertyName>
    <ogc:Literal>N</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>ar_taxis</ogc:PropertyName>
    <ogc:Literal>N</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>ar_trucks</ogc:PropertyName>
    <ogc:Literal>N</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>ar_traff</ogc:PropertyName>
    <ogc:Literal>N</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>ar_deliv</ogc:PropertyName>
    <ogc:Literal>N</ogc:Literal>
  </ogc:PropertyIsEqualTo>
  <ogc:PropertyIsEqualTo>
    <ogc:PropertyName>ar_emerveh</ogc:PropertyName>
    <ogc:Literal>N</ogc:Literal>
  </ogc:PropertyIsEqualTo>
</ogc:And>

<ogc:PropertyIsEqualTo>
  <ogc:PropertyName>ferry_type</ogc:PropertyName>
  <ogc:Literal>H</ogc:Literal>
</ogc:PropertyIsEqualTo>
</ogc:And>
</ogc:Filter>

<sld:MinScaleDenominator>1000.0</sld:MinScaleDenominator>
<sld:MaxScaleDenominator>5000.0</sld:MaxScaleDenominator>

<sld:LineStyleSymbolizer>
  <sld:Stroke>
    <sld:CssParameter name="stroke">
      <ogc:Literal>#C0C0C0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-linecap">
      <ogc:Literal>butt</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-linejoin">
      <ogc:Literal>miter</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-opacity">
      <ogc:Literal>1.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-width">
      <ogc:Literal>2.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-dashoffset">
      <ogc:Literal>0.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-dasharray">3.0 3.0 </sld:CssParameter>
  </sld:Stroke>
</sld:LineStyleSymbolizer>
</sld:Rule>

<sld:Rule>
  <sld:Name>name</sld:Name>
  <sld:Title>title</sld:Title>
  <sld:Abstract>Abstract</sld:Abstract>
```

```
<ogc:Filter>
  <ogc:And>
    <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>func_class</ogc:PropertyName>
      <ogc:Literal>5</ogc:Literal>
    </ogc:PropertyIsEqualTo>
    <ogc:Or>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ar_auto</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ar_bus</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ar_taxis</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ar_trucks</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ar_traff</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ar_deliv</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ar_emerveh</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:Or>
    <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>paved</ogc:PropertyName>
      <ogc:Literal>N</ogc:Literal>
    </ogc:PropertyIsEqualTo>
    <ogc:PropertyIsEqualTo>
      <ogc:PropertyName>ferry_type</ogc:PropertyName>
      <ogc:Literal>H</ogc:Literal>
    </ogc:PropertyIsEqualTo>
  </ogc:And>
</ogc:Filter>

<sld:MinScaleDenominator>1000.0</sld:MinScaleDenominator>
<sld:MaxScaleDenominator>5000.0</sld:MaxScaleDenominator>

<sld:LineSymbolizer>
  <sld:Stroke>
    <sld:CssParameter name="stroke">
      <ogc:Literal>#C0C0C0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-linecap">
      <ogc:Literal>square</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-linejoin">
      <ogc:Literal>round</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-opacity">
      <ogc:Literal>1.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-width">
      <ogc:Literal>2.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-dashoffset">
```

```
                <ogc:Literal>0.0</ogc:Literal>
            </sld:CssParameter>
        </sld:Stroke>
    </sld:LineSymbolizer>
</sld:Rule>

<sld:Rule>
    <sld:Name>name</sld:Name>
    <sld:Title>title</sld:Title>
    <sld:Abstract>Abstract</sld:Abstract>

    <ogc:Filter>
        <ogc:And>
            <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>func_class</ogc:PropertyName>
                <ogc:Literal>5</ogc:Literal>
            </ogc:PropertyIsEqualTo>

            <ogc:Or>
                <ogc:PropertyIsEqualTo>
                    <ogc:PropertyName>ar_auto</ogc:PropertyName>
                    <ogc:Literal>Y</ogc:Literal>
                </ogc:PropertyIsEqualTo>
                <ogc:PropertyIsEqualTo>
                    <ogc:PropertyName>ar_bus</ogc:PropertyName>
                    <ogc:Literal>Y</ogc:Literal>
                </ogc:PropertyIsEqualTo>
                <ogc:PropertyIsEqualTo>
                    <ogc:PropertyName>ar_taxis</ogc:PropertyName>
                    <ogc:Literal>Y</ogc:Literal>
                </ogc:PropertyIsEqualTo>
                <ogc:PropertyIsEqualTo>
                    <ogc:PropertyName>ar_trucks</ogc:PropertyName>
                    <ogc:Literal>Y</ogc:Literal>
                </ogc:PropertyIsEqualTo>
                <ogc:PropertyIsEqualTo>
                    <ogc:PropertyName>ar_traff</ogc:PropertyName>
                    <ogc:Literal>Y</ogc:Literal>
                </ogc:PropertyIsEqualTo>
                <ogc:PropertyIsEqualTo>
                    <ogc:PropertyName>ar_deliv</ogc:PropertyName>
                    <ogc:Literal>Y</ogc:Literal>
                </ogc:PropertyIsEqualTo>
                <ogc:PropertyIsEqualTo>
                    <ogc:PropertyName>ar_emerveh</ogc:PropertyName>
                    <ogc:Literal>Y</ogc:Literal>
                </ogc:PropertyIsEqualTo>
            </ogc:Or>

            <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>paved</ogc:PropertyName>
                <ogc:Literal>Y</ogc:Literal>
            </ogc:PropertyIsEqualTo>

            <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>ferry_type</ogc:PropertyName>
                <ogc:Literal>H</ogc:Literal>
            </ogc:PropertyIsEqualTo>
        </ogc:And>
    </ogc:Filter>

    <sld:MinScaleDenominator>5000.0</sld:MinScaleDenominator>
    <sld:MaxScaleDenominator>10000.0</sld:MaxScaleDenominator>

    <sld:LineSymbolizer>
        <sld:Stroke>
            <sld:CssParameter name="stroke">
                <ogc:Literal>#C0C0C0</ogc:Literal>
            </sld:CssParameter>
            <sld:CssParameter name="stroke-linecap">
                <ogc:Literal>butt</ogc:Literal>
            </sld:CssParameter>
        </sld:Stroke>
    </sld:LineSymbolizer>
</sld:Rule>
```

```
</sld:CssParameter>
<sld:CssParameter name="stroke-linejoin">
  <ogc:Literal>round</ogc:Literal>
</sld:CssParameter>
<sld:CssParameter name="stroke-opacity">
  <ogc:Literal>1.0</ogc:Literal>
</sld:CssParameter>
<sld:CssParameter name="stroke-width">
  <ogc:Literal>8.0</ogc:Literal>
</sld:CssParameter>
<sld:CssParameter name="stroke-dashoffset">
  <ogc:Literal>0.0</ogc:Literal>
</sld:CssParameter>
</sld:Stroke>
</sld:LineSymbolizer>
<sld:LineSymbolizer>
  <sld:Stroke>
    <sld:CssParameter name="stroke">
      <ogc:Literal>#FFFFFF</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-linecap">
      <ogc:Literal>square</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-linejoin">
      <ogc:Literal>round</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-opacity">
      <ogc:Literal>1.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-width">
      <ogc:Literal>6.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-dashoffset">
      <ogc:Literal>0.0</ogc:Literal>
    </sld:CssParameter>
  </sld:Stroke>
</sld:LineSymbolizer>
<sld:TextSymbolizer>
  <sld:Label>
    <ogc:PropertyName>st_name</ogc:PropertyName>
  </sld:Label>
  <sld:Font>
    <sld:CssParameter name="font-family">
      <ogc:Literal>Arial</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="font-size">
      <ogc:Literal>10.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="font-style">
      <ogc:Literal>normal</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="font-weight">
      <ogc:Literal>normal</ogc:Literal>
    </sld:CssParameter>
  </sld:Font>
  <sld:LabelPlacement>
    <sld:LinePlacement>
      <sld:PerpendicularOffset>
        <ogc:Literal>0.0</ogc:Literal>
      </sld:PerpendicularOffset>
    </sld:LinePlacement>
  </sld:LabelPlacement>
  <sld:Fill>
    <sld:CssParameter name="fill">
      <ogc:Literal>#000000</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="fill-opacity">
```

```

        <ogc:Literal>1.0</ogc:Literal>
      </sld:CssParameter>
    </sld:Fill>
  </sld:TextSymbolizer>
</sld:Rule>

<sld:Rule>
  <sld:Name>name</sld:Name>
  <sld:Title>title</sld:Title>
  <sld:Abstract>Abstract</sld:Abstract>

  <ogc:Filter>
    <ogc:And>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>func_class</ogc:PropertyName>
        <ogc:Literal>5</ogc:Literal>
      </ogc:PropertyIsEqualTo>

      <ogc:Or>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_auto</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_bus</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_taxis</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_trucks</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_traff</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_deliv</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
        <ogc:PropertyIsEqualTo>
          <ogc:PropertyName>ar_emerveh</ogc:PropertyName>
          <ogc:Literal>Y</ogc:Literal>
        </ogc:PropertyIsEqualTo>
      </ogc:Or>

      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>paved</ogc:PropertyName>
        <ogc:Literal>Y</ogc:Literal>
      </ogc:PropertyIsEqualTo>

      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ferry_type</ogc:PropertyName>
        <ogc:Literal>H</ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:And>
  </ogc:Filter>

  <sld:MinScaleDenominator>1000.0</sld:MinScaleDenominator>
  <sld:MaxScaleDenominator>5000.0</sld:MaxScaleDenominator>

  <sld:LineSymbolizer>
    <sld:Stroke>
      <sld:CssParameter name="stroke">
        <ogc:Literal>#C0C0C0</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="stroke-linecap">
        <ogc:Literal>butt</ogc:Literal>
      </sld:CssParameter>
    </sld:Stroke>
  </sld:LineSymbolizer>

```

```
        </sld:CssParameter>
        <sld:CssParameter name="stroke-linejoin">
          <ogc:Literal>round</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-opacity">
          <ogc:Literal>1.0</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-width">
          <ogc:Literal>12.0</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-dashoffset">
          <ogc:Literal>0.0</ogc:Literal>
        </sld:CssParameter>
      </sld:Stroke>
    </sld:LineSymbolizer>

    <sld:LineSymbolizer>
      <sld:Stroke>
        <sld:CssParameter name="stroke">
          <ogc:Literal>#FFFFFF</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-linecap">
          <ogc:Literal>square</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-linejoin">
          <ogc:Literal>round</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-opacity">
          <ogc:Literal>1.0</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-width">
          <ogc:Literal>10.0</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="stroke-dashoffset">
          <ogc:Literal>0.0</ogc:Literal>
        </sld:CssParameter>
      </sld:Stroke>
    </sld:LineSymbolizer>

    <sld:TextSymbolizer>
      <sld:Label>
        <ogc:PropertyName>st_name</ogc:PropertyName>
      </sld:Label>

      <sld:Font>
        <sld:CssParameter name="font-family">
          <ogc:Literal>Arial</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="font-size">
          <ogc:Literal>12.0</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="font-style">
          <ogc:Literal>normal</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="font-weight">
          <ogc:Literal>normal</ogc:Literal>
        </sld:CssParameter>
      </sld:Font>

      <sld:LabelPlacement>
        <sld:LinePlacement>
          <sld:PerpendicularOffset>
            <ogc:Literal>0.0</ogc:Literal>
          </sld:PerpendicularOffset>
        </sld:LinePlacement>
      </sld:LabelPlacement>

      <sld:Fill>
        <sld:CssParameter name="fill">
          <ogc:Literal>#000000</ogc:Literal>
        </sld:CssParameter>
        <sld:CssParameter name="fill-opacity">
```

```
                <ogc:Literal>1.0</ogc:Literal>
            </sld:CssParameter>
        </sld:Fill>
    </sld:TextSymbolizer>
</sld:Rule>

<sld:Rule>
    <sld:Name>name</sld:Name>
    <sld:Title>title</sld:Title>
    <sld:Abstract>Abstract</sld:Abstract>

    <ogc:Filter>
        <ogc:And>
            <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>func_class</ogc:PropertyName>
                <ogc:Literal>5</ogc:Literal>
            </ogc:PropertyIsEqualTo>
            <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>ferry_type</ogc:PropertyName>
                <ogc:Literal>B</ogc:Literal>
            </ogc:PropertyIsEqualTo>
        </ogc:And>
    </ogc:Filter>

    <sld:MinScaleDenominator>10000.0</sld:MinScaleDenominator>
    <sld:MaxScaleDenominator>100000.0</sld:MaxScaleDenominator>

    <sld:LineSymbolizer>
        <sld:Stroke>
            <sld:CssParameter name="stroke">
                <ogc:Literal>#5B88FF</ogc:Literal>
            </sld:CssParameter>
            <sld:CssParameter name="stroke-linecap">
                <ogc:Literal>butt</ogc:Literal>
            </sld:CssParameter>
            <sld:CssParameter name="stroke-linejoin">
                <ogc:Literal>miter</ogc:Literal>
            </sld:CssParameter>
            <sld:CssParameter name="stroke-opacity">
                <ogc:Literal>1.0</ogc:Literal>
            </sld:CssParameter>
            <sld:CssParameter name="stroke-width">
                <ogc:Literal>2.0</ogc:Literal>
            </sld:CssParameter>
            <sld:CssParameter name="stroke-dashoffset">
                <ogc:Literal>0.0</ogc:Literal>
            </sld:CssParameter>
            <sld:CssParameter name="stroke-dasharray">
                1.0 3.0 5.0 3.0
            </sld:CssParameter>
        </sld:Stroke>
    </sld:LineSymbolizer>
</sld:Rule>

<sld:Rule>
    <sld:Name>name</sld:Name>
    <sld:Title>title</sld:Title>
    <sld:Abstract>Abstract</sld:Abstract>

    <ogc:Filter>
        <ogc:And>
            <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>func_class</ogc:PropertyName>
                <ogc:Literal>5</ogc:Literal>
            </ogc:PropertyIsEqualTo>
            <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>ferry_type</ogc:PropertyName>
                <ogc:Literal>B</ogc:Literal>
            </ogc:PropertyIsEqualTo>
        </ogc:And>
    </ogc:Filter>
</sld:Rule>
```

```
</ogc:Filter>

<sld:MinScaleDenominator>1000.0</sld:MinScaleDenominator>
<sld:MaxScaleDenominator>10000.0</sld:MaxScaleDenominator>

<sld:LineSymbolizer>
  <sld:Stroke>
    <sld:CssParameter name="stroke">
      <ogc:Literal>#5B88FF</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-linecap">
      <ogc:Literal>butt</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-linejoin">
      <ogc:Literal>miter</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-opacity">
      <ogc:Literal>1.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-width">
      <ogc:Literal>4.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-dashoffset">
      <ogc:Literal>0.0</ogc:Literal>
    </sld:CssParameter>
    <sld:CssParameter name="stroke-dasharray">
      1.0 3.0 5.0 3.0
    </sld:CssParameter>
  </sld:Stroke>
</sld:LineSymbolizer>
</sld:Rule>

<sld:Rule>
  <sld:Name>name</sld:Name>
  <sld:Title>title</sld:Title>
  <sld:Abstract>Abstract</sld:Abstract>

  <ogc:Filter>
    <ogc:And>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>func_class</ogc:PropertyName>
        <ogc:Literal>5</ogc:Literal>
      </ogc:PropertyIsEqualTo>
      <ogc:PropertyIsEqualTo>
        <ogc:PropertyName>ferry_type</ogc:PropertyName>
        <ogc:Literal>R</ogc:Literal>
      </ogc:PropertyIsEqualTo>
    </ogc:And>
  </ogc:Filter>

  <sld:MinScaleDenominator>10000.0</sld:MinScaleDenominator>
  <sld:MaxScaleDenominator>100000.0</sld:MaxScaleDenominator>

  <sld:LineSymbolizer>
    <sld:Stroke>
      <sld:CssParameter name="stroke">
        <ogc:Literal>#000000</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="stroke-linecap">
        <ogc:Literal>butt</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="stroke-linejoin">
        <ogc:Literal>miter</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="stroke-opacity">
        <ogc:Literal>1.0</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="stroke-width">
        <ogc:Literal>2.0</ogc:Literal>
      </sld:CssParameter>
      <sld:CssParameter name="stroke-dashoffset">
```



```
                <ogc:Literal>0.0</ogc:Literal>
            </sld:CssParameter>
            <sld:CssParameter name="stroke-dasharray">
                1.0 3.0 5.0 3.0
            </sld:CssParameter>
        </sld:Stroke>
    </sld:LineSymbolizer>
</sld:Rule>

<sld:Rule>
    <sld:Name>name</sld:Name>
    <sld:Title>title</sld:Title>
    <sld:Abstract>Abstract</sld:Abstract>

    <ogc:Filter>
        <ogc:And>
            <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>func_class</ogc:PropertyName>
                <ogc:Literal>5</ogc:Literal>
            </ogc:PropertyIsEqualTo>
            <ogc:PropertyIsEqualTo>
                <ogc:PropertyName>ferry_type</ogc:PropertyName>
                <ogc:Literal>R</ogc:Literal>
            </ogc:PropertyIsEqualTo>
        </ogc:And>
    </ogc:Filter>

    <sld:MinScaleDenominator>1000.0</sld:MinScaleDenominator>
    <sld:MaxScaleDenominator>10000.0</sld:MaxScaleDenominator>

    <sld:LineSymbolizer>
        <sld:Stroke>
            <sld:CssParameter name="stroke">
                <ogc:Literal>#000000</ogc:Literal>
            </sld:CssParameter>
            <sld:CssParameter name="stroke-linecap">
                <ogc:Literal>butt</ogc:Literal>
            </sld:CssParameter>
            <sld:CssParameter name="stroke-linejoin">
                <ogc:Literal>miter</ogc:Literal>
            </sld:CssParameter>
            <sld:CssParameter name="stroke-opacity">
                <ogc:Literal>1.0</ogc:Literal>
            </sld:CssParameter>
            <sld:CssParameter name="stroke-width">
                <ogc:Literal>4.0</ogc:Literal>
            </sld:CssParameter>
            <sld:CssParameter name="stroke-dashoffset">
                <ogc:Literal>0.0</ogc:Literal>
            </sld:CssParameter>
            <sld:CssParameter name="stroke-dasharray">
                1.0 3.0 5.0 3.0
            </sld:CssParameter>
        </sld:Stroke>
    </sld:LineSymbolizer>
</sld:Rule>

</sld:FeatureTypeStyle>
</sld:UserStyle>
```