

STRAN ZA POPRAVKE, ERRATA

Stran z napako

Vrstica z napako

Namesto

Naj bo

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:

Simon Jud

Izdelava GIS aplikacije za potrebe vzdrževanja cest z dostopom do odprto - kodne PostgreSQL baze

Diplomska naloga št.: 757

Mentor:

izr. prof. dr. Radoš Šumrada

Somentor:

doc. dr. Matevž Dolenc

Ljubljana, 23. 6. 2008

IZJAVA O AVTORSTVU

Podpisani **SIMON JUD** izjavljam, da sem avtor diplomske naloge z naslovom:
**»IZDELAVA GIS-APLIKACIJE ZA POTREBE VZDRŽEVANJA OSI DRŽAVNIH
CEST Z DOSTOPOM DO ODPRTOKODNE PODATKOVNE BAZE POSTGRESQL«.**

Izjavljam, da se odpovedujem vsem materialnim in avtorskim pravicam, ter jih prenašam na UL, Fakulteto za gradbeništvo in geodezijo.

Ljubljana, 22. 5. 2008

BIBLIOGRAFSKO-DOKUMENTACIJSKA STRAN IN IZVLEČEK

UDK:	004.6:625.7:659.2:91(043.2)
Avtor:	Simon Jud
Mentor:	izr. prof. dr. Radoš Šumrada, univ. dipl. inž. geod.
Somentor:	asist. dr. Matevž Dolenc, univ.dipl.inž.grad.
Naslov:	Izdelava GIS-aplikacije za potrebe vzdrževanja osi državnih cest z dostopom do odprtokodne podatkovne baze PostgreSQL
Obseg in oprema:	55 str., 26 sl., 6 pregl., 1 pril.
Ključne besede:	digitalna os ceste, GIS, aplikacija, PostgreSQL/PostGIS

Izvleček:

Osnovna evidenca podatkov o elementih cest se vodi v opisni podatkovni bazi, imenovani Banka cestnih podatkov. S sistemom odsekov in stacionaž je ta baza tudi univerzalni šifrant, ki omogoča zbiranje in vodenje podatkov o cestah. V zadnjih letih se izvaja postopen prehod na geografski informacijski sistem (v nadaljevanju GIS), podatkovni model pa se je ohranil. Nekateri opisni podatki so zato tesno povezani s položajem digitalne osi ceste. To narekuje popolnoma lastno logiko vzdrževanja baze geografskih cestnih podatkov. S komercialnimi GIS-orodji je urejanje tako organiziranih podatkov zelo zamudno, saj ne omogočajo avtomatskih postopkov dinamičnega določevanja opisnih podatkov. Zato smo izdelali popolnoma namensko GIS-aplikacijo, ki je namenjena izključno urejanju osi cest. Pri tem smo upoštevali celotno Metodologijo zajema in vzdrževanja osi državnih cest. Ta je bila izdelana v okviru raziskovalno-razvojnega projekta na Direkciji RS za ceste, ki je tudi upravljalec državnih cest.

BIBLIOGRAPHIC-DOCUMENTALISTIC INFORMATION

UDK:	004.6:625.7:659.2:91(043.2)
Author:	Simon Jud
Supervisor:	Assoc. Prof. Radoš Šumrada, Ph. D.
Co-Supervisor:	Assist. Matevž Dolenc, Ph. D.
Title:	Development of GIS-application for management of road axis by means of opensource PostgreSQL database
Notes:	55 p., 26 fig., 6 tab., 1 app.
Key words:	digital road axis, GIS, application, PostgreSQL/PostGIS

Abstract:

The basic database of road axis elements is managed in an attribute database, called Bank of road axis data. With its own system of road segments and stationariness it is also a universal encipher, which enables collection and management of road's data. Lately a transfer to a geographic information system (GIS) was performed, but the data model has remained the same. Some attributes are therefore tightly related with the position of road axis. It dictates very specific management logic of geographical road axis database. Usage of commercial GIS programmes takes up a good deal of time, because they don't support an automatic dynamic definition of attributes. Therefore, we developed a strictly devoted application for management of road axis data. For this, we took into consideration the entire Methodology of capturing and management of country roads. This document was initiated through a research development project by a Direction RS for roads, which is also a manager of country roads.

ZAHVALA

Za nasvete in strokovno pomoč se zahvaljujem mentorju izr. prof. dr. Radošu Šumradi, somentorju asist. dr. Matevžu Dolencu, mag. Tomažu Gvozdanoviču, Urošu Ranflu, univ. dipl. inž. geod. in ostalim sodelavcem na DFG CONSULTING, d.o.o.

Zahvaljujem se tudi Ivani in moji družini, ki so me v času študija in pisanja diplomske naloge ves čas spodbujali.

KAZALO VSEBINE

1	UVOD	1
2	DIGITALNA OS CESTE	3
2.1	Banka cestnih podatkov	3
2.2	Definicija poteka ceste in položaj osi	6
2.2.1	Položaj osi	7
2.2.1.1	Enostavne osi	7
2.2.1.2	Navidezne povezave	8
2.2.1.3	Kompleksnejši primeri	9
2.2.2	Določevanje odsekov cest	10
2.2.2.1	Določevanje odsekov na avtocestah in hitrih cestah	11
2.3	Osnovni grafični objekti	12
2.3.1	Geometrija	12
2.3.2	Topologija	14
2.4	Izpeljani objekti	16
2.5	Vektorizacija in stopnja generalizacije	17
2.5.1	Zamaknjena križišča	18
2.5.2	Kraki v križiščih	18
2.5.3	Krožna križišča	19
2.6	Podatkovni model opisnih podatkov	20
2.6.1	Identifikatorji	20
2.6.2	Podatki za hierarhično združevanje (agregacija)	21
2.6.3	Opisni podatki v zvezi z geometrijo	21
2.6.3.1	Dolžine	21
2.6.3.2	Stacionaže	22
3	IZDELAVA GIS-APLIKACIJE ZA VZDRŽEVANJE BAZE OSI DRŽAVNIH CEST	25
3.1	Sistemska analiza	26
3.2	Podatkovni model geografskih podatkov	28
3.2.1	Struktura ESRI shape formata	28
3.2.2	PostGIS – WKB format	30

3.2.2.1	Struktura WKB objektov	32
3.3	Konstrukcija in upravljanje s podatkovno bazo PostGIS	33
3.3.1	Administracija baze	33
3.3.2	Primarni uvoz podatkov v bazo	34
3.3.3	Nadzor nad dodeljevanjem unikatnih identifikatorjev	35
3.4	Razvoj in funkcionalnost aplikacije	36
3.4.1	Namestitev osnovnih kontrol pogleda, izbiranja in urejanja	37
3.4.2	Podatkovni sloji	38
3.4.3	Uvoz in shranjevanje podatkov	40
3.4.3.1	Pretvornik geometričnih podatkov (WKB <--> SHP)	40
3.4.3.2	Programski vmesnik libpq	41
3.4.4	Programske procedure pri urejanju osi ceste	42
3.4.4.1	Geometrične in topološke kontrole	43
3.4.4.2	Urejanje odsekov	44
3.4.4.3	Ustvarjanje izpeljanih objektov	45
3.4.4.4	Nekatere dodatne funkcije	46
3.5	Prikaz uporabe aplikacije	46
4	ZAKLJUČKI	49
	VIRI	51
	PRILOGE	52
	Priloga A: Tabela opisnih podatkov sloja digitalne osi državnih cest	52

KAZALO PREGLEDNIC

Preglednica 1: Kategorije in dolžine državnih cest	4
Preglednica 2: Geometrični tipi shape objektov	29
Preglednica 3: Osnovni geometrični tipi po OGC standardizaciji	31
Preglednica 4: Struktura tabele z zadnjimi uporabljenimi identifikatorji	35
Preglednica 5: Definicija slojev v aplikaciji	39
Preglednica 6: Osnovne funkcije programske knjižnice <i>libpq</i>	41

KAZALO SLIK

Slika 1: Evidence o cestah in državni organi	3
Slika 2: Makro in mikro nivo cestnega omrežja (DRSC, Metodologija ..., marec 2007)	6
Slika 3: Primer položaja osi ceste z deljenima smernima voziščema (DRSC, Metodologija ..., marec 2007)	7
Slika 4: Navidezni del priključka AC ali HC (DRSC, Metodologija ..., marec 2007)	8
Slika 5: Potek osi na počivališču (DRSC, Metodologija ..., marec 2007)	9
Slika 6: Primer poteka osi v dvoosem (levo) in krožnem (desno) križišču (DRSC, Metodologija ..., marec 2007)	10
Slika 7: Poimenovanje krakov v avtocestnem priključku (DRSC, Metodologija..., marec 2007)	11
Slika 8: Določevanje osnovnih grafičnih objektov (DRSC, Metodologija..., marec 2007)	12
Slika 9: Enodelna in večdelna polilinja (DRSC, Metodologija..., marec 2007)	13
Slika 10: Zapiranje polilnije (DRSC, Metodologija..., marec 2007)	14
Slika 11: Sekanje polilnije same s seboj (DRSC, Metodologija..., marec 2007)	15
Slika 12: Sekanje osnovnih grafičnih objektov (DRSC, Metodologija..., marec 2007)	15
Slika 13: Primeri visečih vozlišč (DRSC, Metodologija..., marec 2007)	16
Slika 14: Tlorisni (levo) in prečni prikaz (desno) tolerance pri digitalizaciji cestne osi (DRSC, Metodologija..., marec 2007)	17
Slika 15: Generazacija osi v zamaknjenih križiščih (DRSC, Metodologija..., marec 2007)	18
Slika 16: Generalizacija krakov v križiščih (DRSC, Metodologija..., marec 2007)	19
Slika 17: Generalizacija krožnih križišč (DRSC, Metodologija..., marec 2007)	19
Slika 18: Stacionažna tablica; cesta 1380, odsek 404, km 12.0	22
Slika 19: Odsek z dodano stacionažo (DRSC, Metodologija..., marec 2007)	24
Slika 20: Shematski prikaz zapisa prostorskih podatkov v ESRI shape formatu	28
Slika 21: Shematski prikaz PostGIS strukture prostorskih podatkov	31

Slika 22: Struktura WKB objektov	32
Slika 23: Izgled administratorskega grafičnega vmesnika za PostgreSQL/PostGIS	33
Slika 24: Diagram poteka urejanja sloja osi cest	43
Slika 25: Prikaz števila GID-objektov prvotnega odseka (levo) in izpeljanega odseka (desno)	45
Slika 26: Tipičen primer uporabe aplikacije (koraki 1-6)	48

1 UVOD

Informacijski sistemi temeljijo na računalniški tehnologiji, zato je njihov razvoj tesno povezan z razvojem računalništva. Zaradi naraščanja količine podatkov, za katere je bilo potrebno voditi natančne zbirke, je računalništvo doživljalo skokovit razvoj. Vzporedno oz. na osnovi tega so se razvijali informacijski sistemi na vseh področjih družbenega življenja. Celovite rešitve pri vzdrževanju prostorskih podatkov nam nudi tehnologija geografskih informacijskih sistemov (v nadaljevanju GIS). Omogoča nam zajem, organizacijo, vzdrževanje, vizualizacijo, kombiniranje podatkov, analizo in nenazadnje tudi odločanje. GIS v osnovi sestavljajo trije sklopi, ki se med seboj prepletajo in so odvisni drug od drugega:

- **podatki** (rastrski, vektorski, meta podatki),
- **uporabniki** (uporabniki in razvijalci programske opreme),
- **strojna in programska oprema** (osebni računalniki, strežniki, aplikacije).

Največjo oviro pri razvoju sistema GIS je v preteklosti predstavljala zmogljivost računalniške opreme. Danes s tem nimamo težav, saj so kapacitete računalnikov, strežnikov in druge strojne opreme zelo visoke. Zajem prostorskih podatkov prav tako poteka že povsem v digitalni obliki. Vedno znova pa se srečujemo z novimi nalogami, ki bi jih hoteli izvajati s podatki, s katerimi razpolagamo. Ljudje smo nagnjeni k iskanju vedno boljših rešitev in orodij za avtomatizacijo postopkov.

Podatke o državnih cestah vodi in vzdržuje Direkcija Republike Slovenije za ceste (DRSC). Šele v zadnjih letih se izvaja postopen prehod iz opisne podatkovne baze, imenovane Banka cestnih podatkov (v nadaljevanju BCP), na sistem GIS. Ugotovljeno je bilo, da tudi največji komercialni GIS-sistemi, ne omogočajo enostavnega vzdrževanja baze cestnih osi. Razlog se skriva v precej nestandardnemu podatkovnemu modelu, ki ohranja podatkovno strukturo, kot jo definira baza BCP. Opisni podatki so dinamično odvisni od položaja osi. Zato je DRSC izdala javno naročilo za izdelavo aplikacije za vzdrževanje osi državnih cest. Projekt, pri katerem sem sodeloval tudi sam, je pridobilo podjetje DFG CONSULTING, d.o.o. V okviru tega projekta je nastala tudi ta diplomska naloga.

Vsebina diplomske naloge je razdeljena na dva večja sklopa. Prvi del problematizira način vzdrževanja osi, drugi pa podaja praktične programske rešitve. V prvem delu so predstavljena pravila in značilnosti, ki veljajo za linijski sloj digitalne osi ceste. Poznavanje teh pravil je nujno za izdelavo dobre aplikacije, ki bo sposobna usklajenega vzdrževanja podatkov o cestah. Os ceste je obravnavana kot mrežni linijski model s svojimi geometričnimi, topološkimi in opisnimi posebnostmi. Opisana so pravila za določevanje položaja osi glede na cestišče. Predstavljen je podatkovni model ter podrobnejša predstavitev opisnih podatkov.

V drugem delu je predstavljena namizna aplikacija za vzdrževanje sloja osi državnih cest, ki je izdelana v razvojnem okolju Microsoft Visual C++ 6.0. Gre za t.i. vertikalno aplikacijo, kar pomeni, da ni namenjena za splošno uporabo, pač pa je njena funkcionalnost izdelana izrecno za potrebe urejanja in vzdrževanja podatkov o cestah. Predstavljen je podatkovni model in organizacija geografskih podatkov. Prikazane so nekatere pomembnejše programske procedure, ki zagotavljajo učinkovito urejanje podatkov o cestah. Podan je uporabniški primer, ob katerem so razložene posamezne faze postopka urejanja baze državnih cest.

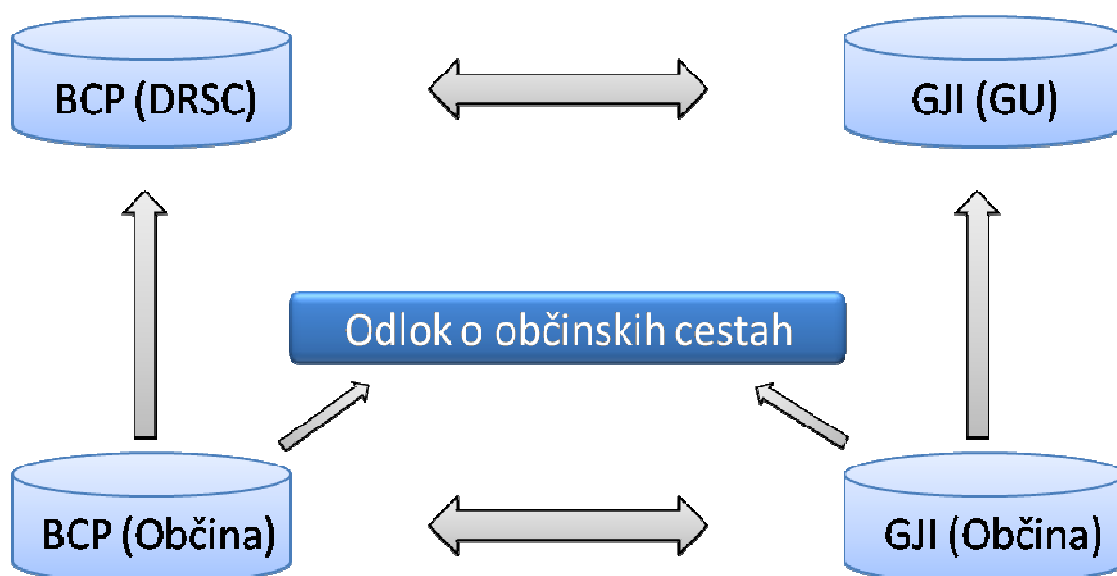
V diplomskem delu ne obravnavam posebej GIS-teorije, saj je bilo na to temo napisanih že kar nekaj diplomskih nalog. Kljub temu pa skušam skozi drugo poglavje predstaviti nekatere glavne zakonitosti tehnologije GIS. Strokovno podkovani bralec ne bo imel težav z razumevanjem vsebine. Morda bo odkril tudi kakšno nepravilnost. Za laičnega bralca pa je priporočljivo osnovno poznavanje GIS-tehnologije.

2 DIGITALNA OS CESTE

To poglavje je povzeto po raziskovalno razvojnem projektu Direkcije RS za ceste z naslovom Metodologija vzdrževanja sloja osi državnih cest s postavitvijo podatkovnega modela. Navedena dejstva temeljijo na zakonodaji in pravilnikih, ki urejajo področje cest. Nekateri podatki so privzeti iz vsebine spletne strani Direkcije RS za ceste (www.dc.gov.si).

2.1 Banka cestnih podatkov

Javne ceste so državne in občinske ceste. Na podlagi Zakona o javnih cestah /ZJC/ (Ur. l. RS št. 33/2006) in Pravilnika o načinu označevanja javnih cest in o evidencah o javnih cestah in objektih na njih (Ur. l. RS št. 47/1997 in 2/2004), je DRSC dolžna voditi evidenco o javnih cestah oz. Banko cestnih podatkov (BCP). Evidenco o državnih cestah vodi in vzdržuje DRSC. Evidenco o občinskih cestah in objektih na njih pa vodi občinska uprava, ki posreduje podatke DRSC. Na podlagi Zakona o urejanju prostora /ZUreP/ (Ur. l. RS št. 110/2002) je DRSC dolžna voditi tudi evidenco o državnih cestah kot delu zbirnega katastra gospodarske javne infrastrukture (v nadaljevanju GJI), s katerim upravlja Geodetska uprava RS. Slika 1 prikazuje odnose med podatkovnima bazama BCP in GJI ter državnimi organi, ki jih vodijo.



Slika 1: Evidence o cestah in državni organi

Širši problem predstavlja usklajenost med temi evidencami, kjer prihaja do topoloških neujemanj med podatki. Ta tematika je obravnavana v diplomskem delu Vladimirja Zormana, z naslovom Postopek vzpostavitve topološko urejenega podatkovnega sloja cest za potrebe zbirnega katastra gospodarske javne infrastrukture (2007, Diplomaska naloga). Težišče te diplomske naloge pa je predvsem v iskanju tehničnih rešitev za obvladovanje evidence državnih cest.

Po podatkih, ki so bili objavljeni decembra 2006, je skupna dolžina državnih cest 6425 km. Od tega so avtoceste in večji del hitrih cest v upravljanju Družbe za avtoceste v Republiki Sloveniji (DARS), z ostalimi cestami pa upravlja DRSC. Preglednica 1 prikazuje kategorizacijo državnih cest in njihove skupne dolžine (www.dc.gov.si).

Preglednica 1: Kategorije in dolžine državnih cest

oznaka	kategorija	dolžina v km
AC	avtoceste	505
HC	hitre ceste	102
G1	glavne ceste I	483
G2	glavne ceste II	447
R1	regionalne ceste I	954
R2	regionalne ceste II	1225
R3	regionalne ceste III	2114
RT	regionalne turst. ceste	595
	skupaj:	6425

Zbiranje podatkov o državnih cestah se izvaja že od leta 1954, v digitalni obliki kot Banka cestnih podatkov pa od leta 1974. Ta evidenca je bila zasnovana kot računalniško vodena zbirka opisnih podatkov, ki služijo kot informacija pri upravljanju s cestami. S svojim sistemom odsekov in stacionaž je BCP tudi univerzalni šifrant, ki omogoča zbiranje podatkov

o cestnem omrežju. BCP se vodi v skladu z prej omenjenim Zakonom o javnih cestah ter Pravilnikom o evidencah o javnih cestah in objektih na njih, ki natančno predpisuje vsebino, način zbiranja in razpolaganje s podatki. Podatki se zbirajo na podlagi poročil o izvedenih delih in meritvah cest. V okviru vodenja evidenc o cestah se zbirajo podatki o (www.dc.gov.si):

- poteku cest v prostoru in njihovem vzdrževanju,
- prečnih in vzdolžnih elementih cest,
- objektih na cestah,
- prometu,
- prometni signalizaciji in opremi,
- investicijsko tehnični dokumentaciji,
- stanju vozišč,
- stanju premostitvenih objektov,
- prometnih nesrečah (zbira in vodi Policija).

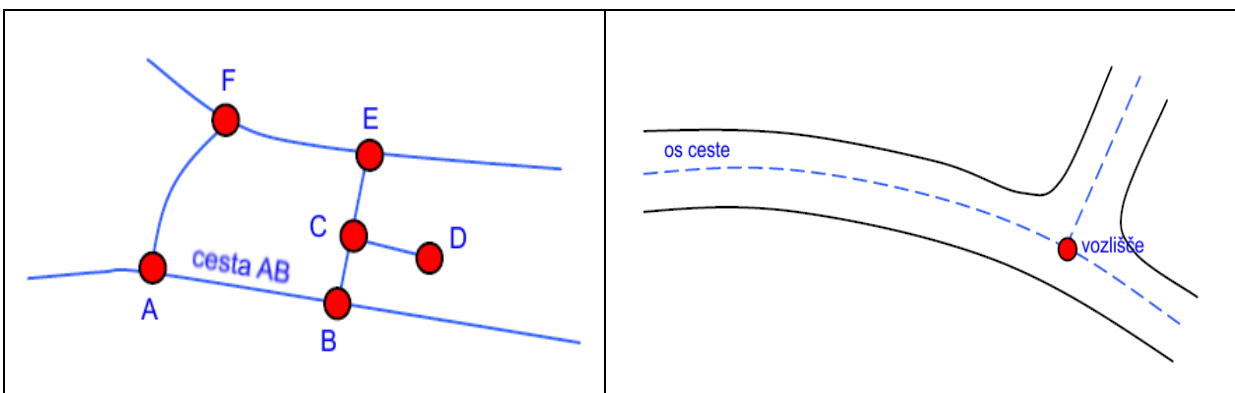
S prenosom evidence o cestah v sistem GIS se pojavi pojem **digitalne osi ceste**. Ta predstavlja osnovo za vodenje georeferencirane baze cestnih podatkov. Os ceste, kot grafični objekt, je nosilec tako opisnih kot tudi geometričnih podatkov o poteku ceste v prostoru. Za vzpostavitev in vzdrževanje GIS-baze linijskega sloja osi cest, pa moramo zadovoljiti tako geometričnim in topološkim pravilom za GIS, kot tudi pravilom, ki jih narekujejo ceste s svojim podatkovnim modelom.

Digitalna os za državne in lokalne ceste je bila vzpostavljena v letu 1995 na osnovi kartografskega gradiva. V letu 2003 pa je bila vzpostavljena os ceste na osnovi kinematične GPS-izmere. Zaradi zahtev po vključitvi dodatnih podatkov v os ceste, predvsem za potrebe transportnih in navigacijskih sistemov in vzpostavitve katastra GJI, je bilo potrebno ponovno določiti celovito metodologijo zajema in vzdrževanja baze osi cest. Le-ta je navedena v raziskovalno-razvojnem projektu z naslovom Metodologija vzdrževanja sloja osi državnih cest ... (DRSC, marec 2007).

2.2 Definicija poteka ceste in položaj osi

Cesta je načrtno speljana pot ali linijska poveza med dvema točkama na zemeljski površini, ki je namenjena prometu z vozili. Več povezanih cest, ki se stikajo v vozliščih, tvori cestno omrežje. Potek osi ceste lahko definiramo na makro in mikro nivoju (DRSC, Metodologija ..., marec 2007):

- **makro nivo**; cestno omrežje, potek ceste med kraji,
- **mikro nivo**; potek končne digitalne osi glede na cestišče (položaj osi).



Slika 2: Makro in mikro nivo cestnega omrežja (DRSC, Metodologija ..., marec 2007)

Osnovni podatek o cesti je njen potek v naravi. Definiran je splošno, brez koordinat, z imeni krajev, ki so dovolj poznani. Ta podatek lahko pridobimo iz naslednjih virov (DRSC, Metodologija ..., marec 2007):

- uradni list, kjer je cesta uradno opredeljena z začetno in končno točko (mesto, naselje, priključitev na drugo cesto), lahko pa tudi z vmesnimi točkami (kraji skozi katere poteka),
- pregledne karte (1:25 000 ali 1:50 000), kjer so vrisane omenjene ceste.

Potek ceste, podan z začetno in končno točko, določa njeno smer, s tem pa tudi smer naraščanja stacionaže. Vprašanje pa je, kje določiti os ceste glede na cestišče.

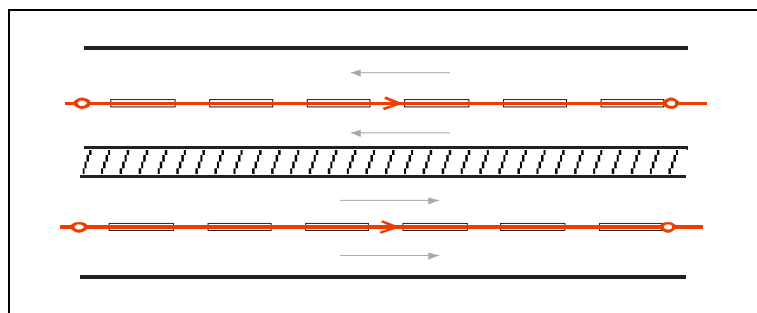
2.2.1 Položaj osi

Pod izrazom "položaj osi" obravnavamo lego digitalne vektorske osi kot generaliziranega prikaza ceste glede na vozišče. Pozicioniranje osi je odvisno od vrste ceste (širina, število voznih pasov, ločena vozišča, posebni primeri). Glede na kompleksnost določevanja položaja osi, lahko govorimo o enostavnih oseh, navideznih povezavah in križiščih (DRSC, Metodologija..., marec 2007).

2.2.1.1 Enostavne osi

O enostavni osi govorimo, kadar pri grafičnih elementih osi ne prihaja do stikov večih grafičnih objektov. V takih primerih je enostavno zagotoviti topološko pravilnost. Primeri enostavnih osi cest so naslednji:

- **ceste brez smernih vozišč;** os poteka po sredini vozišča v smeri stacionaže, ki je definirana s potekom odseka,
- **ceste z dvema smernima voziščema;** os poteka po sredinski črti v smeri stacionaže,
- **ceste z dvema smernima voziščema in več prometnimi pasovi;** os ceste poteka po sredinski dvojni črti, ki ločuje smerni vozišči,
- **ceste z deljenima smernima voziščema;** os poteka na obeh voziščih po sredini posameznega vozišča v smeri stacionaže (osi potekajo vzporedno v smeri stacionaže).



Slika 3: Primer položaja osi ceste z deljenima smernima voziščema (DRSC, Metodologija ..., marec 2007)

Odsek oz. del odseka obravnavamo kot cesto z dvema ločenima smernima voziščema, če je izpolnjen naslednji pogoj: smerni vozišči sta ločena z arhitektonsko oviro (robnik, zelenica, odbojna ograja, permanetna ograja z verižico), ki je daljša kot 200 m (DRSC, Metodologija..., marec 2007).

2.2.1.2 Navidezne povezave

V primerih, ko se cestni odseki priključujejo na ceste nižjega ali višjega reda, lahko prihaja do nesorazmernega podaljšanja osi glede na dejansko stanje. V primeru priključka avtoceste, leta poteka po delu vozišča, ki bi v drugačnem primeru predstavljal odstavni pas. Na tem delu bi se os priključka dejansko podvajala z osjo avtoceste. Tako ne bi dobil pravilne dolžine odseka. Do iste težave prihaja tudi na stikih med priključki in na razcepkih.

Zaradi omenjenih problemov se uvede t.i. navidezni del odseka oz. navidezna povezava. Navidezna povezava je del odseka z lastno geometrijo in enakimi opisnimi podatki, kot jih ima celoten odsek, s to razliko, da je začetna stacionaža enaka končni, njegova dolžina pa ne prispeva k skupni dolžini odseka (več o stacionaži v nadaljevanju). Na ta način se izognemo neupravičenemu podaljšanju skupne dolžine odseka, ko seštejemo vse dolžine njegovih elementov. Slika 4 ponazarja določitev položaja navideznega dela odseka.



Slika 4: Navidezni del priključka AC ali HC (DRSC, Metodologija ..., marec 2007)

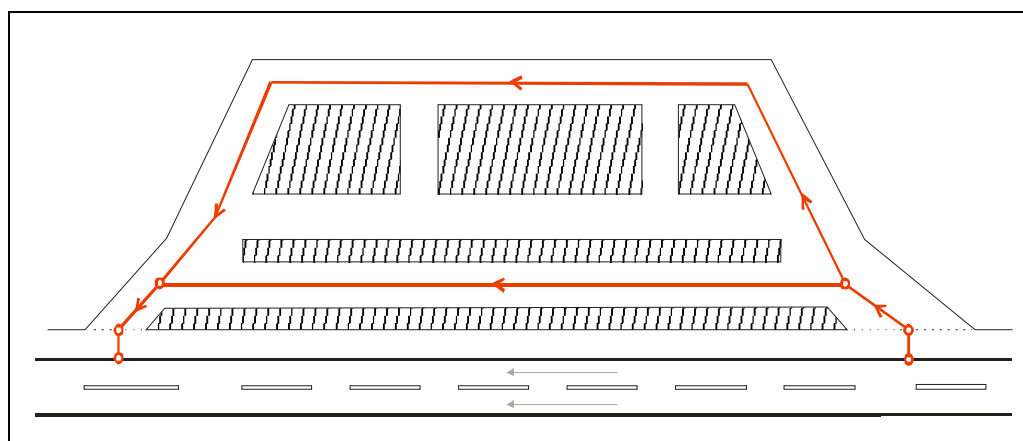
Os priključka grafično določimo tako, da ne posega v telo avtoceste ali hitre ceste (do zunanjenja roba vozišča). Zaradi zagotovitve povezave priključka z osjo avtoceste ga podaljšamo z navidezno povezavo pravokotno na os navezovalne osi. Navidezna povezava grafično pripada priključku, vendar dolžina elementa navidezne povezave ne prispeva k skupni dolžini celotnega odseka, začetna stacionaža pa je enaka končni (DRSC, Metodologija..., marec 2007).

2.2.1.3 Kompleksnejši primeri

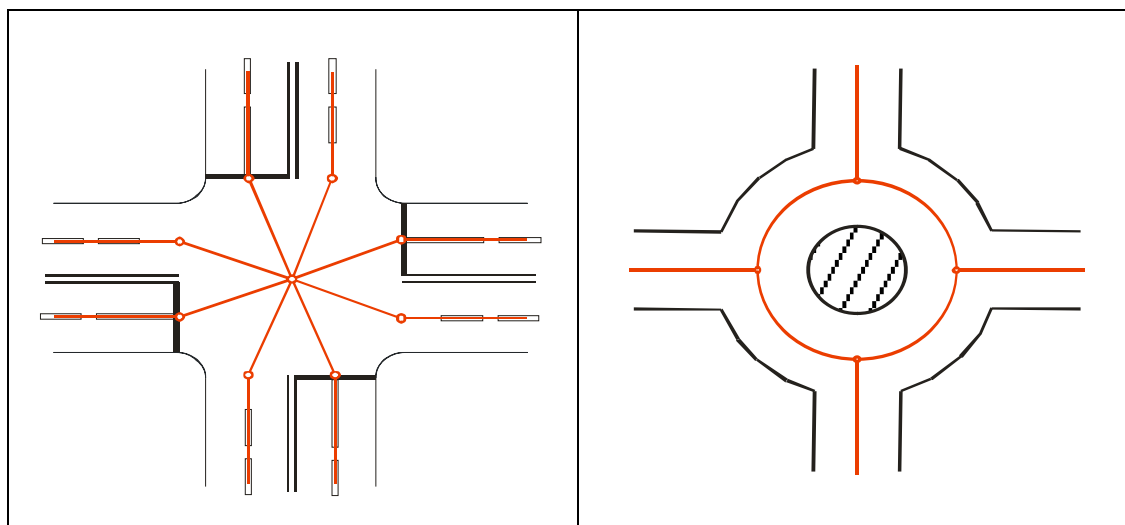
O kompleksnejših primerih govorimo kadar pride do spajanja več grafičnih objektov osi v eni ali večih točkah. To se zgodi v križiščih. Glede na način določitve osi obravnavamo naslednje vrste križišč:

- **enostavna križišča** (stik dveh ali več enosnih cest, osi cest se stikajo v eni sami skupni točki v sredini - težišču križišča),
- **križišča z dvoosnimi cestami** (možnosti: enosna – dvoosna, dvoosna – dvoosna cesta, osi cest se stikajo v težišču križišča, os dvoosne ceste prelomimo na stop črti in jo »zravnamo« nazaj v višini stop črte na drugi strani križišča),
- **krožna in Y križišča** (os ceste postavimo na sredino cestišča),
- **počivališča** (os ceste zajamemo po notranjem in zunanjem obodu počivališča).

Sliki 5 in 6 ponazarjata določitev položaja osi v nekaterih kompleksnejših primerih.



Slika 5: Potek osi na počivališču (DRSC, Metodologija ..., marec 2007)



Slika 6: Primer poteka osi v dvoosem (levo) in krožnem (desno) križišču (DRSC, Metodologija ..., marec 2007)

2.2.2 Določevanje odsekov cest

Preden se lotimo pravil za določevanje položaja osi ceste, se moramo najprej spoznati z pojmom odseka ceste. Pravila za določevanje odsekov podaja Pravilnik o evidencah o javnih cestah in objektih na njih. Navajam del 3. člena tega pravilnika:

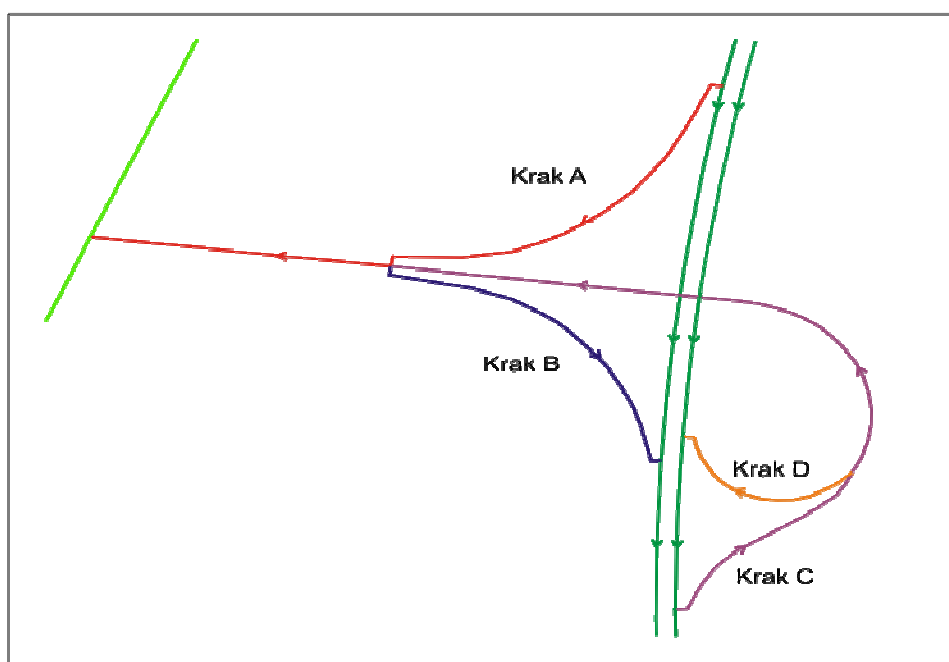
- javne ceste se lahko razdelijo na več odsekov, ki so medsebojno razmejeni s križiščem dveh ali več kategoriziranih cest ali izjemoma z drugo značilno mejno točko odseka (npr. državna meja, prelaz, železniška postaja, križišče z nekategorizirano cesto, ipd.),
- Odsek javne ceste ne sme biti krajši od 100 m, razen če je celotna javna cesta krajša od te dolžine.
- Odsek državne ceste se označi s štirimestno številko.

Cesta je torej sestavljena iz večih odsekov. Vsak odsek ceste nosi podatek o svoji dolžini (stacionaži). Začne se s stacionažo 0.000 km v razmejitveni točki in narašča v smeri poteka ceste. Vsota stacionaž odsekov ceste mora biti enaka dolžini celotne ceste. Odsek ceste pa je sestavljen iz večih osnovnih grafičnih objektov (DRSC, Metodologija..., marec 2007).

2.2.2.1 Določevanje odsekov na avtocestah in hitrih cestah

Odsek na avtocesti ali hitri cesti je del ceste, katerega začetek in konec praviloma razmejujeta izvennivojska priključka ali razcep na cesto enake ali nižje kategorije, izjemoma pa tudi katera druga značilna mejna točka odseka (npr. državna meja). Na avtocestah in hitrih cestah obravnavamo dva tipa odsekov: odsek avtoceste in priključek. Na sliki 7 sta prikazana oba tipa odsekov (odsek avtoceste v zeleni barvi in sistem priključkov v različnih barvah). Pri določevanju teh odsekov veljajo naslednja pravila (DRSC, Metodologija..., marec 2007):

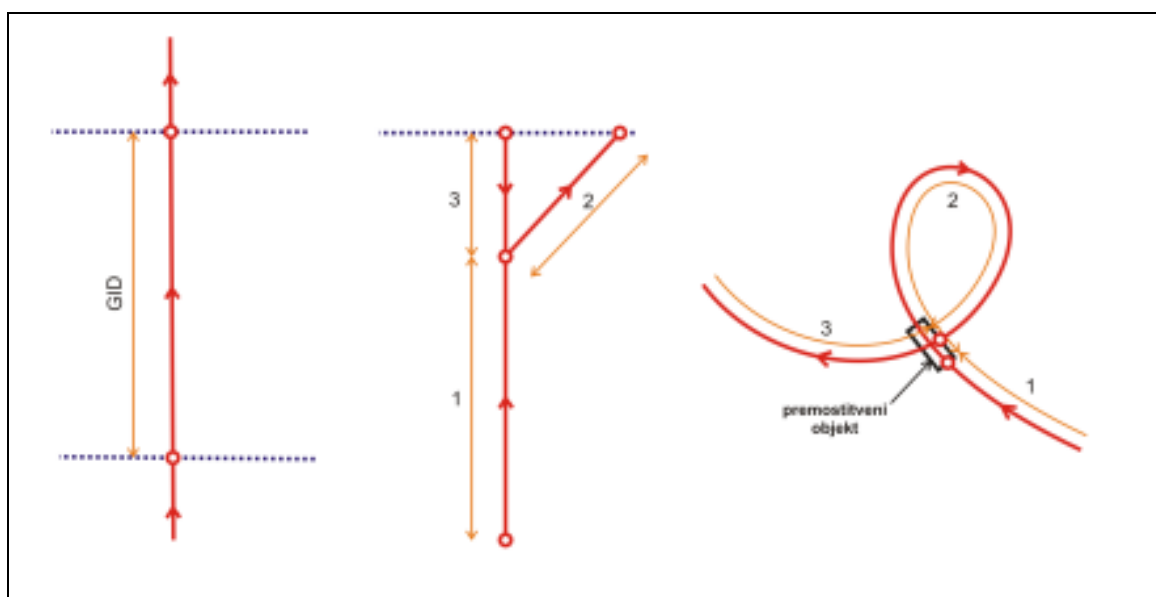
- začetek avtocestnega odseka predstavlja točka na začetku objekta (podvoz, nadvoz) v smeri stacionaže,
- sistemi priključkov oz. kraki (dovozi in izvozi) na posamezni lokaciji so oštevilčeni z isto številko odseka.



Slika 7: Poimenovanje krakov v avtocestnem priključku (DRSC, Metodologija..., marec 2007)

2.3 Osnovni grafični objekti

Osnovni grafični objekt oz. grafični identifikator (v nadaljevanju GID) je najmanjši del osi, ki je prikazan z enim grafičnim objektom (linijo ali polilinijo). V splošnem so GID-objekti na državnih cestah prekinjeni tam, kjer se priključuje kakšne druga cesta (državna, občinska, ...). Prav tako se GID prekine tudi na premostitvenih objektih ali drugih značilnih točkah (most, nadvoz, podvoz, prelaz, ...). Na sliki 8 je prikazano določevanje GID-objektov v nekaterih situacijah.

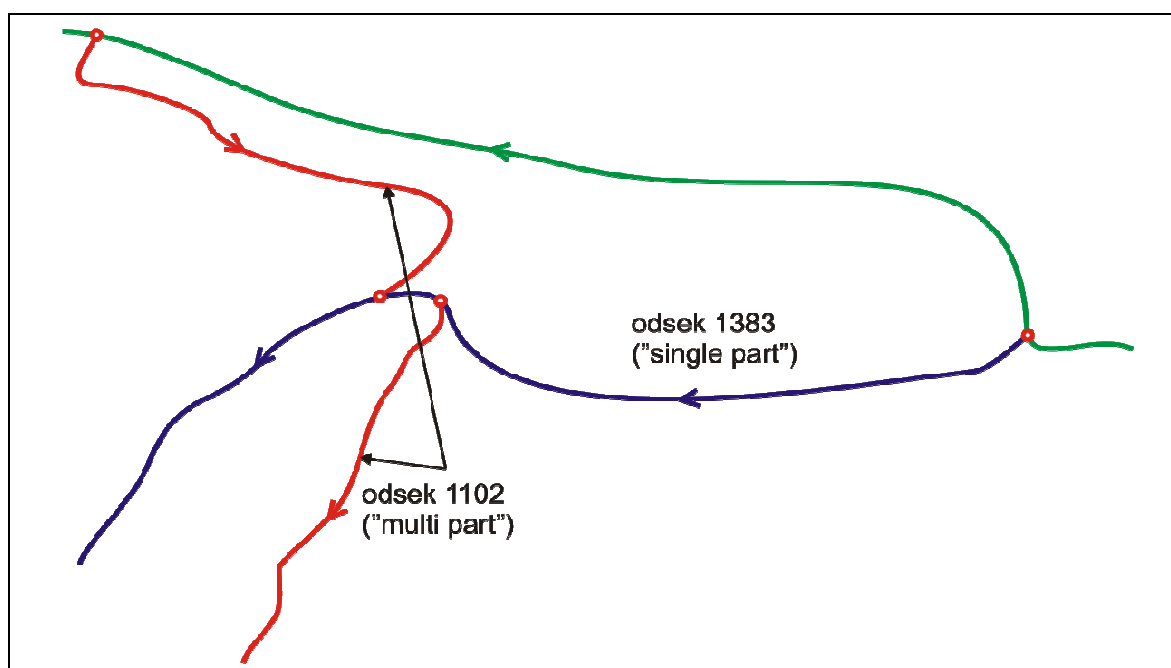


Slika 8: Določevanje osnovnih grafičnih objektov (DRSC, Metodologija..., marec 2007)

2.3.1 Geometrija

Geometrijo uporabljamo kot sredstvo za kvantitativni opis (položaj, velikost, obliko, orientacijo) objektov v prostoru s pomočjo koordinat in matematičnih funkcij. Matematične funkcije, ki jih uporabljamo za opis geometrije objekta, so odvisne od tipa referenčnega koordinatnega sistema, ki ga uporabljamo za definicijo položaja objekta. Geometrija je edina GIS-informacija, ki se spremeni, kadar želimo podatke transformirati v drugi koordinatni sistem. Položaj objekta opišemo s koordinatami. Geometrične lastnosti osnovnega grafičnega objekta osi ceste so naslednje (DRSC, Metodologija..., marec 2007):

- topološka oblika osi je polilinija, kar pomeni zaporedje vsaj dveh neistoležnih točk,
- podatki o točkah so 3D (XYZ) v državnem koordinatnem sistemu (Gauss-Kruegerjev k.s.),
- koordinate so zaokrožene na končno položajno natančnost 0.001 m (1mm),
- polilinija mora biti iz enega dela. Več-delna polilinija dovoljuje več neskljenjenih delov, ki nastopajo kot en grafični objekt s skupnimi opisnimi podatki. Razlika med eno-delno in več-delno polilinijo je ponazorjena na sliki 9.



Slika 9: Enodelna in večdelna polilinija (DRSC, Metodologija..., marec 2007)

Pri geometrični kontroli objektov je potrebno izvesti naslednje kontrole (DRSC, Metodologija..., marec 2007):

- polilinija mora imeti vsaj dve točki,
- zaporedne točke polilinije nesmejo biti istoležne,
- polilinija ne sme imeti »ostrih« zavojev (kotna toleranca),
- polilinija se ne sme »preveč« približati sama sebi (dolžinska toleranca).

2.3.2 Topologija

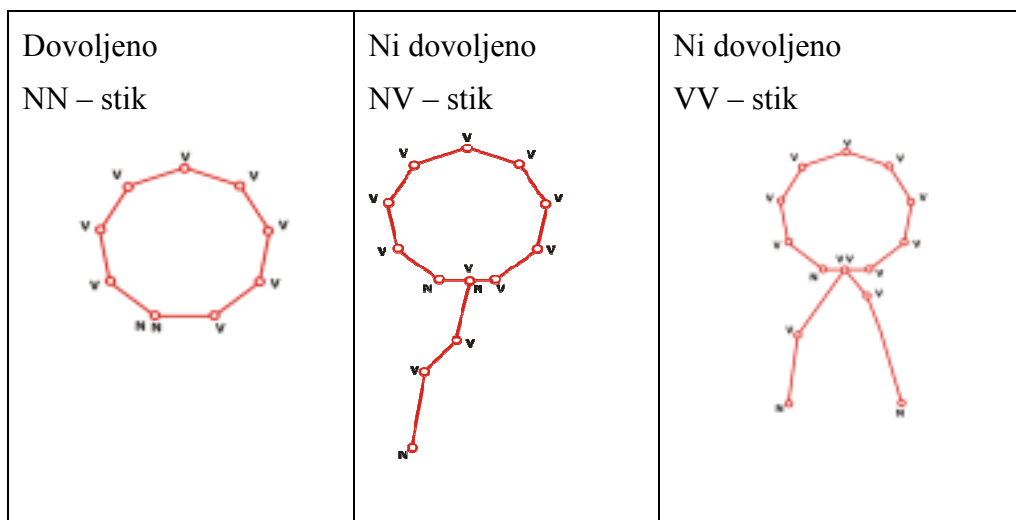
Topologija podaja logične odnose med objekti v prostoru, ki temeljijo na njihovem relativnem položaju. Ti odnosi se pri transformacijah med različnimi koordinatnimi sistemi ne spremenijo. Za definiranje topologije linijskih objektov moramo najprej opredeliti pojma vozlišče in lomna točka (DRSC, Metodologija..., marec 2007):

- **vozlišče** (označujemo z N) je začetna ali končna točka polilinije (ker smo opredelili polilinijo kot eno-delno, ima polilinija natanko dve vozlišči),
- **lomna točka** (označujemo z V) je vsaka točka polilinije, ki ni vozlišče (polilinija, ki ima le dve točki, ima dve vozlišči in nič lomnih točk).

Pri linijskih objektih prihaja do nekaterih topološko dvoumnih situacij, kjer so potrebne dodatne obravnave: **stikanje točk polilinije** (istoležnost nesosednjih točk), **sekanje polilinije same s seboj**, **sekanje med dvema linijskima objektoma**, **viseča vozlišča** (DRSC, Metodologija..., marec 2007).

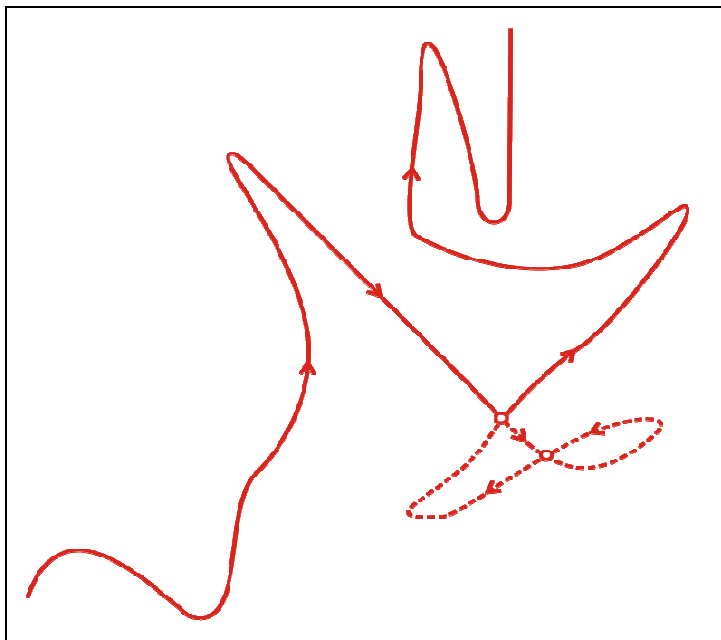
Pri stikih lahko obravnavamo naslednje situacije:

- stik NN (»O zapiranje«) je dovoljen (krožišča, obračališča, ...),
- stik NV (»P zapiranje«) ni dovoljen; objekt je potrebno razdeliti na dva,
- stik VV (»R zapiranje«) ni dovoljen.



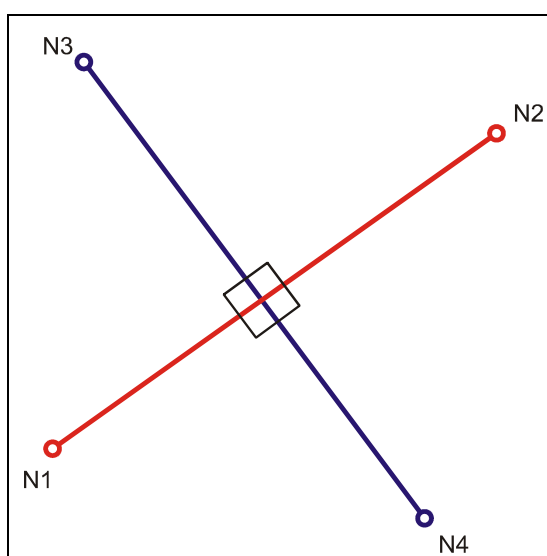
Slika 10: Zapiranje polilinije (DRSC, Metodologija..., marec 2007)

Sekanje polilinije same s seboj ni dovoljeno. Pojavi se lahko v dveh primerih: polilinija gre sama sebi v nadvoz (potrebno jo je razdeliti na več delov), ali pa gre za topološko napako (nepravilno zaporedje točk) (slika 11).



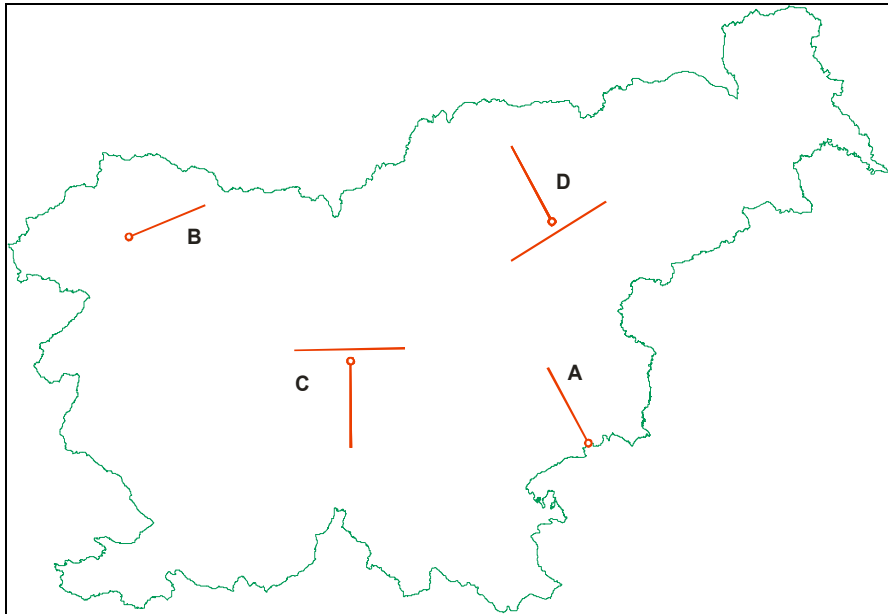
Slika 11: Sekanje polilinije same s seboj (DRSC, Metodologija..., marec 2007)

Sekanje med dvema linijskima objektoma je dovoljeno, če gre za nadvoz ali podvoz. Če gre za istonivojsko sekanje, je v križišče potrebno vstaviti vozlišče (slika 12).



Slika 12: Sekanje osnovnih grafičnih objektov (DRSC, Metodologija..., marec 2007)

Viseča vozlišča predstavljajo topološki primer, ko se objekt slepo konča. Taki primeri so lahko dovoljeni (slepe doline, državna meja, dejanski zaključek ceste, ...), ali pa so napake zajema osi cest. Na sliki 13 so predstavljeni naslednji primeri: A - zaključek ceste na meji (dovoljeno), B - slepa cesta npr. vrata (dovoljeno), C - slepa cesta kljub neposredni bližini druge ceste (dovoljeno), D - viseče vozlišče kot posledica napake (ni dovoljeno)



Slika 13: Primeri visečih vozlišč (DRSC, Metodologija..., marec 2007)

2.4 Izpeljani objekti

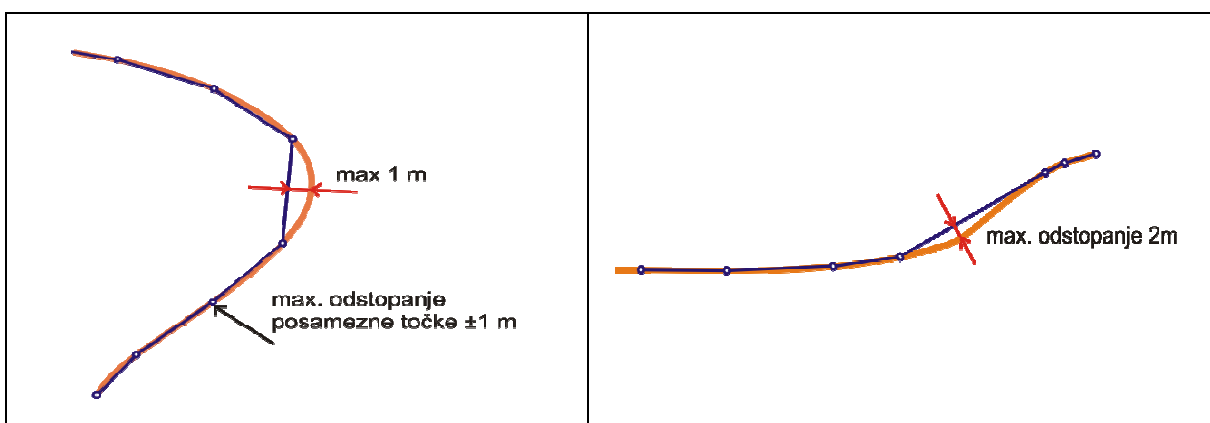
Osnovni grafični objekti na sloju državnih cest so prilagojeni topološko usklajenemu delu z celotnim cestnim omrežjem Republike Slovenije (vključuje tudi občinske ceste, gozdarske ceste, ...). Prekinjeni so na vseh križiščih, kjer prihaja do priključitve ostalih cest. Za potrebe izvajanja raznih analiz samo na sloju državnih cest je takšna razdelitev osi nepotrebna in zelo neprimerna. Zato je potrebno ustvariti sloj izpeljanih objektov na nivoju odsekov državnih cest. Izpeljani objekti so torej prekinjeni med posameznimi odseki in na križiščih med državnimi cestami. Prevzame se celotna geometrija osnovnih grafičnih objektov (vmesna vozlišča postanejo lomne točke), opisni podatki pa se ustrezno prenesejo in preračunajo. Pri tem se ohrani izvorna topologija objektov (DRSC, Metodologija..., marec 2007).

2.5 Vektorizacija in stopnja generalizacije

Pravila za določevanje položaja osi ceste so pravzaprav opisana v prejšnjih poglavjih. Nismo pa še nič povedali o natančnosti določitve položaja točk pri digitalizaciji osi. Osnovno pravilo je, da mora gostota točk posamezne osi slediti ukrivljenosti odseka ceste. Gostota točk je pogojena (DRSC, Metodologija..., marec 2007):

- **s planarno ukrivljenostjo**, tako da na posameznem delu polilinije med dvema zaporednima točkama maksimalna pravokotna razdalja ne odstopa od idealne linije za več kot 1 m,
- **z vertikalno ukrivljenostjo**, tako da na posameznem delu polilinije med dvema zaporednima točkama maksimalna višina, izmerjena pravokoto od idealne linije, ne odstopa več kot 2 m.

Slika 14 prikazuje dovoljene tolerance v horizontalni in vertikalni smeri pri digitalizaciji osi.

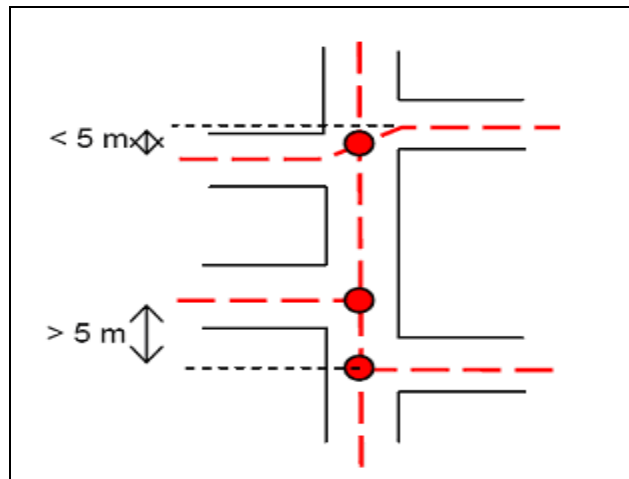


Slika 14: Tlorisni (levo) in prečni prikaz (desno) tolerance pri digitalizaciji cestne osi (DRSC, Metodologija..., marec 2007)

Pravila generalizacije uporabljamo pri poenostavljenem prikazu cestnega omrežja. Za osnovno predstavitev pa generalizacije ne uporabljamo, ker želimo os ceste predstaviti z dejansko geometrijo. Kljub temu na mikro nivoju pride do dvoumnih situacij (v križiščih in posebnih primerih), kjer je potrebno določiti pravila zajema in generalizacije osi cest.

2.5.1 Zamaknjena križišča

Če je razdalja med osema na glavni osi manjša kot 5 m, se tako križišče predstavi z enim vozliščem. Če pa je razdalja med osema večja od 5 m, je potrebno v križišču podati dve vozlišči (DRSC, Metodologija..., marec 2007). Slika 15 prikazuje opisana primeri.



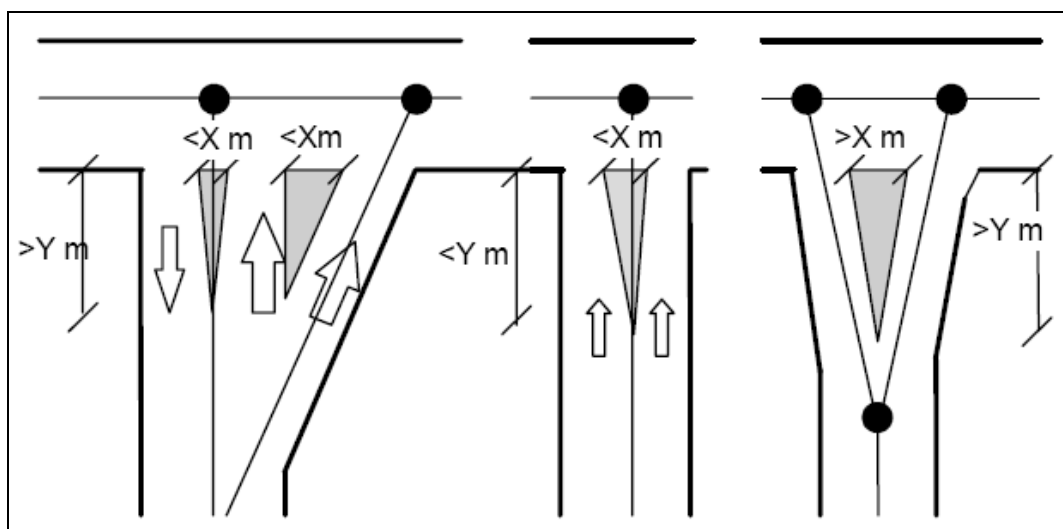
Slika 15: Generalizacija osi v zamaknjenih križiščih (DRSC, Metodologija..., marec 2007)

2.5.2 Kraki v križiščih

Kraki v križiščih so lahko vzpostavljeni fizično ali pa simbolno. Krak je fizično vzpostavljen, kadar je izveden arhitektonsko in predstavlja neko trajno oviro oz. pregrado. Kadar so kraki označeni s črtami, ki jih je možno prečkati, pravimo, da je krak označen simbolno. Fizični krak je potrebno generalizirati pri naslednjih kriterijih (DRSC, Metodologija..., marec 2007):

- dolžina kraka je krajša od Y m, ali pa je širina ožja od X m,
- dolžina kraka je krajša od Y m, ali pa deli vozišče v isti smeri,
- širina kraka je ožja od Z m.

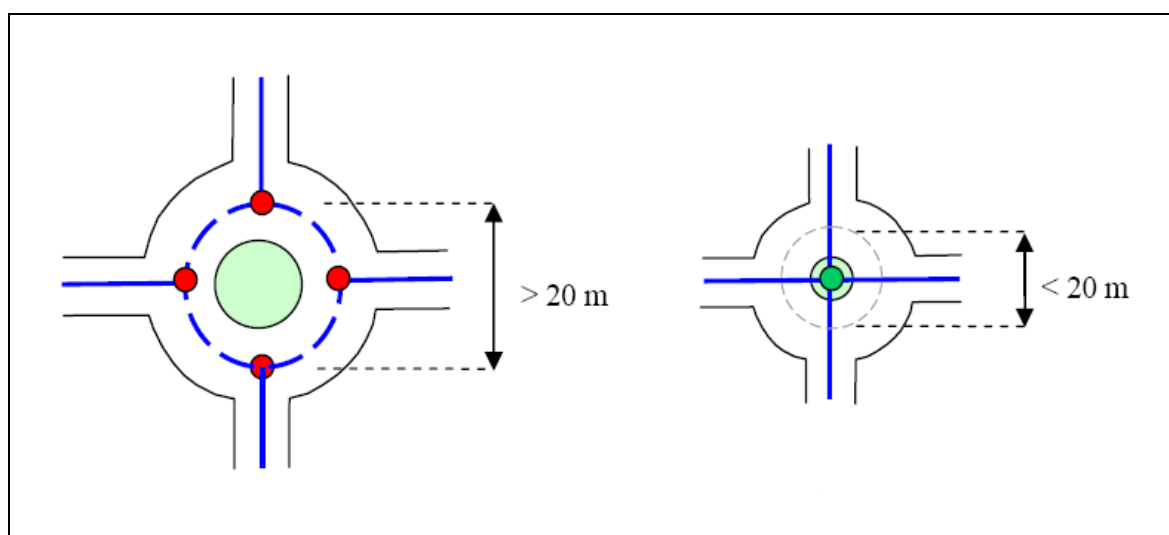
Priporočene vrednosti za X , Y in Z so: $X = 10$ m, $Y = 25$ m, $Z = 4$ m. V vseh ostalih primerih je potrebno križišča opisati z dejansko geometrijo. Na sliki 16 so prikazani nekateri primeri generalizacije krakov v križiščih glede na vrednosti X , Y in Z .



Slika 16: Generalizacija krakov v križiščih (DRSC, Metodologija..., marec 2007)

2.5.3 Krožna križišča

Krožno križišče je predstavljeno z dejansko geometrijo, če ima premer večji od 20 m. Kadar je premer krožnega križišča manjši od 20 m, se ga opiše z vozliščem. To prikazuje slika 17. V takem primeru je potrebno v opisnem delu vozlišča (križišča) voditi podatek o krožnem križišču (DRSC, Metodologija..., marec 2007).



Slika 17: Generalizacija krožnih križišč (DRSC, Metodologija..., marec 2007)

2.6 Podatkovni model opisnih podatkov

Opisni podatki služijo za opredelitev, razvrstitev, prepoznavo ali izražanje stanja izbrane entitete (v našem primeru je to osnovni grafični objekt). Vsak opisni podatek ima svoje ime, format, podatkovni tip in domeno, ki pogojuje njegove dovoljene vrednosti (Kvamme in sod. 1997).

Opisni podatki elementov digitalne osi ceste so razdeljeni v dva glavna sklopa:

- opisni podatki, ki jih vodi **DRSC**,
- **GJI** opisni podatki, ki jih vodi GURS za vpis objektov v kataster GJI.

Celotna preglednica, ki vsebuje skupaj 45 opisnih polj, je opisana v prilogi (**Priloga A**). GJI opisni podatki se sicer vodijo v bazi, vendar jih ne urejamo v fazi urejanja osi. Za to skrbi GURS, v okviru vzdrževanja katastra GJI. Na osnovi tega pa mora DRSC posodobiti svojo bazo. Zato bomo v nadaljevanju podrobneje obravnavali le opisne podatke, ki so pomembni za DRSC. Te lahko kategoriziramo v naslednje skupine (DRSC, Metodologija..., marec 2007):

- **identifikatorji,**
- **podatki za hierarhično združevanje (agregacija),**
- **podatki v zvezi z geometrijo,**
- **drugi opisni podatki.**

2.6.1 Identifikatorji

Za identifikatorje veljajo naslednja pravila:

- gre za naravno število (pozitivno, celo), običajno je števec,
- biti mora enolično, za kar mora poskrbeti aplikacija oziroma baza,
- ko je identifikator odslužil svojemu namenu (v primeru, da ukinemo objekt), ga ne smemo več uporabiti za nove objekte,
- na ta identifikator se sklicujejo vsi zapisi tega objekta.

2.6.2 Podatki za hierarhično združevanje (agregacija)

Pri delu s podatki o cestni osi prihaja do potrebe po združevanju osnovnih grafičnih objektov v hierarhično višje objekte (npr: izpeljane objekte, odseke, ceste). Združevanje omogočajo opisni podatki, ki vsebujejo identifikatorje hierarhično višjih objektov:

- identifikator višjega hierarhičnega objekta (npr. GID ima ID odseka, in ID ceste); gre za identifikator in ne za kakšne druge oznake,
- zaporedno številko nižje enote znotraj višje.

2.6.3 Opisni podatki v zvezi z geometrijo

V opisnih tabelah se nahajajo tudi podatki, ki govorijo o geometričnih lastnostih osi. Pri tem se smatra za podatke o geometričnih lastnostih le podatke, ki so v zvezi z dolžinsko komponento osi, ne pa na primer širino ceste ali podobne opisne podatke. Ti podatki so (DRSC, Metodologija..., marec 2007):

- podatki, ki ne ustrezajo dejanskim geometričnim podatkom, pač pa so privzeti iz drugih virov (npr. uradna dolžina ceste, ki ne ustreza dolžini izračunani iz geometrije),
- podatki, ki jih zaradi kompleksnosti ali dolgotrajnosti izračuna ne želimo izračunavati vedno znova iz geometrije objektov.

Opisni podatki v zvezi z geometrijo so vezani na osnovni grafični objekt. Zagotovljena mora biti konsistentnost vzdrževanja teh podatkov pri spremembah položaja osi. Ustrezno preračunane, jih vodimo tudi na izpeljanih objektih. Med podatke o geometričnih lastnostih osi štejemo **dolžine** in **stacionaže** (DRSC, Metodologija..., marec 2007).

2.6.3.1 Dolžine

Glede na vir pridobivanja podatkov o cestah ločimo dva tipa dolžin cestnih odsekov: **geometrična dolžina** (izračunana iz koordinat) in **uradna dolžina** (določena v odredbah).

Geometrično dolžino izračunamo iz koordinat digitalne osi v 2D ali 3D prostoru. Uradna dolžina pa je pridobljena iz odloka o kategorizaciji cest in je objavljena v Uradnem listu RS.

Tendenca je, da so uradne dolžine, ki jih vodimo, enake geometričnim. Zato jih je smiselno voditi na 0.01 m natančno, saj se je v preteklosti zaradi zaokroževanja na cele metre dogajalo, da skupna dolžina ceste ni ustrezala vsoti dolžin njenih delov (DRSC, Metodologija..., marec 2007).

2.6.3.2 Stacionaže

Stacionaže predstavljajo enodimenzionalni (1D) koordinatni sistem vzdolž linijskega objekta. Vsak odsek je usmerjen, kar pomeni, da ima začetek odseka stacionažo 0.00 m, konec pa stacionažo, ki je enaka dolžini odseka. Za vsak objekt ali pojav na ali ob osi ceste, lahko podamo stacionažo objekta, kar predstavlja njegovo dolžinsko koordinato vzdolž odseka. V kolikor se os ceste na nekem mestu položajno spremeni (premik osi, izravnava ovinka, ...), se spremenijo stacionaže vseh objektov od točke spremembe naprej. Tak sistem je za vodenje podatkov z geografskega informacijskega vidika dokaj neprimeren. V splošnem je namen, da se pri vodenju podatkov o cestah postopoma preide s sistema stacionaž na sistem koordinat v državnem koordinatnem sistemu (DRSC, Metodologija..., marec 2007).

V praksi je sistem stacionaž realiziran s stacionažnimi tablicami, ki so postavljene na vsakih 0.5 km, v smeri stacionaže na desni strani ceste (v kolikor je to mogoče). Določevanje in označevanje stacionaž cestnih odsekov, natančneje določa »Pravilnik o evidencah o javnih cestah«. Slika 18 prikazuje realizacijo stacionažnih oznak v naravi.



Slika 18: Stacionažna tablica; cesta 1380, odsek 404, km 12.0

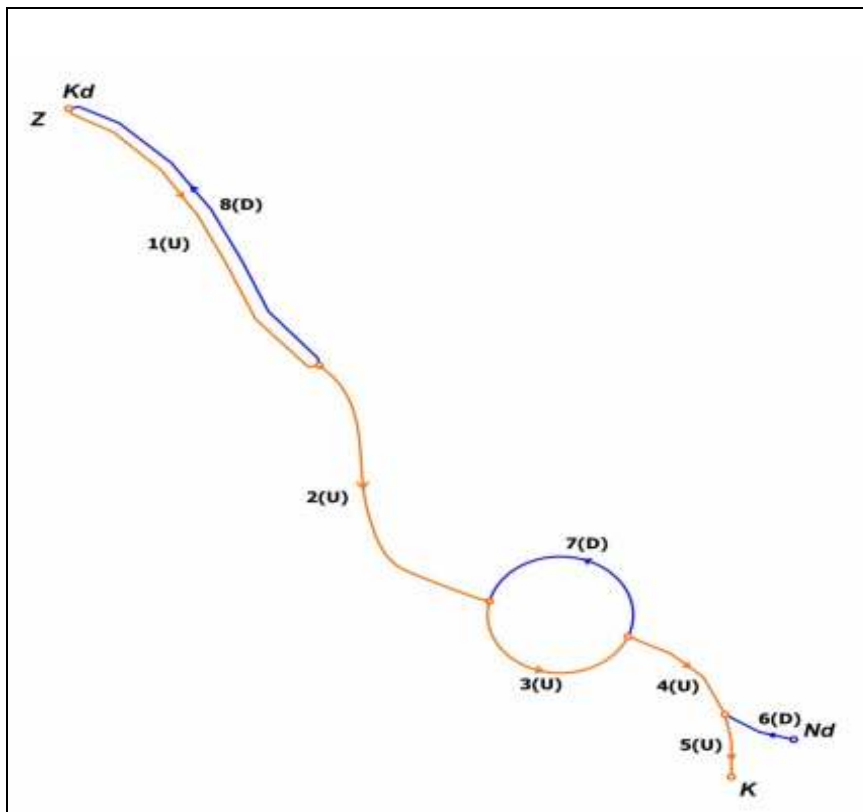
V zvezi s stacionažo so bila uvedena nekatera pravila in kontrole ter dodatni opisni podatki. Aplikacija za vzdževanje osi mora omogočati naslednje stacionažne kontrole (DRSC, Metodologija..., marec 2007):

- stacionaže med zaporednimi GID-objekti se morajo »stikati« (končna stacionaža enega objekta mora biti enaka začetni stacionaži naslednjega objekta),
- vsota stacionaž mora biti enaka oz. približno enaka uradni dolžini odseka,
- podatek o stacionaži ne sme biti prazna ali negativna vrednost, lahko pa je večja od dolžine odseka (dodana stacionaža) ali celo nič (navidezna stacionaža).

Da bi te kontrole lahko izvajali, mora obstajati podatek o **vrsti stacionaže**:

- »U« - uradna stacionaža,
- »D« - dodana (presežna) stacionaža,
- »N« - navidezna stacionaža.

Vrsto stacionaže »U« se dodeli vsem GID-objektom, ki jih potrebujemo, da pridemo od začetka do konca odseka po najkrajši poti v smeri poteka stacionaže. Vsota stacionaž teh objektov je enaka uradni dolžini odseka. Vsem morebitnim dodatnim grafičnim delom v obratni smeri stacionaže, se dodeli vrsto stacionaže »D«. Vsota stacionaž vseh GID-objektov z vrsto stacionaže »U« in »D« predstavlja skupno dolžino vozišča odseka. Navideznim delom odseka, ki jih uporabljamo zgolj za topološko povezovanje odsekov (začetna stacionaža je enaka končni), dodelimo vrsto stacionaže »N«. S tem se doseže, da je vsem grafičnim objektom dodeljena stacionaža neodvisno od stacionaže, ki je označena v naravi s stacionažnimi tablicami (DRSC, Metodologija..., marec 2007). Slika 19 prikazuje primer odseka z elementi dodane stacionaže in oštevilčevanje GID-objektov.



Slika 19: Odsek z dodano stacionažo (DRSC, Metodologija..., marec 2007)

3 IZDELAVA GIS-APLIKACIJE ZA VZDRŽEVANJE BAZE OSI DRŽAVNIH CEST

V uvodnem poglavju je bilo omenjeno, da komercialni GIS-sistemi (npr. ESRI izdelki) ne omogočajo enostavnega vzdrževanja baze cestnih podatkov zaradi svojevrstnega podatkovnega modela. Glavna težava je predvsem v dinamičnem vzdrževanju opisnih podatkov, ki so odvisni od položaja elementov osi. Zato je potrebno izdelati GIS-aplikacijo, ki bo omogočala obvladovanje teh podatkov.

Glede na namembnost ali uporabnost delimo računalniške aplikacije na horizontalne in vertikalne. Vertikalne aplikacije so povsem usmerjene v neko specifično področje (npr. bančništvo, gradbeništvo, ...). Uporabljamo jih lahko le na področju, za katerega so bile narejene. Za razliko od tega horizontalne aplikacije niso vezane na neko konkretno uporabno področje. Tipičen primer horizontalnih aplikacij so različni pisarniški paketi, kot sta Microsoft Office ali OpenOffice.

Iz navedene problematike v prejšnjem poglavju sledi, da bo naša aplikacija za vzdrževanje osi cest strogo vertikalne narave. Izdelana bo le za urejanje cestnih podatkov s točno določenim podatkovnim modelom.

Proces nastajanja vsake programske opreme, vsebuje naslednje razvojne faze: (Kvamme in sod., 1997):

- **strateško načrtovanje,**
- **sistemska analiza,**
- **sistemsko načrtovanje,**
- **izgradnja sistema,**
- **izvedba,**
- **zagon in testiranje,**
- **delovanje in vzdrževanje.**

Glede na kronološko zaporedje izvajanja razvojnih faz, poznamo dva glavna razvojna modela: (Kvamme in sod., 1997):

- **zaporedni ali kaskadni razvojni model** (vsaka faza se izvede zaporedno in v celoti),
- **postopni ali krožni razvojni model** (razvojne faze se izvajajo vzporedno).

Razvoj aplikacije za vzdrževanje osi cest je potekalo po krožnem razvojnem modelu. Zato njena predstavitev ne sledi kronološkemu zaporedju razvojnih faz, kot jih definira zaporedni razvojni model, ampak je razdelana na dve osnovni sestavini sistema. To sta podatkovna baza PostgreSQL/PostGIS in aplikacija, kot taka. Najprej je obravnavana sistemska analiza, sledi podatkovni model geografskih podatkov, organizacija podatkov v bazi, predstavitev funkcionalnosti aplikacije, ki dela s temi podatki, ter uporabniški primer, ki uporabnika vodi skozi tipičen primer uporabe.

3.1 Sistemska analiza

Odločitev o vzpostavitvi sistema in izbiri razvojnega okolja, je v veliki meri odvisna od uporabniških zahtev. Naročnik (DRSC) je podal naslednje zahteve za delovanje aplikacije za vzdrževanje omrežja osi državnih cest:

- deluje lokalno na osebem računalniku, operacijsko okolje je Windows XP,
- deluje nad odprtokodno PostgreSQL/PostGIS podatkovno bazo, ki je prav tako nameščena lokalno,
- omogoča urejanje in vzdrževanje sloja osi državnih cest v skladu z metodologijo, ki je predstavljena v prejšnjem poglavju,
- omogoča uvoz in izvoz podatkov v ESRI Shape formatu,
- omogoča vodenje in izvoz podatkov za vpis v zbirni kataster GJI,
- omogoča uvoz preglednih rastrskih in vektorskih slojev kot so DOF, skanogrami, državna meja, meje občin, ... ,
- vodi podatke o opuščeni odsekih ali delov odseka (arhiv) in planiranih odsekih,

- vodi podatke o objektih cestne infrastrukture z razpetino nad 5m.

Odločilno vlogo pri izbiri razvojnega okolja ima prav gotovo operacijski sistem, na kateremu bo nameščena aplikacija. Aplikacija je namizna, število uporabnikov je relativno majhno ter deluje na operacijskem sistemu »Okna«. V kolikor bi šlo za internetno aplikacijo, bi bilo nujno izbrati tako razvojno okolje, ki bi omogočalo namestitvev aplikacije na kateremkoli operacijskem sistemu. Takšni so npr. odptokodni GIS-programi (JUMP, uDig, gvSIG, ...), ki so v večini izdelani v programskem jeziku Java. Ena od možnosti je seveda uporaba in nadgradnja kakšnega od teh programov, kar bi bilo iz vidika prenosljivosti aplikacije zelo zaželeno. Ker pa to ni potreben pogoj, lahko uporabimo programskegi jezik C++, v katerem že obstaja kar nekaj razvitih GIS-algoritmov.

Geodetsko podjetje DFG CONSULTING d.o.o., ki je tudi izvajalec tega projekta, ima bogato tradicijo s programiranjem specializiranih, namenskih GIS-aplikacij v programskem jeziku C++. Razvojno okolje, ki se je največ uporabljalo, je Microsoft Visual C++ 6.0. Obstoječa programska koda temelji na Microsoftovi sistemski knjižnici MFC (*Microsoft Foundation Class Library*). Z razvojem programske opreme za potrebe najrazličnejših projektov, se je izoblikovalo nekaj zelo zmogljivih »geo-knjižnic«. Nekatere glavne funkcionalnosti, ki jih te knjižnice omogočajo, so:

- delo s Shapefile formatom (shplib),
- prikazovanje geolociranih piramidnih rastrskih podob (DOF),
- fotogrametrično obdelovanje podatkov,
- obdelovanje video formatov,
- osnovne funkcionalnosti GIS pregledovalnikov in urejevalnikov (oddaljevanje, približevanje, izbiranje objektov, premikanje točk, ...).

Ob pregledu obstoječe programske kode, s katero razpolaga podjetje DFG CONSULTING, lahko ugotovimo, da je izdelava aplikacije za vzdrževanje osi cest povsem izvedljiva. Odločili smo se za izdelavo aplikacije v programskem jeziku C++, razvojno okolje pa je Microsoft Visual C++ 6.0.

3.2 Podatkovni model geografskih podatkov

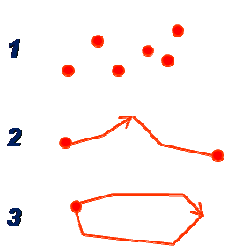
Aplikacija za vzdrževanje osi cest mora omogočati delo s podatki v shape formatu in nad odprtokodno podatkovno bazo PostgreSQL/PostGIS. V nadaljevanju sta predstavljena oba načina zapisa geografskih podatkov.

3.2.1 Struktura ESRI shape formata

Shape format je bil razvit v podjetju Environmental Systems Research Institute (ESRI), ki je v svetovnem merilu vodilno na področju tehnologije GIS. Postal je najbolj razširjen standard za obdelavo geolociranih podatkov. Zapis podatkov v shape formatu predstavljajo tri obvezne datoteke z istim imenom in naslednjimi končnicami:

- **.shp** (grafični objekti),
- **.shx** (indeksi grafičnih objektov),
- **.dbf** (opisni podatki grafičnih objektov),

Na sliki 20 je shematsko prikazana vsebina teh treh datotek.

SHP	SHX	DBF																																																		
	<table border="1"> <thead> <tr> <th>ID</th> <th>X_{min}</th> <th>Y_{min}</th> <th>X_{max}</th> <th>Y_{max}</th> <th>OFF SET</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>...</td> <td>...</td> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>2</td> <td>...</td> <td>...</td> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>3</td> <td>...</td> <td>...</td> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>...</td> <td></td> <td></td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	ID	X _{min}	Y _{min}	X _{max}	Y _{max}	OFF SET	1	2	3						<table border="1"> <thead> <tr> <th>ID</th> <th>Att1</th> <th>Att2</th> <th>Att3...</th> </tr> </thead> <tbody> <tr> <td>1</td> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>2</td> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>3</td> <td>...</td> <td>...</td> <td>...</td> </tr> <tr> <td>...</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	ID	Att1	Att2	Att3...	1	2	3			
ID	X _{min}	Y _{min}	X _{max}	Y _{max}	OFF SET																																															
1																																															
2																																															
3																																															
...																																																				
ID	Att1	Att2	Att3...																																																	
1																																																	
2																																																	
3																																																	
...																																																				
<ul style="list-style-type: none"> • VLR (Variable Length Record) • Začetne / končne točke so vozišča, vmesne točke pa lomne točke • Tri osnovni tipi objektov: <ul style="list-style-type: none"> - točka - linija - poligon 	<ul style="list-style-type: none"> • FLR (Fixed Length Record) • Indeksna tabela – povezuje grafični del (SHP) z atributnim (DBF) • Ker so SHP datoteke VLR, nam OFFSET pove kje v SHP datoteki se nahaja grafika z določenim ID-jem • Za vsak vnos imamo zapisano mejo, kje se grafika nahaja (Li. BOX - odlično za hitro iskanje) 	<ul style="list-style-type: none"> • FLR (Fixed Length Record) • Tabela z opisnimi podatki • Lahko so poljubni 																																																		

Slika 20: Shematski prikaz zapisa prostorskih podatkov v ESRI shape formatu

Vsaka entiteta v shp datoteki se ujema z natančno enim indeksom v shx datoteki in točno določenim zapisom v dbf datoteki. Vrstni red zapisov v posameznih datotekah, ki sestavljajo ESRI Shape, mora biti identičen v vseh datotekah. Npr. entiteta številka tri v shp datoteki mora biti povezana s tretjim zapisom v shx datoteki in mora imeti pripadajoče opisne podatke zapisane v tretji vrstici dbf datoteke.

V dbf tabeli se nahajajo opisni podatki, ki so načeloma dveh podatkovnih tipov: tekstovni (C) in numerični (N). Atributna polja so definirana z imenom polja, tipom, maksimalnim številom znakov v polju ter s številom decimalnih mest.

Iz kombinacije geometričnih tipov objektov (točka, linija, poligon) in njihovih razsežnosti (2D, 3D), lahko obravnavamo več tipov prostorskih podatkov, ki jih lahko hranimo v Shape formatu. Vendar pa so v eni datoteki lahko samo objekti istega tipa. Preglednica 2 prikazuje nekatere izmed možnih tipov shape objektov:

Preglednica 2: Geometrični tipi shape objektov

vrednost	geometrija	opis
0	Null Shape	brez geometrije
1	Point	2D točka
3	Polyline	2D polilinija
5	Polygon	2D poligon
11	PointZ	3D točka
13	PolylineZ	3D polilinija
15	PolygonZ	3D poligon

Obstajajo še drugi, redkeje uporabljeni tipi, kot npr. MultiPoint, PolylineM, Za potrebe naše aplikacije, ki ni namenjena pregledovanju raznovrstnih shape datotek (čeprav to omogoča), temveč je zelo namensko usmerjena, zadostuje poznavanje enostavnih tipov. Format zapisa digitalne osi državnih cest je **PolylineZ**.

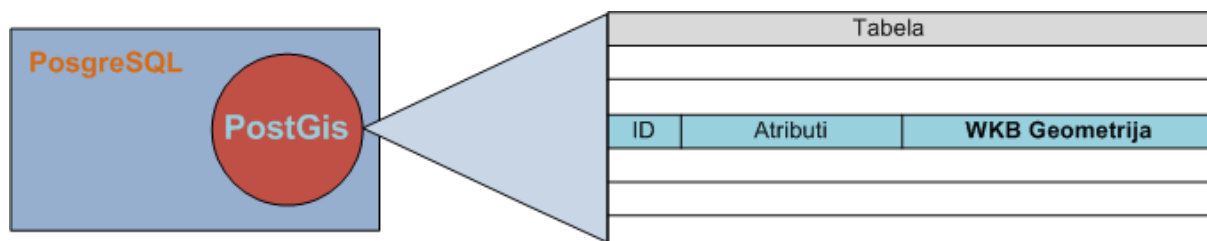
3.2.2 PostGIS – WKB format

PostgreSQL je odprtokodna, objektno-relacijska podatkovna baza. Razvoj sistema se je začel pred dvajsetimi leti na kalifornijski univerzi Berkeley. Danes za razvoj skrbi mednarodna skupnost razvijalcev, neodvisno od komercialnih podjetij. PostgreSQL je postal robusten in zelo zmogljiv sistem, ki se lahko primerja z največjimi komercialnimi sistemi. Namestitev sistema je mogoča na vseh pomembnejših operacijskih sistemih, kot so Linux, UNIX in Windows. Programska oprema (izvorna koda, inštalacijski paketi, programski vmesniki, grafični vmesniki, ...) je na voljo v različnih programskih jezikih pod pogoji BSD licence (*Berkeley Software Distribution*), ki dovoljuje uporabo in razširjanje brez omejitev. Dovoljuje uporabo programske opreme tako za komercialne, kot nekomercialne namene. Samo nekateri izmed programskih jezikov, za katere obstajajo programski vmesniki oz. knjižnice so: C/C++, Java, .Net, Perl, Python, Ruby,... . PostgreSQL temelji na SQL 92/99 standardu in izpolnjuje vse pogoje, ki veljajo za objektno-relacijske baze (www.postgresql.org).

Vendar pa PostgreSQL sam po sebi ne omogoča dela z prostorskimi podatki. To funkcionalnost prinaša šele nadgradnja sistema z vtičnikom **PostGIS**, ki vsebuje definicije geometričnih tipov in funkcionalnost, ki omogoča GIS-analize nad prostorskimi podatki v bazi. PostGIS se kontinuirano razvija od leta 2001 v kandskem podjetju **Refractions Research** in je prav tako odprto-kodne narave. Na voljo je pod pogoji GPL (*General Public Licence*) licence. Razvoj vtičnika PostGIS sledi smernicam OGC/OpenGis (Open Geospatial Consortium) standardom za geografske podatke (www.postgis.org).

Skupina objektov enega geometričnega tipa je predstavljena z eno tabelo v PostGIS bazi. Vsak objekt je predstavljen z eno vrstico oz. enim zapisom v tej tabeli. Grafični objekt je sestavljen iz:

- **enoličnega identifikatorja**; pozitivno celo število,
- **ne-geometričnih opisnih podatkih**; standardni SQL podatkovni tipi,
- **podatka o geometriji objekta**; WKB (Well Known Binary).



Slika 21: Shematski prikaz PostGIS strukture prostorskih podatkov

Opisna polja tabele so največkrat definirana s standardnimi SQL podatkovnimi tipi, lahko pa so tudi bolj kompleksna. Geometrija je npr. zapisana v posebnem geometrijskem polju (omogoča PostGIS). Zapis je binaren (šestnajstiški) v WKB formatu. Primeren je za shranjevanje podatkov v bazi, za interpretacijo pa je primernejši WKT format (Well Known Text). Omenjena formata sta določena po OGC standardizaciji. Za primerjavo si oglejmo oba zapisa za točko T (1,1):

- WKT: POINT(1, 1),
- WKB: 010100000000000000000000F03F000000000000F03F.

Podobno kot pri shape formatu obstaja tudi tu več geometričnih tipov oz. WKB objektov. Spodnja preglednica 3 prikazuje nekatere izmed enostavnejših geometričnih tipov.

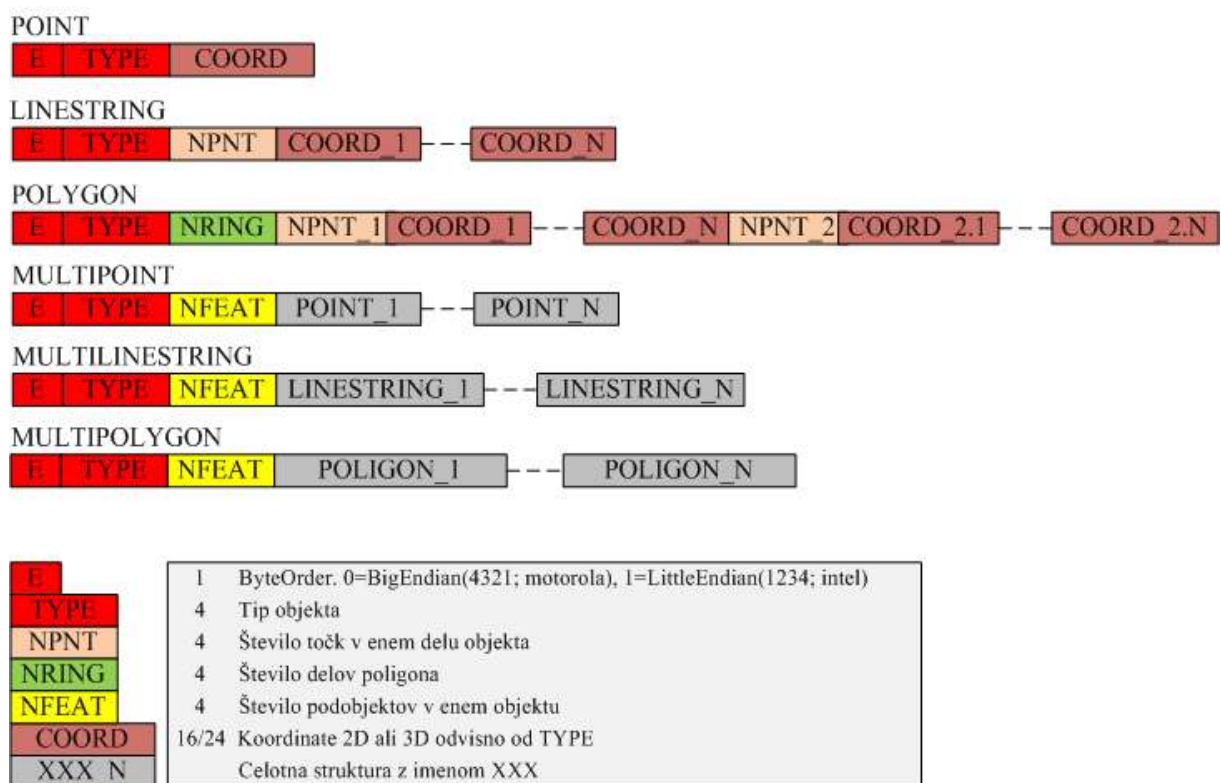
Preglednica 3: Osnovni geometrični tipi po OGC standardizaciji

vrednost	geometrija	opis objekta
1	Point	točka
2	Linestring	polilinija
3	Polygon	poligon
4	MultiPoint	več točkovni
5	MultiLinestring	več linijski
6	MultiPolygon	več poligonski

Našteti geometrični tipi objektov imajo še svoje variacije v razsežnost (2D, 3D). Obstajajo še nekateri drugi, kompleksnejši geometrični tipi, kot so npr.: PointZM, MultiPoligonM, MultiLineStringZM, GeometryCollecton, Ti so težavnejši za obravnavo. V poglavju 2.3.1 smo povedali, da osnovni grafični objekt cestnega omrežja ne sme biti sestavljen iz več delov. Zato stremimo k temu, da uporabljamo samo enostavne geometrične tipe. Digitalna os državnega cestnega omrežja je tipa **Linestring** v trirazsežnem prostoru (3D koordinate).

3.2.2.1 Struktura WKB objektov

WKB format je zapisan v šestnajstiškem številskem sistemu (znaki: 0-9 in A-F). Vsak geometrični tip ima svojo strukturo zapisa. Ta je prikazana na sliki 22.

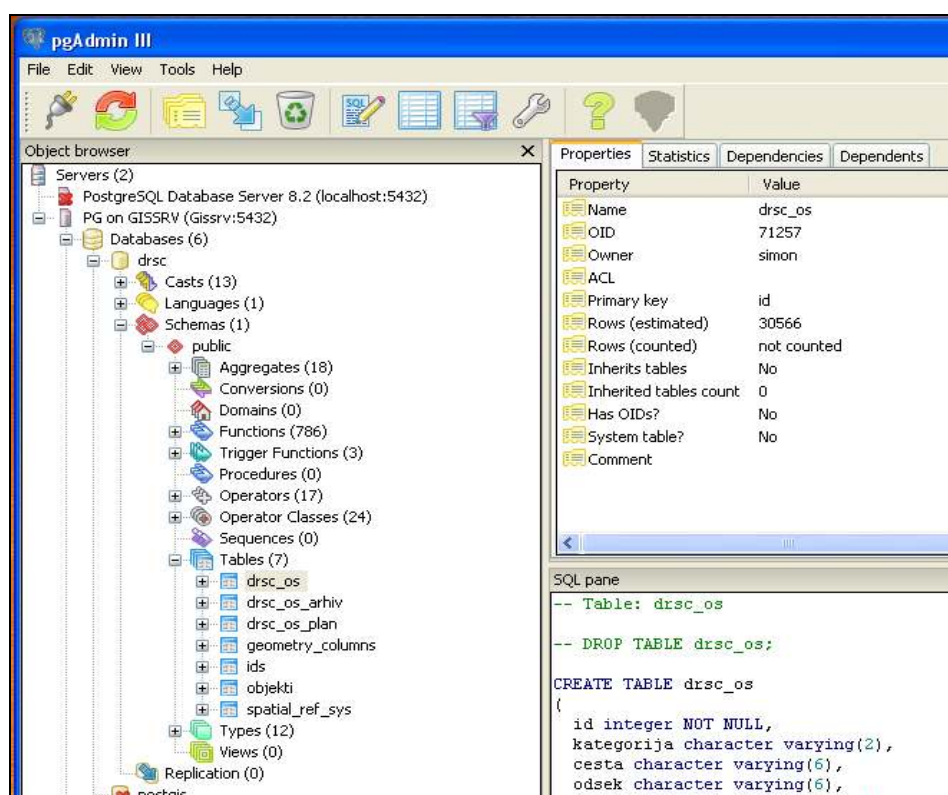


Slika 22: Struktura WKB objektov

3.3 Konstrukcija in upravljanje s podatkovno bazo PostGIS

3.3.1 Administracija baze

Pod izrazom administracija baze razumemo začetno vpostavitve baze, tabel, uporabnikov, spreminjanje nastavitvev, vpogled v stanje in vzdrževanje baze, itd. Za to je najprimernejša uporaba grafičnega vmesnika PgAdminIII, ki se namesti ob namestitvi baze (zaslonska slika je prikazana na sliki 23). Na voljo imamo tudi nekatere ukazne programčke, ki jih lahko uporabimo za zaganjanje daljših SQL-procedur (navadno preko *.bat datotek). Pravzaprav je jezik, ki ga podpira baza PostgreSQL/PostGIS, proceduralni jezik PL/pgSQL. To je razširitev standardnega SQL-jezika, ki omogoča združevanje in izvajanje večih ukazov na strani baze. Omogoča tudi ustvarjanje novih funkcij in sprožilcev procedur nad podatki v bazi. Poleg administracije, omogočajo ti ukazni programčki še uvoz in izvoz shape datotek (*shp2pgsql.exe*, *pgsql2shp.exe*), varnostno kopiranje baze, ponovno vzpostavljanje baze (*pg_dump.exe*, *pg_restore.exe*), itd.



Slika 23: Izgled administratorskega grafičnega vmesnika za PostgreSQL/PostGIS

Tabele v bazi ustvarjamo, preurejamo, brišemo, itd. s standardnimi SQL-ukazi: CREATE, ALTER, DROP, Posebnost je polje za geometrične podatke, ki se ustvari posebej, preko PostGIS funkcije *AddGeometryColumn*. Nad tabelami lahko vzpostavimo tudi določene omejitve, indekse in pravila, ki določajo način zapisovanja podatkov v tabelo. Spodaj je prikazan način vzpostavitve podatkovne tabele s sestavljenim PL/pgSQL-ukazom, ki smo ga uporabljali za ustvarjanje tabel. Ukaz ustvari tabelo z imenom 'naselja', doda stolpec za geometrijo ('the_geom') in ustvari t.i. prostorski indeks ('naselja_gist').

```
CREATE TABLE naselja (id integer NOT NULL, ime varchar(30),  
povrsina numeric, y_c integer, x_c integer);  
SELECT AddGeometryColumn('public','naselja','the_geom',-1,'POLYGON',2);  
CREATE INDEX naselja_gist ON naselja USING gist (the_geom);
```

Z zgornjim ukazom se ustvarijo štiri omejitve in en indeks:

- primarni ključ na polju z identifikatorji (zahteva unikatno, neprazno vrednost),
- omejitev dimenzije podatkov (podatki istih dimenzij),
- omejitev gometričnega tipa podatkov (podatki iste geometrije),
- omejitev referenčnega sistema (podatki istega referenčnega sistema),
- gist indeks (prostorsko indeksirani podatki za hitrejše prostorsko iskanje).

3.3.2 Primarni uvoz podatkov v bazo

Zgoraj je bil omenjen ukazni program *shp2pgsql*, ki omogoča uvoz shape datotek v bazo. Vendar pa pri tem nimamo nikakršne kontrole nad načinom vpostavitve tabele in dodelitve tipa geometrije. Pri testiranju uvoza podatkov s tem programom je ugotovljeno naslednje:

- program je zelo odziven,
- tabelo generira na podoben način, kot je opisan v poglavju zgoraj, vendar generira dodatno, novo polje z unikatnimi identifikatorji,
- geometrični tip, ki ga dodeli objektom, je vedno tipa 'MULTI'.

V prejšnjem poglavju je rečeno, da več-delnih objektov ne podpiramo. Zato smo se odločili, da bomo podatke uvažali v bazo kar preko naše aplikacije. Razvili smo dodatno administratorsko funkcionalnost, ki omogoča ustvarjanje, brisanje tabel in tudi uvoz podatkov v bazo. Postopek je zelo enostaven: na podlagi strukture opisne tabele in tipa geometričnih objektov sestavimo SQL-ukaz ter ustvarimo tabelo v bazi, shape datoteko uvozimo v aplikacijo, potem pa vse objekte na tem sloju zapišemo v bazo. S tem smo dosegli popolno skladnost med našo aplikacijo in podatki v bazi.

3.3.3 Nadzor nad dodeljevanjem unikatnih identifikatorjev

Objekti, zapisani v podatkovnih bazah, morajo oz. je priporočljivo da imajo dodeljene enolične identifikatorje, po katerih se lahko sklicujemo na te objekte. S prej omenjenim načinom ustvarjanja tabel, določimo katero polje je tisto, ki vsebuje enolične identifikatorje. Za dodeljevanje identifikatorjev objektom, ki se na novo zapisujejo v bazo, lahko skrbi baza, lahko pa nadzor prevzamemo sami. Odločili smo se za slednjo možnost.

Rešitev smo poiskali z uvedbo dodatne tabele v bazi, ki vsebuje zadnje uporabljene identifikatorje v posamezni tabeli v bazi. Pred zapisovanjem podatkov v bazo pridobimo zadnji uporabljen identifikator za željeno tabelo. Na osnovi števila objektov, ki jih želimo zapisati v bazo, povečamo vrednost identifikatorja na novo vrednost, objektom pa dodelimo identifikatorje znotraj tega intervala. Preglednica 4 prikazuje strukturo tabele z zadnjimi uporabljenimi identifikatorji.

Preglednica 4: Struktura tabele z zadnjimi uporabljenimi identifikatorji

ime tabele	ime polja z ID-ji	zadnji ID
Tabela A	ID_Polje (A)	N (A)
Tabela B	ID_Polje (B)	N (B)
Tabela C	ID_Polje (C)	N (C)
...

Podatkovne baze v splošnem omogočajo delo več uporabnikov hkrati. V našem primeru sta tako baza kot aplikacija nameščeni lokalno, predviden pa je en sam uporabnik. Kljub temu pa smo zapisovanje podatkov v bazo načrtovali za več uporabnikov. To je sicer velika prednost, ki jo omogočajo podatkovne baze, vendar je pri programiranju aplikacij, ki delujejo nad bazami, potrebna velika previdnost, da ne pride do konfliktnih situacij med uporabniki. Pri dodeljevanju identifikatorjev obstaja namreč možnost, da dva uporabnika istočasno ali pa z zelo majhnim časovnim zamikom, zahtevata isti zadnji ID. V tem primeru bi prišlo do prekrivanja intervala dodeljenih identifikatorjev, shranjevanje takšnih objektov pa baza zavrača. Konfliktnim situacijam se izognemo z upotabo SQL-ukaza '**SELECT FOR UPDATE**', ki zaklene zahtevani zapis v tabeli dokler se ne izvede osvežitev zapisa oz. do konca trenutne transakcije. Med tem časom so drugi uporabniki 'na čakanju' za pridobitev identifikatorjev.

Da bi se o tem lahko prepričali, sem sestavil majhen ukazni program, ki je stalno v naključnih majhnih časovnih presledkih zahteval naključno število identifikatorjev. Program sem zagnal na petih računalnikih hkrati. Deloval je približno pol ure. Pri tem se je zapisovala datoteka s statistiko vseh transakcij. Analiza rezultatov je pokazala, da ni prišlo do nobenih prekrivaj med intervali pridobljenih identifikatorjev.

3.4 Razvoj in funkcionalnost aplikacije

Aplikacija je izdelana v razvojnem okolju Visual C++ 6.0, z uporabo Microsoftove systemske knjižnice MFC. Projektu, in s tem programu, smo dali delovno ime **ZajeC**, kar izhaja iz besedne zveze "zajem cest". ZajeC je eno-dokumentni grafični program, ki ima vse funkcionalnosti splošnega pregledovalnika geografskih podatkov, funkcionalnost urejevalnika pa je izdelana specifično za potrebe vzdrževanja osi državnih cest.

V nadaljevanju je podana namestitev osnovnih kontrol za spreminjanje pogleda, iziranja in urejanja objektov. Sledi specifikacija podatkovnih slojev, način uvoza in shranjevanja podatkov, ter procedure za urejanje osi.

3.4.1 Namestitev osnovnih kontrol pogleda, izbiranja in urejanja

Vsaka GIS-aplikacija, ki omogoča urejanje grafičnih podatkov, mora imeti vgrajeno funkcionalnost spreminjanja pogleda (približevanje, oddaljevanje, premikanje po mapi), izbiranja objektov ter urejanje grafičnih in opisnih podatkov. V splošnem je odločitev o tem, kam je bolje namestiti neko kontrolo, odvisna od pogostosti uporabe le te. Kontrole, ki se ne uporabljajo zelo pogosto, so nameščene v menijski vrstici. Bolj pogosto uporabljene izpostavimo na orodno vrstico. Najpogosteje uporabljene kontrole pa morajo imeti tudi (ali pa samo) bližnjice preko tipkovnice in miške. V nadaljevanju navajam dotične funkcije in njihovo namestitev.

Spreminjanje pogleda:

- stopenjsko približevanje: Alt + desni klik ali Alt + kolesček na miški naprej,
- stopenjsko oddaljevanje: Alt + levi klik ali Alt + kolesček na miški nazaj,
- približevanje z določitvijo območja: Alt + desni klik + vlečenje miške,
- pogled na celotno območje podatkov: Alt + Shift + levi klik,
- pogled na območje trenutno aktivnega sloja: menijska in/ali orodna vrstica,
- pogled na območje trenutno izbranih objektov: menijska in/ali orodna vrstica.

Izbiranje objektov:

Izbiranje objektov poteka na trenutno aktivnem sloju. Način za izbiranje objektov se aktivira s pritiskom na tipko S (ang. select). Kontrole za izbiranje objektov so naslednje:

- grafična izbira enega samega objekta: desni klik,
- posamezno dodajanje objektov k izbranim: Shift + desni klik,
- posamezno odstranjevanje objektov iz izbranih: Ctrl + desni klik,
- izbor vseh objektov: Ctrl + tipka A (angl. all),
- odstranitev vseh izbranih iz izbranih: Esc (angl. escape),
- izbiranje po atributih: tipka F (angl. find), menijska in/ali orodna vrstica – dialog.

Urejanje grafičnih in opisnih podatkov:

- ustvarjanje novega objekta: tipka N (angl. new), sledi digitalizacija z klikanjem,
- končanje objekta: tipka Q (angl. quit), sledi določitev opisnih podatkov,
- urejanje atributov: tipka E (angl. edit), prikaže se dialogno okno za vnos podatkov,
- premikanje točk izbranega objekta: tipka M (angl. move),
- vstavljanje točke na obstoječi objekt: tipka I (angl. insert),
- brisanje obstoječih objektov: desni klik (izbira objekta), tipka Delete,
- združevanje dveh označenih objektov: tipka J (angl. joint),
- razdelitev enega objekta na dva na izbranem verteksu: tipka B (angl. break).

3.4.2 Podatkovni sloji

Podatkovni sloji predstavljajo t.i. nosilce podatkov. Kot je že bilo omenjeno, je program ZajeC zasnovan zelo namensko. To pomeni, da omogoča urejanje samo tistih podatkov, ki so za to predvideni. Ni omogočeno dodajanje novih podatkovnih slojev, spreminjanje obstoječih nastavitev, pa je omejeno na funkcionalne potrebe. Podatki so znotraj programa organizirani po podatkovnih slojih, ki so v naprej definirani in prilagojeni podatkovnemu modelu podatkov. Vsi sloji se vzpostavijo ob zagonu aplikacije.

V splošnem obstajata dve vrsti slojev, na katere se uvaža podatke: **vektorske** in **rastrske**. Podatkovni sloj je definiran z geometričnim tipom shape objektov, strukturo opisnih podatkov (za sloj cest glej prilogo A), načinom izrisovanja in še nekaterimi drugimi nastavitvami. Vektorski sloji lahko torej vsebujejo le podatke v shape formatu. Rastrski sloji pa so izpeljani iz vektorskih v smislu obrisa geolociranih slik (poligoni), kamor se nalaga vsebino. Sloji so lahko definirani kot dinamični (odklenjeni, urejanje podatkov je dovoljeno) ali kot statični (zaklenjeni za urejanje). Rastrski so praviloma statične narave, vektorski pa dinamične. Preglednica 5 podaja specifikacijo slojev v programu ZajeC.

Preglednica 5: Definicija slojev v aplikaciji

ime sloja	tip shp geometrije	urejanje	vir
Državne ceste	PolylineZ	DA	baza
Državne ceste - arhiv	PolylineZ	DA	baza
Državne ceste - plan	PolylineZ	DA	baza
Objekti	PolylineZ	DA	baza
Občinske ceste	PolylineZ	NE	shape
Gozdne ceste	PolylineZ	NE	shape
Nekategorizirane ceste	PolylineZ	DA	baza
Izpeljani objekti	PolylineZ	NE	shape
Križišča	PointZ	DA	baza
Državna meja	PolylineZ	NE	baza
Občinske meje	Polygon	NE	baza
Območja upravljanja	Polygon	NE	baza
Skanoگرامi	Polygon - Raster	NE	tif + tfw
DOF5, DOF1	Polygon - Raster	NE	tif + tfw

Iz zgornje preglednice 5 je razvidno, kateri sloji so dinamični in kateri statični. Navedeni so tudi viri posameznih podatkov oz. od kod se uvažajo. Vsi dinamični podatki se nahajajo v podatkovni bazi PostGIS. V bazi so shranjeni tudi nekateri vektorski pregledni sloji (državna meja, občinske meje, območja upravljanja). Ostali pregledni sloji se nahajajo na podatkovnem disku in se jih lahko uvozi po potrebi.

3.4.3 Uvoz in shranjevanje podatkov

Aplikacija je zasnovana tako, da je praktično vseeno, od kod se podatki uvažajo. Obdelava podatkov pa poteka interno nad shape objekti. Če so podatki formatirani drugače, kot npr. v primeru podatkovne baze PostGIS, potrebujemo t.i. pretvornik podatkov.

Za uvoz in shranjevanje **shp datotek** se uporabi knjižnica **shplib**, ki je zelo optimizirana in preverjeno dobro deluje. Pri uvozu podatkov iz shape datotek, se vsi podatki naložijo v pomnilnik računalnika. Sledi faza urejanja podatkov in nato shranjevanje. Pri shranjevanju se shranijo vsi podatki nazaj v isto shape datoteko, ali pa v novo datoteko (ukaz: Shrani kot). Za vhodno-izhodne operacije se uporabljata le dva ukaza: **uvozi in izvozi** podatke.

Pri prenosu podatkov med bazo in aplikacijo, je zgodba nekoliko drugačna. Prednost podatkovnih baz je v tem, da nam ni potrebno vedno znova shranjevati vseh podatkov, pač pa samo tiste, ki so se med uporabo spremenili. Uporabnik lahko tudi določi, katere podatke bo uvozil iz baze in katere bo zapisal ali osvežil v bazi. V splošnem se uporablja štiri osnovne vhodno-izhodne ukaze, ki zagotavljajo uspešno vzdrževanje baze podatkov:

- **izberi** podatke iz baze – SELECT,
- **zapiši** nove podatke v bazo – INSERT,
- **posodobi** obstoječe, a spremenjene podatke – UPDATE,
- **briši** podatke, ki jih ne potrebujemo iz baze – DELETE.

Za realizacijo teh ukazov pa je potreben pretvornik, ki bo ustrezno pretvoril WKB objekte v shape objekte in obratno.

3.4.3.1 Pretvornik geometričnih podatkov (WKB <--> SHP)

Uspešen prenos opisnih podatkov smo zagotovili z identično strukturo tabel v aplikaciji in v bazi. Problem ostaja še geometrični del podatkov, ki so shranjeni v bazi. Geometrijo objektov smo pridobili na identičen način, kot ostale opisne podatke, le da je ta v šestnajstiškem WKB formatu (glej poglavje 3.2.2). Zato je bilo potrebno narediti pretvornik iz WKB objektov v SHP objekte in obratno. Za prvi del, kar pomeni branje in pretvarjanje šestnajstiškega zapisa, smo funkcije sestavili sami. Pri obratnem postopku, to je zapisovanju v bazo, pa smo

uporabili PostGIS funkcijo **GeomFromEWKT**. Argument funkcije je formatirani tekstovni zapis koordinat shape objektov (WKT), ki ga funkcija pri izvedbi SQL ukaza, pretvori v šestnajstiški WKB format.

3.4.3.2 Programski vmesnik libpq

Za komunikacijo med aplikacijo in **PostgreSQL/PostGIS** bazo, se uporabi statična knjižnica za programski jezik C, **libpq**, ter nekatere dinamične knjižnice (*.dll), ki se namestijo ob namestitvi baze. Libpq je zbirka programskih funkcij, ki omogočajo uporabniškimi programom vzpostavitev povezave z bazo, izvrševanje poizvedovanj po njej in sprejemanje vrnjenih rezultatov. Rezultate poizvedovanja se pridobijo v obliki tabele rezultatov (objekt PGresult), katere pretvorimo v svoje objekte (shape objekti). Spodnja preglednica 6 navaja nekatere osnovne funkcije, ki se jih uporablja za komunikacijo z bazo PostgreSQL.

Preglednica 6: Osnovne funkcije programske knjižnice libpq

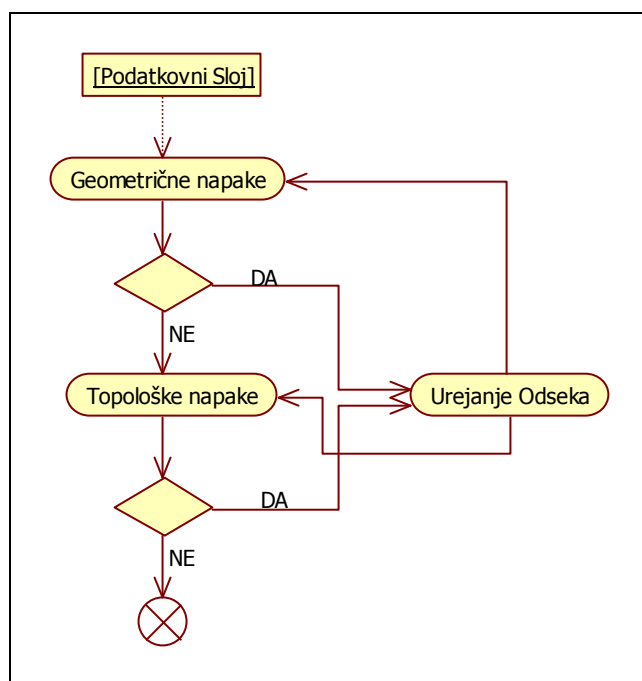
ime funkcije	parametri funkcije	opis
PQconnectdb	parametri za povezavo	vzpostavi novo povezavo z bazo
PQfinish	objekt povezave – PGconn	zapre povezavo z bazo
PQstatus	objekt povezave - PGconn	preveri status povezave
PQexec	PGconn, SQL stavek	izvede ukaze in vrne rezultate
PQresulStatus	objekt rezultatov – PGresult	preveri status izvršitve ukaza
PQgetvalue	PGresult, št. vrstic, št. stolpev	vrne vrednost enega polja
PQclear	objekt rezultatov – PGresult	počisti rezultate v pomnilniku

3.4.4 Programske procedure pri urejanju osi ceste

Urejanje osi v osnovi poteka na nivoju osnovnega grafičnega objekta. Ker pa so opisni podatki posameznih GID-objektov nekega odseka, tesno povezani s položajem osi, je predmet urejanja največkrat kar celoten odsek. Omeniti velja, da urejanje poteka nad objekti, ki so naloženi v pomnilnik računalnika. Uporabnik mora poskrbeti za tak način uvažanja podatkov, da ima na voljo celotne obstoječe odseke iz baze. V nasprotnem primeru bi urejanje izvajali na nepopolnih odsekih, kar bi vodilo k napačnim rezultatom (napačna določitev stacionaž). Da do takšnih napak ne bi prihajalo, se uvaža celotna baza osi državnih cest že kar ob zagonu programa. Uporabnik pa ima tudi možnost, da naknadno uvozi le objekte, ki jih definira sam, z ustreznim SQL-ukazom preko dialognega okna (npr. samo elemente enega odseka).

Predpogoj za pravilno ureditev osi ceste je, da morajo imeti vsi GID-objekti dodeljen opisni podatek o odseku in vrsti stacionaže. Če ta pogoj ni izpolnjen, ne vemo kateremu odseku pripada, posledično pa so rezultati obdelave tega odseka napačni. Podatek o vrsti stacionaže pa je pomemben za pravilno preoštevilčevanje in določitev stacionaž vseh GID-objektov nekega odseka. Na podlagi teh dveh podatkov lahko določimo vse ostale, dinamične opisne podatke (dolžina GID-objekta, dolžina odseka, zaporedna številka GID-objekta znotraj odseka, začetna in končna stacionaža vsakega GID-objekta v odseku). Zato smo razvili enostavno proceduro, ki vsakemu objektu na sloju cest preveri veljavnost podatka o odseku in vrsti stacionaže ter omogoči odpravo napak.

Urejanja osi se lahko lotimo na dva načina oz. iz dveh nivojev. Prvi način omogoča sistematično urejanje baze iz vidika celotnega sloja osi, drugi pa omogoča lokalno urejanje posameznih odsekov. Vendar pa preko prvega načina, ki vsebuje le odkrivanje geometričnih in topoloških napak, preidemo na drugi lokalni način, ki poleg naštetega omogoča popolno vsebinsko urejanje oz. preurejanje odsekov. Ko odkrijemo kakšno geometrično ali topološko napako na enem odseku, moramo zaradi že znanih razlogov urediti celoten odsek. Potem grafičnem urejanju odseka sledi zagon procedur, ki preračunajo opisne podatke elementov odseka. Postopek je iterativen. Spodnja slika 24 prikazuje postopek vspostavitve geometrično in topološko urejene baze osi državnih cest.



Slika 24: Diagram poteka urejanja sloja osi cest

3.4.4.1 Geometrične in topološke kontrole

Geometrične in topološke kontrole so lahko torej izvajajo na dveh nivojih: na nivoju celotnega sloja, ali na nivoju posameznega odseka. Razlika je le v hitrosti obdelave, saj je količina podatkov, ki jih je potrebno obdelati, na nivoju odseka mnogo manjša.

Pri geometričnih kontrolah se sprožijo procedure, ki na vsakem od podanih objektov ugotavljajo ali:

- ima vsak linijski objekt vsaj dve točki: štetje točk objekta,
- zaporedne točke niso istoležne oz. ali niso preveč skupaj: računanje razdalje med dvema zaporednima točkama, dolžinska toleranca je 1m,
- polilinija nima ostrih zavojev: računanje razlike smernega kota med dvema paroma treh zaporednih točk, kotna toleranca je 15°.

Pri topoloških kontrolah pa se sprožijo procedure, ki ugotavljajo:

- ali se polilinije stikajo samo v vozliščih (stikanje v lomnih točkah ni dovoljeno),

- viseča vozlišča (operater se na podlagi interpretacije odloči ali gre za napako ali ne),
- sekanje polilinijskih samih s seboj: računanje preseka dveh daljic iste polilinijske,
- sekanje med dvema polilinijskima (operater interpretira in se odloči).

3.4.4.2 Urejanje odsekov

Proces urejanja odseka sestoji iz dveh faz: ročnega urejanja grafičnih in opisnih podatkov, ter avtomatske določitve opisnih podatkov v zvezi z geometrijo. Ti se morajo ponovno določiti ob vsaki spremembi položaja osi. Pri urejanju odsekov, lahko pride do naslednjih situacij:

- spremenili nismo nobenega objekta: v bazi se ne spremeni nič,
- enemu ali večim objektom smo spremenili samo nekatere opisne podatke: v bazi se osvežijo le objekti, ki so jim bili spremenjeni opisni podatki,
- enemu ali večim objektom smo spremenili geometrijo: sprožijo se procedure, ki preračunajo podatke za celoten odsek, v bazi se osvežijo objekti celotnega odseka.

Avtomatsko urejanje odsekov vsebuje precej zapletene procedure, saj je treba elemente odseka ustrezno preštevilčiti na osnovi geometrijskega zaporedja in usmerjenosti le teh. To je predpogoj za pravilno določitev začetne in končne stacionaže. Poglavitne procedure za pravilno urejanje odseka, ki si morajo slediti v sledečem vrstnem redu, so:

- generiranje odseka,
- preštevilčevanje elementov odseka,
- izračun grafične dolžine odseka in stacionaž elementov odseka,
- določitev ostalih opisnih podatkov,
- shranjevanje sprememb v podatkovno bazo.

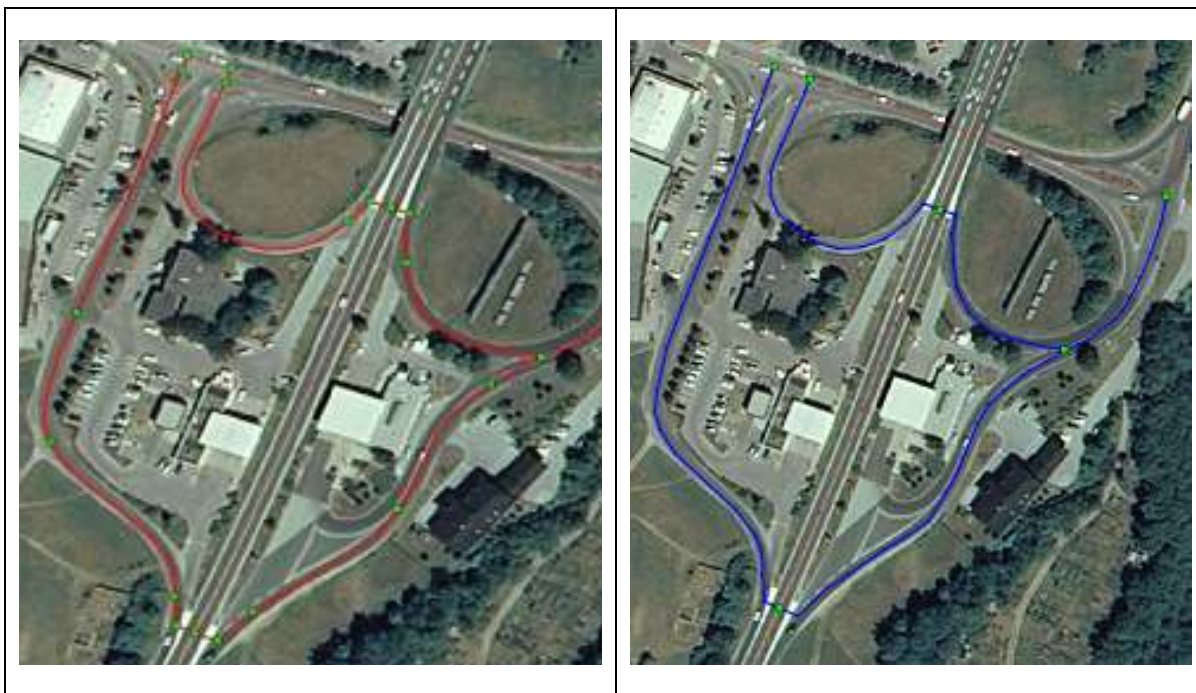
3.4.4.3 Ustvarjanje izpeljanih objektov

V poglavju 2.4 je razloženo kaj so izpeljani objekti, in kakšen je njihov namen. Naročnik jih bo uporabljal za izdelavo različnih analiz državnih cest v drugih, splošnejših GIS-orodjih.

Izpeljani objekti se ustvarijo v nekem trenutku iz osnovnih grafičnih elementov. Celotna geometrija GID-objektov, razen podvojenih točk na stikih objektov, se prenese na izpeljane objekte. Za njih je pripravljen tudi poseben sloj v aplikaciji, kamor jih nalagamo ob ustvarjanju. Ohrani se tudi enaka struktura opisnih podatkov, njihove vrednosti pa se ustrezno preračunajo. Dejansko gre za iste algoritme, kot pri urejanju odsekov: preoštevilčevanje in določitev stacionaž. Nato se tako ustvarjeni objekti izvozijo v shape datoteko, katero se lahko uporablja v drugih GIS-orodjih.

Osnovni algoritem pri sestavljanju izpeljanih objektov, je ugotavljanje načina stika med dvema GID-objekti odseka. Potrebno pa je definirati še dodatne ustavitvene pogoje, kjer se morajo izpeljani objekti zaključiti:





- na prehodih med dvema odseki,
- na križiščih najmanj treh GID-objektov istega odseka.



Slika 25: Prikaz števila GID-objektov prvotnega odseka (levo) in izpeljanega odseka (desno)

3.4.4.4 Nekatere dodatne funkcije

Aplikacija ponuja še nekatere dodatne funkcije, ki uporabniku olajšujejo delo:

- **ustvarjanje rondoja na osnovi treh točk**; ikona: ,
- **obračanje smeri izbranih linijskih objektov in prikaz usmerjenosti**; ikona: ,
- **izrisovanje vozlišč in lomnih točk na posameznem sloju**; ikoni: ,
- **izris osi cest po kategorijah**; ikona: ,
- **prenos podatkov med podatkovnimi tabelami v bazi.**

3.5 Prikaz uporabe aplikacije

Za lažjo predstavitev o samem programu in njegovi funkcionalnosti, je v nadaljevanju v šestih korakih prikazana tipična uporaba aplikacije za urejanje osi cest. Prikazan je primer, ko uporabnik odkrije napako na delu enega odseka, ga uredi, ponovno določi opisne podatke vseh GID-objektov tega odseka in shrani spremenjene podatke. Skozi postopek niso prikazane vse funkcionalnosti, pač pa le nekatere, pri katerih so opazne grafične spremembe.

	<p>Korak 1:</p> <p>V kolikor so v konfiguracijski datoteki pravilno navedeni parametri za povezavo na PostgreSQL bazo, se ob zagonu programa uvozita dva sloja: pregledni sloj državne meje in sloj državnih cest. Če se ob zagonu ti podatki ne prikažejo, je potrebno nastaviti parametre za povezavo in podatke ponovno uvoziti.</p>
---	--



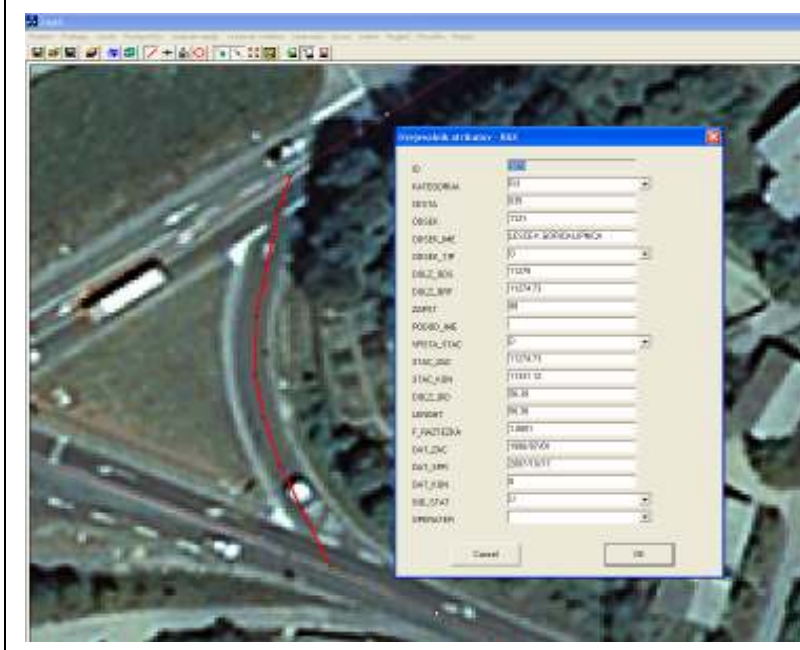
Korak 2:

Prikazana je osnovna funkcionalnost pregleda podatkov. Izrisana so vozlišča, lomne točke, in usmerjenost posameznih GID-objektov. Podlaga je DOF5. Prikazano je dialogno okno za iskanje objektov po opisnih podatkih.



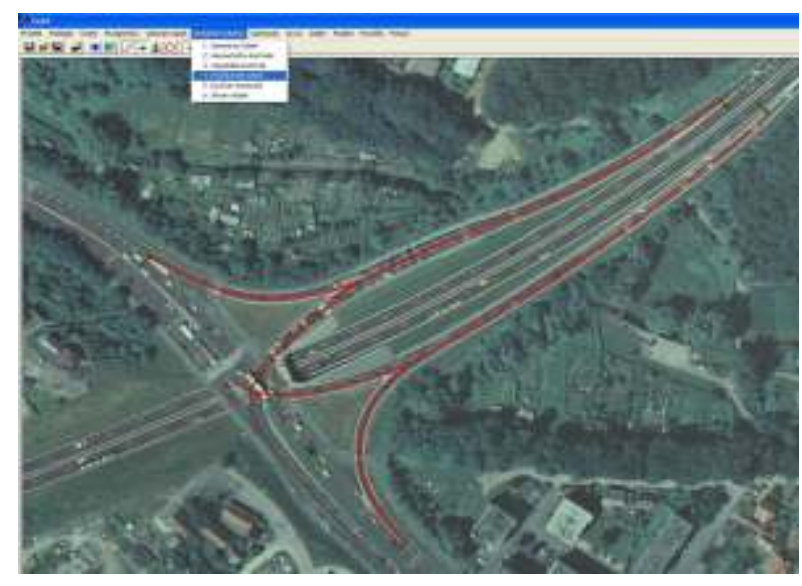
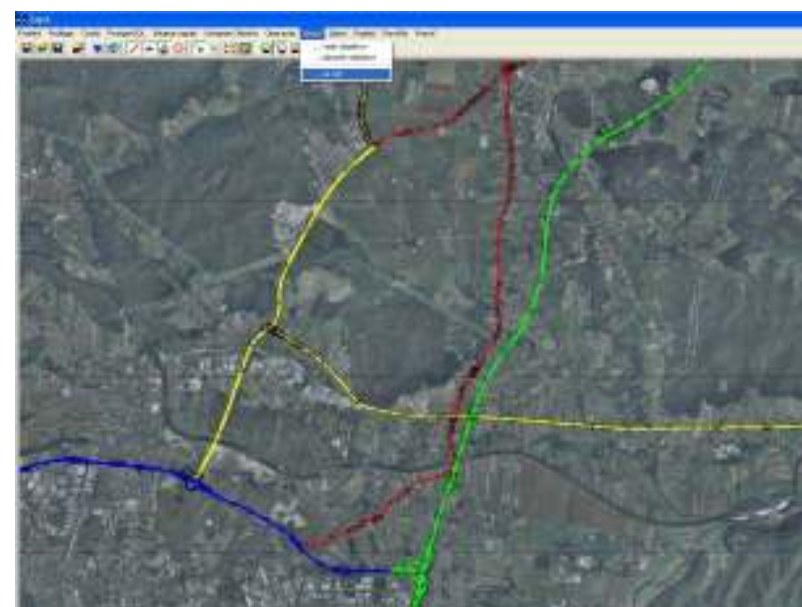
Korak 3:

S procedurami odkrivanja geometričnih in topoloških napak sta najdeni dve napaki na istem GID-objektu. Zgornja napaka je odkrita z algoritmom kotne tolerance, spodnja pa z algoritmom, ki išče viseča vozlišča.



Korak 4:

Geometrične in topološke napake, najdene v prejšnjem koraku, se ročno uredijo. Pri tem mora operater upoštevati pravila za določevanje položaja osi, ki so navedena v drugem poglavju. Po potrebi se uredijo tudi opisni podatki v dialognem oknu.

	<p><u>Korak 5:</u></p> <p>Izbran je celoten odsek, ki je bil v prejšnjem koraku predmet urejanja. To naredi procedura <i>generiraj odsek</i>, izbran pa mora biti vsaj en GID tega odseka. Za pridobitev točnih rezultatov, je priporočljivo izvajanje ukazov, kakor si sledijo v meniju <i>Urejanje odseka</i>. Na koncu se lahko celoten odsek shrani v bazo.</p>
	<p><u>Korak 6:</u></p> <p>Izbrane objekte ali pa celoten sloj državnih cest se lahko izvozi v shape datoteko na način, kot je predviden za vpis podatkov v GJI. To pomeni, da se pri izvozu zapišejo v GJI opisne podatke nekatere privzete vrednosti. Teh podatkov se drugače ne ureja v samem programu. Slika na levi pa prikazuje izris državnih cest po kategorijah.</p>

Slika 26: Tipičen primer uporabe aplikacije (koraki 1-6)

4 ZAKLJUČKI

Podatkovni model digitalne osi cest je za GIS nekoliko nestandarden. Izvira iz opisne baze BCP in je z manjšimi dopolnitvami prenesen na sistem GIS. To povzroča vrsto težav, saj so opisni podatki odvisni od samega položaja osi. Poleg tega predmet urejanja niso posamezni objekti, ampak skupina objektov, ki tvorijo odseke. Zato je bilo potrebno dobro premisliti, na kakšen način bo potekalo urejanje teh podatkov.

Program ZajeC, ki smo ga izdelali na podjetju DFG CONSULTING, d.o.o., je izrazita vertikalna aplikacija, ki je namenjena izključno urejanju in vzdrževanju baze osi državnih cest. V osnovi dela s shape formatom. Podatki o osi ceste in še nekateri drugi podatki pa se nahajajo v odprtokodni PostgreSQL/PostGIS podatkovni bazi. Zasnova programa je prirejena podatkovnemu modelu podatkov v bazi. Ti pa so povsem neodvisni in jih lahko uporabljamo tudi z drugimi GIS-orodji, ki omogočajo dostop do baze.

Aplikacija in baza sta nameščeni lokalno. Uvoz in izvoz ter izmenjava podatkov poteka v obliki shape datotek, kar za potrebe naročnika zadošča. V kolikor bi do podatkov v bazi hoteli dostopati preko spleta, bi morali uvesti druge načine prenosa podatkov (spletni servisi). Danes se za izmenjavo podatkov preko spleta in med različnimi podatkovnimi bazami (Oracle, MySQL,...) veliko uporablja standardizirani jezik XML (Extensible Markup Language) oz. njegove izpeljave, kot je npr. GML (Geography Markup Language). Če bi se pojavila potreba po tovrstnem uvozu/izvozu podatkov, bi v program vgradili še dodatno funkcionalnost, ki bi shape objekte zapisala xml datoteko in obratno.

S takšno namizno aplikacijo, ki za komunikacijo s podatkovno bazo PostgreSQL uporablja programski vmesnik libpq, je preko spleta mogoče dostopati do podatkov v bazi tudi brez dodatnih spletnih servisov. Baza namreč sama deluje kot servis, prenos podatkov pa poteka v prvotni binarni obliki. To sem preiskusil tudi sam, vendar je zaradi slabe povezave pretok podatkov trajal predolgo. S hitrejšo povezavo pa bi bilo to povsem izvedljivo. Pri tem pa se moramo zavedati problema spletne (ne)varnosti. Na usmerjevalniku je namreč potrebno odpreti vrata 5432 na katerih »poslušaja« baza, s čimer postanemo ranljivi pred »napadalci«. Komunikacija med bazo in oddaljeno aplikacijo se sicer lahko vzpostavi po kodirani SSL

(Secure Socket Layer) povezavi. To zagotavlja določeno mero varnosti za samo bazo, ne pa tudi za celotni računalniški sistem, ki se nahaja za usmerjevalnikom.

Nadaljnji razvoj programa je omogočen in je popolnoma prilagodljiv. V kolikor se pojavijo zahteve po dodatni funkcionalnosti, jih lahko enostavno dogradimo in izvedemo novo verzijo programa.

VIRI

Direkcija Republike Slovenije za ceste, 2008.

<http://www.dc.gov.si> (17.5.2008).

ESRI Shapefile Technical Description - White Paper. 1998. Redlands, Environmental Systems Research Institute, Inc.: 28 str.

Kvamme, K., Oštir-Sedej, K., Stančič, Z., Šumrada, R. 1997. Geografski informacijski sistemi. Ljubljana, Znanstvenoraziskovalni center Slovenske akademije znanosti in umetnosti: 476 str.

Metodologija vzdrževanja sloja osi državnih cest s postavitvijo podatkovnega modela. 2004. Ljubljana, Direkcija Republike Slovenije za ceste: 67 str.

PostGIS, 2008.

<http://postgis.refractions.net/> (12.4.2008)

PostgreSQL, 2008.

<http://www.postgresql.org/> (24.4.2008)

pgAdmin III Documentation - Version 2.0. 2007. pgAdmin Development Team

PostgreSQL 8.2.5 Documentation. 2006. The PostgreSQL Global Development Group

PRILOGE

Priloga A: Tabela opisnih podatkov sloja digitalne osi državnih cest

	Št.	Ime polja	T	V	D	Opis atributa
DRSC - atributi	1	ID	N	16	0	ID (DRSC) – auto number, GJI export = ID_UPR
	2	KATEGORIJA	C	2	0	Kategorija ceste: dodeli DRSC po pravilniku
	3	CESTA	C	6	0	Številka ceste: dodeli DRSC o pravilniku
	4	ODSEK	C	6	0	Številka odseka: dodeli DRSC o pravilniku
	5	ODSEK_IME	C	42	0	Opis poteka odseka
	6	ODSEK_TIP	C	1	0	Tip odseka: Dodeljeno v BCP (glej dodatni opis)
	7	DOLZ_ODS	N	7	0	Uradna dolžina odseka (podana v odredbi, BCP, ...)
	8	DOLZ_GRF	N	7	2	Vsota vseh GID-ov, kjer je vrednost atributa pri VRSTA_STAC = U
	9	ZAPST	N	3	0	Zaporedje GID-ov na posameznem odseku
	10	PODOD_IME	C	42	0	Ime več GID-ov, ki tvorijo npr. rondo, krak A, ...
	11	VRSTA_STAC	C	1	0	atribut vpliva na računanje stacionaže
	12	STAC_ZAC	N	7	2	vrednost stacionaže na začetnem N posameznega GID-a
	13	STAC_KON	N	7	2	vrednost stacionaže na končnem N posameznega GID-a
	14	DOLZ_GID	N	7	2	3D dolžina GID-a oz. je 0 če je VRSTA_STAC = N
	15	LENGHT	N	7	2	2D dolžina GID-a
	16	F_RAZTEZKA	N	5	4	količnik razlike med DOLZ_GID/LENGHT oz. je 1,000 če je VRSTA_STAC = N
	17	DAT_ZAC	C	10	0	datum vnosa odseka v DRSC os (podatek iz BCP) (dd/mm/yyyy)
	18	DAT_SPR	C	10	0	datum spremembe posameznega GID-a (grafični in atributni) (dd/mm/yyyy)
	19	DAT_KON	C	10	0	datum, ko GID dobi status B (dd/mm/yyyy)
	20	GID_STAT	C	1	0	Status GID-a: U – v uporabi, B – brisan, P - planiran

	21	OPERATER	C	42	0	ime operaterja, ki je vnesel nov, spreminjal oz. brisal posamezen GID
GJI - atributi	22	ID_GJI	N	10	0	Enolična identifikacijska številka objekta (entitete) v sistemu zbirnega katastra GJI. Atribut je dodeljen s strani GU ob prvem vpisu. Če je bil atribut že posredovan upravljavcu, ga mora le-ta voditi in v primeru spreminjanja ali brisanja objekta (entitete) ta ID tudi uporabiti. ID pri atributnih podatkih mora biti identičen ID-ju lokacijskih podatkov iste entitete!
	23	TIP_SPR	C	1	0	Tip spremembe podatkov
	24	SIF_VRSTE	N	4	0	Vrsta objekta
	25	CC_KLAS	N	5	0	Šifra vrste objekta po CC-SI klasifikaciji Določena na osnovi Uredbe o uvedbi in uporabi enotne klasifikacije vrst objektov in o določitvi objektov državnega pomena (Uradni list RS, št. 33/03) ter Metodoloških pojasnil in navodil za razvrščanje objektov po enotni klasifikaciji vrst objektov (cc-si).
	26	TOPO	N	1	0	Topološka oblika objekta
	27	NAT_YX	N	2	0	Natančnost določitve položaja objekta (y,x koordinate) Izražena s srednjim pogreškom meritve, ki je bila uporabljena za določitev koordinat točke ali lomov linij ali lomov meje poligona.
	28	Z	N	7	2	Absolutna nadmorska višina temena objekta /Samo za točkovne objekte!/ <i>točkovne objekte!</i>
	29	NAT_Z	N	2	0	Natančnost določitve absolutne nadmorske višine objekta V primeru linijskih in poligonskih objektov je to natančnost najslabše določene točke objekta.
	30	GJI	N	1	0	Atribut GJI Določa, ali je objekt GJI ali druga infrastruktura, ki nima statusa GJI, temveč je evidentirana v katastru zaradi interesa lastnika.
	31	VIR	N	2	0	Vir Vir iz katerega je bil pridobljen podatek o lokaciji.
32	DAT_VIR	C	8	0	Datum podatkovnega vira V primeru terenskega zajema je to datum zajema. Datum se zapiše v obliki YYYYMMDD (leto, mesec, dan).	

33	MAT_ST	N	7	0	Matična številka upravljavca objekta Iz Poslovnega registra Slovenije. Če infrastruktura ni GJI, je atribut neobvezen.
34	MAT_GJS	N	7	0	Matična številka izvajalca GJS na objektu Iz Poslovnega registra Slovenije. Če infrastruktura ni GJI, je atribut neobvezen.
35	ID_EL	C	15	0	Identifikacijska številka zadnjega elaborata sprememb podatkov objekta GJI v sistemu zbirnega katastra GJI za objekt GJI
36	DAT_EL	C	8	0	Datum zadnjega vnosa podatkov objekta GJI v zbirni kataster GJI Datum se zapiše v obliki YYYYMMDD.
37	DIM_YX	N	6	2	Zunanja tlorisna dimenzija objekta (v m) za ceste= širina cestišča.
38	DIM_Z	N	6	2	Zunanja vertikalna dimenzija objekta (v m) Pomeni razliko med najvišjo in najnižjo točko objekta. V primeru točkovnih in poligonskih objektov je to višina objekta, v primeru linijskih objektov (npr. vodov) pa je to vertikalni premer cevi, ki je v večini primerov enak kot zunanji premer cevi.
39	OPU	N	2	0	Opuščenost objekta Podatek o objektih GJI, ki po prenehanju delovanja niso bili odstranjeni - tista infrastruktura, ki je nihče ne uporablja in jo je dovoljeno uničiti.
40	ATR1	N	2	0	Specifični atribut 1 Pod tem atributom se za različne vrste objektov vodijo različne karakteristike objektov.
41	ATR2	N	2	0	Specifični atribut 2 Pod tem atributom se za različne vrste objektov vodijo različne karakteristike objektov.
42	ATR3	N	5	0	Specifični atribut 3 Pod tem atributom se za različne vrste objektov vodijo različne karakteristike objektov.
43	ATR4	C	8	0	Specifični atribut 4 Pod tem atributom se za različne vrste objektov GJI vodijo različne karakteristike objektov. = ODSEK

	44	ATR5	C	5	0	Specifični atribut 5 Pod tem atributom se za različne vrste objektov vodijo različne karakteristike objektov.
	45	OPIS	C	30	0	Dodaten opis Po želji upravljavec objektu GJI doda poljubno informacijo, ki v ostalih vpisih ni zajeta. = ODSEK_IME

Podrobnejši opis posameznih atributov:

Atribut št. 6: Tip odseka

ODSEK_TIP	POMEN
O	navadni odsek
A	avtocestni odsek
V	paralelni odsek
P	priključek
D	drugi odseki cest