



Univerzitet u Novom Sadu
Fakultet tehničkih nauka u
Novom Sadu



mr Đorđe Obradović

MODEL RAČUNARSKOG SIMULACIONOG SISTEMA ZA UPRAVLJANJE GEOPROSTOROM U USLOVIMA NEODREĐENOSTI

- doktorska disertacija -

Novi Sad 2011.

SADRŽAJ

Predgovor	i
1 Uvodna razmatranja	1
1.1 Standardi za upravljanje geoprostornim podacima	2
OpenGIS Standardi	3
ISO standardi	4
W3C standardi	5
1.2 Modeli za reprezentaciju prostornih procesa	5
1.3 Modeli prostornih podataka u uslovima neodređenosti	9
Osnovni pojmovi i definicije	13
2 Model proširivog softverskog sistema za upravljanje geoprostorom u uslovima neodređenosti	19
2.1 Model podataka za opis prostorne neodređenosti	19
Linearni fazi prostor – Fazi tačka	23
Uređenje elemenata \mathbb{R}^1 linearnog fazi prostora	30
Fazi duž	33
Fazi trougao	40
Fazi krug	43
2.2 Prostorne operacije	45
Prostorna merenja	45
Prostorne operacije	48
Fazi konveksni omotač	48
Prostorne relacije	64

Modeliranje procesa	70
3 Softverska arhitektura i prototip sistema	79
3.1 Softverska arhitektura sistema	79
Podsistem za skladištenje podataka	80
Prezentacioni podsistem	82
Podsistem za simulaciju procesa	84
3.2 Implementacija prototipa	86
Biblioteka osnovnih tipova i funkcija	88
Biblioteke klasa za modeliranje procesa	94
Podsistem za skladištenje podataka	97
FGeoIS i FGeoISProcess podsistemi	103
GMLToJava i GMLToCSharp podsistem	119
Primer korištenja	121
AutoCad proširenje	125
Opis korišćenja implementiranih podsistema	136
4 Verifikacija modela	139
4.1 Detekcija fazi linija	139
4.2 Detekcija topografskih simbola	143
Model podataka	147
Arhitektura implementiranog sistema	148
Rezultat	155
5 Zaključak	161
6 Literatura	165

Predgovor

Više od 80% podataka sa kojima čovek operiše predstavlja prostorno referencirane podatke. Gotovo da nema delatnosti u kojoj se ne koriste prostorni i prostorno vremenski podaci. Oblasti su mnogobrojne i obuhvataju proizvodnju, trgovinu, zdravstvo, životnu sredinu, itd.

Danas se za upravljanje prostornim resursima koristi čitav niz metoda i tehnika integriranih u složene sisteme čija je osnovna funkcionalnost uslovljena vremenski ažurnim i obuhvatnim geoprostornim podacima, odnosno prikupljanjem, obradom, skladištenjem, prezentacijom, distribucijom i razmenom raznovrsnih geoprostorno referenciranih podataka. Jedna od karakteristika ovakvih sistema je prisustvo nepreciznosti i neodređenosti. Pri tome se nepreciznost i neodređenost pojavljuju i u podacima i u procesima sistema.

Karakteristika informacionih sistema, pa i računarskih simulacionih sistema je kontinuirani razvoj i stalno unapređenje novim modelima, metodama i tehnikama. Posledica ove karakteristike je potreba za njihovom modularnom arhitekturom koja dozvoljava jednostavnu integraciju novih komponenti u sistem.

Predmet istraživanja ove doktorske disertacije je razvoj i implementacija modela proširivog računarskog simulacionog sistema za upravljanje geoprostorom u uslovima neodređenosti koji omogućava integraciju različitih postojećih sistema kao i razvoj novih algoritama za obradu, novih modela za reprezentaciju procesa i njihovu međusobnu integraciju sa ciljem unapređenja kvaliteta i efikasnosti u poslovima koji se odnose na upravljanje geoprostornim resursima u realnim uslovima. Jedan od zahteva koje nameću sistemi za upravljanje geoprostornim resursima u realnim uslovima je i postojanje razlike između stvarnih karakteristika (dimenzije, položaj i sl.) realnog objekta i njegove reprezentacije u informacionom sistemu.

U okviru istraživanja u ovoj disertaciji identifikovane su dve osnovne grupe modela u oblasti geoinformacionih sistema. Prva grupa obuhvata načine reprezentacije geoprostornih podataka. Druga grupa modela obuhvata načine korišćenja ovih reprezentacije u procesima prikupljanja i obrade podataka, i rasuđivanja na osnovu njih.

Osnovni doprinos ove disertacije je formalna specifikacija okruženja u kojem su obe grupe modela proširene podrškom za reprezentaciju nepreciznih i neodređenih geoprostornih podataka i procesa.

Za modeliranje neodređenosti i nepreciznosti korišćen je fazi pristup.

Softverski simulacioni sistem modeliran je kao višeslojna arhitektura a za modeliranje je korišćen UML 2.0. Implementacija je izvršena kombinacijom više objektno orijentisanih programskih jezika. Softverska arhitektura sistema bazirana je na Internet tehnologijama i važećim standardima u oblastima softverskog inženjerstva i upravljanja geoprostornim podacima i uz oslonac na savremene alate za upravljanje prostornim podacima.

Tekst disertacije sastoji se iz pet poglavlja.

U prvom poglavlju data je analiza radova i istraživanja u predmetnoj oblasti. Na početku je dat detaljan prikaz modela i standarda za prikupljanje i reprezentaciju geoprostornih podataka sa posebnim osvrtom na izvore i načine modeliranja neodređenosti i nepreciznosti. Na kraju prvog poglavlja dat je i pregled osnovnih pojmova i formula koje se koriste u narednim poglavljima.

Drugo i treće poglavlje predstavljaju centralni deo rada i sadrže glavni originalni naučni doprinos.

Drugo poglavlje sadrži formalnu specifikaciju modela proširivog simulacionog sistema za upravljanje geoprostorom u uslovima neodređenosti koji obuhvata:

- (i) modeliranje podataka i
- (ii) modeliranje procesa (servisa).

U trećem poglavlju opisana je softverska arhitektura proširivog simulacionog sistema i implementacija prototipa simulacionog softverskog sistema. Implementirani sistem predstavlja okruženje sastavljeno od grupe alata i podsistema za formalno modeliranje nepotpunih i nepreciznih geoprostornih podataka i geoprostornih servisa nad njima. Geoprostorni servisi odnose se na procese prikupljanja, analize, obrade i korišćenja nepotpunih georeferenciranih podataka.

Četvrto poglavlje sadrži specifikaciju sistema kojim je izvršena verifikacija predloženog modela na primeru izdvajanja geoprostornih podataka iz rasterskih topografskih karata, modeliranje procesa za njihovu obradu i na kraju skladištenje u sistemu za upravljanje geoprostornim podacima.

Na kraju, u petom poglavlju analizirani su rezultati i doprinosi doktorske disertacije, data su zaključna razmatranja kao i predlozi pravaca daljih istraživanja.

Bibliografija sadrži bibliografske jedinice koje su direktno pomenute u tekstu disertacije. U disertaciji je usvojen IEEE stil citiranja referenci.

Zahvaljujem se svim članovima Komisije koji su svojim korisnim sugestijama doprineli da ona bude jasnija i preglednija.

Zahvaljujem se i prof. dr. Nebojši Raleviću koji je strpljivo pratio ovaj rad i sugestijama pomogao da se unapredi kvalitet i korektnost matematičkih dokaza.

Takođe, želim da se zahvalim prof. dr. Dušanu Surli i prof. dr. Dušanu Petrovačkom koji već više od petnaest godina prate moj rad a koji su meni i mnogima pomogli da iskreno zavolimo profesiju kojom se bavimo.

Posebnu zahvalnost dugujem mentoru prof. dr. Zori Konjović i prof. dr. Endreu Papu za nesebičnu podršku u toku izrade disertacije.

Takođe se zahvaljujem supruzi, sinu, ćerci i roditeljima na razumevanju i podršci.

Novi Sad, 2011.

Đorđe Obradović

Poglavlje 1

Uvodna razmatranja

U različitim oblastima nauke postoji potreba za upravljanjem geometrijskim, geografskim i prostornim podacima, odnosno podacima koji su u vezi sa nekim prostorom. Prostor u ovom slučaju može da bude 2D ili 3D apstrakcija površine zemlje, medicinskog snimka, unutrašnjosti zgrade, raspored lanaca molekula proteina i slično.

Najčešće se pod sistemima za upravljanje prostornim podacima podrazumeva GIS (Geo informacioni sistemi) za koji je razvijen čitav niz standarda koji definišu načine modeliranja, razmenu podataka kao i servise koje takvi sistemi mogu da pružaju. Objekti od interesa pored prostornih osobina uvek imaju i *obične* alfanumeričke osobine pa se zato ovi sistemi razvijaju kao proširenje relacionog sistema za upravljanje podacima.

Güting je u radu [1] dao minimalan skup osobina koje jedan sistem mora da poseduje da bi se okarakterisao kao sistem za upravljanje prostornim podacima ili kraće prostorna baza:

- a) prostorna baza je baza podataka,
- b) poseduje prostorne tipove podataka u modelu podataka i u upitnom jeziku,
- c) obezbeđuje indeksiranje prostornih podataka i efikasne algoritme za prostorno spajanje (*spatial join*),

Osobina pod a) je trivijalna i odražava potrebu da se, pored prostornih geometrijskih podataka, koriste i “ne prostorni” (alfanumerički) podaci. Sistem koji je u stanju da vodi računa samo o prostornim podacima ne bi bio praktično upotrebljiv bez podrške za modeliranje i pretraživanje alfanumeričkih podataka. (b) Prostorni tipovi podataka su *tačka*, *linija* i *region*. Ovim tipovima se omogućuje apstrakcija za modeliranje strukture geometrijskih objekata u prostoru, njihovih međusobnih relacija, osobina i operacija. (c) Sistem mora da poseduje mehanizme za efikasno pretraživanje velike kolekcije objekata koji se, na primer, nalaze u određenoj oblasti. Za ovo je neophodno posedovanje komponente sistema za prostorno indeksiranje. Takođe neophodno je obezbediti podršku za povezivanje objekata različitih klasa (različitih tabela) preko *prostornih veza*.

Konzorcijum OGC (Open Geospatial Consortium) koji se bavi standardizacijom u oblasti geoprostornih sistema za upravljanje podacima

dao je veoma sličnu definiciju sistema za upravljanje prostornim podacima [2]:

Prostorna baza podataka je baza podataka koje je optimizovana za čuvanje i pretraživanje podataka koji su u nekoj vezi sa prostorom uključujući tačke, linije i poligone.

Između ostalog, OGC konzorcijum izdvojio je osnovne grupe prostornih operacija koje jedna prostorna baza treba da obezbedi:

- a) prostorna merenja (rastojanja između tačaka, linija i regiona),
- b) prostorne funkcije (transformacija postojećih prostornih podataka, određivanje zajedničkih osobina),
- c) prostorni predikati (logički predikati koji se koriste u pretragama),
- d) konstruktori i
- e) funkcije za *nadgledanje* (*observers*).

U knjizi [3] istaknuto je da postoji ekspanizija u istraživanju prostornih baza podataka, kao i da je velik broj tehničkih problema računarske prirode već rešen. Naravno, postoji i mnogo nerešenih problema u ovoj oblasti pa istraživači pažnju usmeravaju u nekoliko pravaca: kvalitet prostornih podataka i neodređenost, korisnički interfejsi i ekonomičnost prostornih baza. Tako je otvoreno nekoliko interesantnih zadataka: Kako obezbediti podršku za procese pretraživanja, skladištenja i pretraživanja prostornih podataka gde su izvori velike mreže fiksnih ili mobilnih senzora za prikupljanje prostornih podataka u realnom vremenu? Kako prevazići problem u neodređenosti ulaznih podataka? Kako rešiti nedostatak interoperabilnosti između sistema? Kako obezbediti korisne servise na ograničenim, često i po procesnoj moći limitiranim platformama poput mobilnih telefona? Kako prilagoditi i implementirati tehnike računarske inteligencije u ovakve sisteme?

U ovom radu izvršena je specifikacija proširive softverske arhitekture u kojoj bi se integrisali podaci iz raznih izvora, postojeći sistemi proširili novim servisima za simulaciju prostornih procesa, obezbedilo njihovo modeliranje i upravljanje u uslovima neodređenosti i nepreciznosti.

Standardi za upravljanje geoprostornim podacima

Infrastruktura prostornih podataka (Spatial Data Infrastructure - SDI) je pojam koji označava okruženje u kojem više različitih sistema učestvuje a objedinjuje prostorne podatke, metapodatke, korisnike i alate. Zbog potrebe da integrišu prostorne podatke na globalnom nivou SDI inicijative su obično organizovane na nacionalnom nivou od državnih institucija kao što

je, na primer, NSDI (National Spatial Data Infrastructure) u SAD, ili čak i na transnacionalnom nivou kao što su INSPIRE u Evropskoj Uniji ili UNSDI za Ujedinjene Nacije.

Osnovne programske komponente koje čine SDI su:

- 1) Klijentski programi koji omogućuju prezentaciju, pretraživanje i analizu prostornih podataka;
- 2) Katalog servisa koji obezbeđuje pretraživanje metapodataka o prostornim servisima, prostornim podacima i prostornim resursima;
- 3) Servisi prostornih podataka koji omogućuju razmenu prostornih podataka preko Interneta;
- 4) Servisi za obradu koji obuhvataju servise koji se koriste za razne transformacije prostornih podataka;
- 5) Prostorne baze koje se koriste za skladištenje, pretraživanje i obradu prostornih podataka.

Kao podrška za interakciju koriste se OpenGIS standardi (WMS, WFS i GML), ISO standardi(ISO 19115) i standardi W3C konzorcijuma (SVG, X3D).

Konzorcijum OGC ili OpenGIS osnovan je 1994. godine i čine ga preko 200 članova (kompanije, državne agencije i akademske institucije). Između ostalih, odbor direktora konzorcijuma čine članovi iz sledećih kompanija i institucija: Google, Oracle, IBM, Autodesk, Microsoft, ESRI, Columbia Univerzitet CIESIN.

OpenGIS Standardi

Open GIS standari obuhvataju više od 50 standarda, od koji će ovde biti opisani samo najvažniji standardi za implementaciju simulacionog modela koji je predmet ovoga rada. To su sledeći standardi.

WPS (Web Processing Service) standard opisuje način na koji korisnik može pokrenuti geoprostorni proces (ulaz izlaz i kako se pokreće izvršavanje procesa).

WCS (Web Coverage Service) standard kojim se omogućuje interoperabilnost *coverage* podataka poput satelitskih snimaka, digitalnih aero snimaka i drugih podataka koji se dobijaju merenjam a rezultat je rasterski snimak.

WMS (Web Map Service) obezbeđuje http interfejs prema georeferenciranim slikama odnosno mapama.

WFS (Web Feature Service) standardom je specificiran način za postavljanje upita i preuzimanje geoprostornih podataka. Ovim standardom je takođe omogućena manipulacija skupom geoprostornih podataka.

WCPS (Web Coverage Processing Service) opisuje jezik za ekstrakciju, obradu i analizu više dimenzionih coverage objekata (rasterski podaci dobijeni od senzora, satelitski snimci i statistički podaci).

TML (Transducer Markup Language) je protokol za razmenu podataka između aplikativnog i prezentacionog sloja gde se razmenjuje velika količina podataka u obliku tokova podataka u realnom vremenu.

SensorML (Sensor Model Language) jezik za opis modela senzora koji obuhvataju geometriju, dinamiku i fizičke karakteristike senzora. Senzori mogu da budu jednostavni vizualni termometri, kompleksni elektronski mikroskopi ili sateliti za snimanje površine zemlje.

GML (Geography Markup Language) je XML jezik za opis geografskih objekata. GML kao i svaki drugi XML bazirani jezik sadrži dva dela *shemu* i *podatke*. U praksi GML dopušta kreiranje specijalizovanih gramatika preko kojih se detaljnije opisuje specifična vrsta geografskih podataka (geoprostorni podaci u poljoprivredi, saobraćaju, državnoj administraciji) a izvode se jednostavnim proširenjem GML jezika. Jedan od primera je CityGML standard.

KML – je XML jezik fokusiran na prezentaciju geografskih podataka kao i anotaciju mapa i slika. Takođe, koristi se i za korisničku navigaciju.

ISO standardi

Unutar ISO organizacije Tehnički komitet TC211 zadužen je za razvoj standarda koji se odnose na digitalne geoprostorne podatke. U nadležnosti TC211 je trenutno 49 standarda od kojih se izdvajaju:

ISO 6709 standard za reprezentaciju pozicije tačkastih geografskih objekata preko njegovih koordinata

ISO 19101 Standard za opis geografskih informacija odnosno referentnog modela za integraciju svih standarda iz grupe 19100. U ovom standardu definisan je osnovni element *Feature* geoprostornih sistema i njegova definicije je: *Feature* je abstrakcija objekta iz realnog sveta čiji su geografske osobine povezane sa lokacijom relativnom u odnosu na planetu zemlju. ISO 19113 Standardi koji opisuju principe o kvalitetu geoprostornih podataka. Standard ISO 19128 opisuje WMS standard definisan OGC

konzorcijumom. ISO 19136 GML jezik opisan u Open GIS standardima. ISO 19138 Standard koji se bavi merenjem kvaliteta geoprostornih podataka. Standardi ISO 19132 -19134 odnose se na lokacijski određene servise (*location based services*)

W3C standardi

Konzorcijum W3C takođe je definisao određene standarde iz oblasti upravljanja geoprostornim podacima. Jedna od najčešćih specifikacija koje se odnose i na prostorne i geometrijske objekte je SVG (Scalable Vector Graphics) format za reprezentaciju 2D vektorskih podataka. Drugi često korišten standard je X3D koji predstavlja XML ekvivalent VRML formata kojim je omogućeno upravljanje georeferenciranim 3D podacima.

U trenutku pisanja ove disertacije nisu postojali zvanični standardi za upravljanje nepreciznim geoprostornim podacima (standardi koje je propisala neka od prethodno navedenih organizacija). Međutim, u radu [4] Moriss i Petry dali su predlog proširenja GML standarda podrškom za modeliranje nepreciznih prostornih podataka. U njihovom radu neprecizni prostorni podatak modelira se fazi skupom koji je predstavljen kao kolekcija konačno mnogo alfa rezova (precizni geometrijski objekti). Autori su takođe istakli da je za ovako izabranu reprezentaciju potrebno razviti poseban ili proširiti neki od postojećih upitnih jezika kojima bi se obezbedile neke od osnovnih prostornih operacija. OGC konzorcijum doneo je u dokumentu [5] nacrt predloga (koji još nije usvojen) standarda UnCrtML za reprezentaciju nepreciznih prostornih podataka. U njihovom predlogu koristi se probabilistički pristup a posebna pažnja je posvećena analizi kvaliteta merenih podataka korišćenjem statističkih metoda.

Modeli za reprezentaciju prostornih procesa

Pored prostornih osobina kojima se modelira stanje objekta, veoma je važno poznavanje zakonitosti promene stanja (dinamike sistema). Utvrđivanje zakonitosti ponašanja sistema jedna je od najčešćih tema istraživanja u većini naučnih disciplina. Poznavanjem zakonitosti promene stanja moguće je predvideti ponašanje sistema ili uticati na njegovo ponašanje tako da se dobiju željeni rezultati ili stanja. Primeri koji opravdavaju potrebu modeliranja zakonitosti ponašanja sistema nalaze se u medicini,

tehnologiji, računarstvu, mašinstvu, poljoprivredi, elektrotehnici, sportu, društvenim naukama i drugima. Zato postoji veoma veliki broj različitih definicija, odnosno opisa značenja pojma proces. Tako se pod pojmom proces u inženjerskim disciplinama posmatra skup transformacija stanja ulaznih elemenata u cilju proizvodnje željenih elemenata. Termodinamički procesi predstavljaju promenu energetske odnosno termodinamičke osobine sistema. U proizvodnji hrane proces predstavlja transformaciju osnovnih sastojaka u finalni proizvod odnosno hranu. U medicini proces predstavlja transformaciju bioloških i hemijskih osobina živih organizama.

Zajednička karakteristika svih definicija je da proces predstavlja niz promena stanja nekog sistema.

Karakter ovih promena može biti takav da se sistem pod istim uslovima *uvek* ponaša na isti način. Takve procese nazivamo *determinističkim* procesima. Ako se promene stanja razlikuju za iste uslove tada takve procese nazivamo *stohastičkim* procesima.

Tri grube klasifikacije procesa koji će se u ovom radu razmatrati su: vremenski procesi, prostorni procesi i prostorno vremenski procesi. Za ovaj rad od interesa su prostorno vremenski procesi.

Vremenski procesi su procesi promenljivi u vremenu. Kod prostornih procesa promene stanja sistema povezane su sa promenama prostornih karakteristika. Na primer, proces određivanja lokacije za izgradnju obdaništa ako se znaju lokacije stambenih zgrada, raspodela stanovništva ili proces planiranja izgradnje puteva u odnosu na konfiguraciju terena. Vremensko prostorni procesi prate promene sistema u vremenu i prostoru. Primeri vremensko prostornih procesa su širenje epidemije, šumski požari ili upravljanje saobraćajem.

Modeliranje procesa predstavlja osnovni zadatak prilikom utvrđivanja zakonitosti ponašanja sistema.

Analizom radova koji se bave modeliranjem procesa izdvajaju se tri grupe načina za modeliranje:

- (i) parcijalnim diferencijalnim jednačinama
- (ii) automatima prelaza stanja i celularnim automatima
- (iii) opšte namenskim jezicima za modelovanje

Prva grupa radova [6-9] odnosi se na matematičke modele ponašanja sistema u kojima se ponašanje opisuje parcijalnim diferencijalnim jednačinama. Matematički modeli se prema autorima klasifikuju u odnosu na nekoliko osobina: linearni i nelinearni, deterministički i stohastički, statički i dinamički i na kraju diskretni i kontinualni sistemi. Takođe,

istaknuto je da je prilikom modelovanja potrebno vršiti evaluaciju modela. Najjednostavniji način je poređenje izmerenih stvarnih vrednosti i vrednosti dobijenih modelom. Rastojanje ili razlika merenih i izračunatih vrednosti često je posebna tema istraživanja (radovi [10-19]) a odnosi se na modeliranje metričkih prostora i osobina elemenata koji ih čine. Cilj evaluacije modela je da se razlika ili rastojanje između stvarnih ili izmerenih vrednosti i izračunatih vrednosti postanu minimalne. Modeli uređenja razmatrani su u radovima [20-26], gde su posmatrani različiti aspekti uređenja elemenata koji po prirodi ne moraju biti klasični realni brojevi već kompleksne matematičke strukture poput fazi skupova, redova, funkcija i relacija.

Druga grupa radova odnosi se na istraživanja u oblasti konačnih automata i celularnih automata. Jednu od najranijih formalnih definicija za opis stohastičkih procesa postavio je ruski matematičar Andrej Markov 1906. godine u radu [27]. Markovljevi procesi (ime su dobili po autoru) predstavljaju automat prelaza stanja u kojem su prelazi modelirani kao slučajne promenljive. Grupi ranih istraživanja u oblasti modeliranja procesa primenom automata prelaza stanja pripada formalna definicija i praktična primena koju je postavio Alan Turing u radu [28] 1937. godine. U radu je opisana takozvana *Turingova mašina* ili "*automatska mašina*" (a-mašina) za modeliranje determinističkih procesa, koja predstavlja osnovu za razvoj računara kao mašina za izvršavanje računskih procesa. Von Nojman je u nekoliko radova objavljenih u periodu (1948. - 1953.) (preuzeto iz [29]) predstavio osnovne osobine Teorije kompleksnih automata. U radu [30] Von Nojman je postavio prve definicije celularnih automata. Ova specifična vrsta automata pogodna je za istraživanje prostorno vremenskih procesa jer na jednostavan način modelira uticaje susednih objekata. Jedan od najpoznatijih primera celularnih automata prikazan je 1970. godine pod nazivom *Igra života (Game of Life)*, u kojoj je John Conway na primeru u kojem se ćelije mogu naći u jednom od dva moguća stanja u dvodimenzionalnom prostoru primenom jednostavnih pravila opisao vrlo kompleksan prostorni proces.

U trećoj grupi nalaze se istraživanja u kojima se proces predstavlja kao jedan scenario izvršavanja računskog programa. Računski programi mogu da budu predstavljeni proceduralno, objektno ili korišćenjem agentski orijentisanih metodologija. U specifičnim primenama kao što su poslovni procesi razvijeni su jezici za modelovanje, odnosno izvršenje poslovnih procesa kao što su, na primer, BPML i BPEL. Specifikacija jezika je rezultat standardizacije u oblasti modeliranja, simulacije i upravljanja poslovnih procesa. U oblasti modeliranja i standardizacije prostornih

procesa u radovima [31-35] istaknuto je da, pored razvoja modela za predstavljanje geoprostornih procesa, postoji potreba da se pojedini algoritmi u obliku servisa učine dostupnim širem krugu korisnika putem standardizovanih interfejsa.

Open GIS konzorcijum za standardizaciju razvio je standard WPS. Ovim standardom predviđeno je da se procesi u obliku servisa izvršavaju kao web servisi, a da za opis, prenos parametara i rezultata koriste posebno definisan *xml* format.

Na slici 1.1. prikazan je dijagram komponenti WPS servera, koji sadrži proizvoljan broj procesa. Procesima se pristupa preko tri standardna interfejsa. Prvi interfejs *GetCapabilities* služi za preuzimanje liste raspoloživih servisa, interfejs *DescribeProcess* služi za opis pojedinačnog servisa i na kraju interfejs *Execute* služi za pokretanje servisa. U nastavku su dati primeri korišćenja prva dva interfejsa.

Primeri:

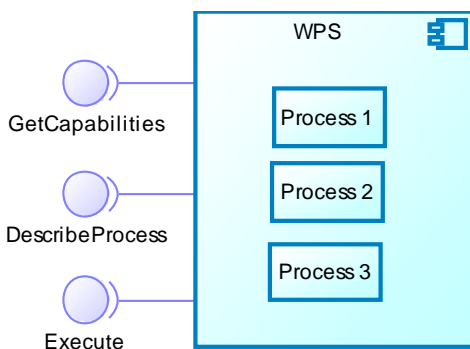
GetCapabilities

<http://deegree3-demo.deegree.org/deegree-wps-demo/services?service=WPS&request=GetCapabilities>

DescribeProcess

http://deegree3-demo.deegree.org/deegree-wps-demo/services?service=WPS&version=1.0.0&request=DescribeProcesses&identifier=st_joinadjacentlines

Prva verzija 1.0.0 WPS standarda objavljena je sredinom 2007. godine. Nekoliko implementacija servera koji podržavaju WPS standard implementirani su kao web aplikacije u kojima se servisi definišu kreiranjem *xml* datoteke koja opisuje servis i *class* datoteke u kojoj je servis opisan.



Slika 1.1 Dijagram komponenti WPS servera

Modeli prostornih podataka u uslovima neodređenosti

Značajni problemi sa kojima se suočavamo prilikom modeliranja prostornih podataka su problem *jasnih granica* i problem ograničenja koje nameće računarska podrška ili okruženje.

Prvi problem odražava slučaj kada su prostorne osobine objekata takve da je njihove granice nemoguće predstaviti preciznim geometrijskim oblicima (npr. granice reke ili granice planine).

Drugi problem je kompleksniji i moguće ga je podeliti u nekoliko kategorija:

- Nepreciznost u ulaznim podacima.
 - Uređaji za merenje pozicije rade sa određenom tačnošću.
 - Prilikom digitalizacije izmerenih podataka (za potrebe transporta, skladištenja i obrade) vrši se aproksimacija ovih podataka.
- Modelovanje realnih objekata pomoću geometrijskih objekata. (objekat koji ima pravougli oblik zamenjujemo pravougaonikom).
- Nepreciznost u algoritmima i prostornim procesima - kontinualno modelovanje i diskontinualna implementacija
 - aritmetika sa pokretnim zarezom,
 - korišćenje iterativnih postupaka u približnom računanju,
 - računarska geometrija.

Löffler je u radu [36] izdvojio dva aspekta nepreciznosti u podacima: *određenost* i *preciznost*. *Određenost* predstavlja meru korektnosti neodređenog podatka, odnosno u kojoj meri podatak sa kojim se radi odstupa od tačne vrednosti. Sa druge strane *preciznost* je osobina nepreciznog podatka koja ne zavisi od tačnosti podatka. Primer za ovu vrstu nepreciznosti je slučaj u kojem prostorni podatak predstavljamo mrežom mogućih vrednosti. Što je mreža gušća preciznost je veća i obrnuto. *Određenost* i *preciznost* su po autoru dva odvojena koncepta, tako da podatak može biti precizan ali neodređen ili određen ali neprecizan. Precizan a neodređen podatak ne sadrži korisnu informaciju i on najčešće nije od interesa za odlučivanje.

Analizom radova koji se bave modeliranjem prostornih podataka uočeno je da postoje tri pristupa modeliranju neodređenosti prostornih podataka. U prvu grupu [37-40] spadaju *tačni* modeli, druga grupa [41], [42] su probabilistički modeli, a treća grupa su *fazi* modeli [10], [43-46].

Tačni modeli zasnivaju se na modeliranju nepreciznih prostornih podataka korištenjem klasičnih preciznih geometrijskih objekata. Ovaj pristup se

najčešće koristi. Tako na primer, prostorno proširenje za Oracle sistem za rukovanje podacima (Spatial Oracle) kao dodatni parametar prilikom poziva prostornih funkcija koristi *toleranciju*. Na ovaj način je vrlo jednostavno rešen problem nepreciznosti podataka. Rastojanje između dva objekta koji se nalaze na različitim pozicijama može biti i 0 u zavisnosti od korištene tolerancije. Autori koji su problem nepreciznosti modelirali *tačnim modelima* koristili su relativno jednostavne geometrijske oblike (krugove, pravougaonike i linije) za modeliranje a onda su nad tako definisanim modelima razvijali algoritme za njihovu obradu. Guibas, Selasin i Stolfi u radu [37] nazvali su grupu predikata i funkcija za obradu nepreciznih prostornih objekata "Epsilon geometrija" u kojoj su formalno opisani mehanizmi koji se komercijalno koriste u Spatial Oracle sistemu za upravljanje prostornim podacima. U ovom radu opisani su osnovni prostorni predikati *preklapanje*, *kolinearnost*, *između dve tačke* i *unutar trougla* u uslovima kad se nepreciznost modelira krugovima. Naime, prostorni predikat *preklapanje* (*coincidence*) između dva objekta će biti tačan ako dve kružnice sa centrom na poziciji objekta i poluprečnikom ϵ imaju zajedničkih tačaka i tada kažemo da se dva objekta " *ϵ preklapaju*". Dobra osobina ovog pristupa je što se nepreciznost modelira relativno jednostavnim formalnim matematičkim aparatom uz minimalno korišćenje računarskih resursa. Međutim, loša osobina proizilazi iz činjenice da se koristi binarna logika u kojoj je izraz ili tačan ili netačan i činjenica da se nepreciznost svih objekata modelira istim parametrom bez obzira na tačnost samog podatka (neki podaci mogu biti precizniji ali će se prilikom analize posmatrati na isti način kao i podaci koji su lošeg kvaliteta).

Jedan od najranijih zabeleženih probablističkih modela za modeliranje geometrijske nepreciznosti predložio je Carl Friedrich Gauss (1777-1855). On je u knjizi *Teorija kretanja nebeskih tela* [47] opisao način za određivanje pozicije nebeskog tela na osnovu konačnog broja nepreciznih merenja, poznat kao *metoda najmanjih kvadrata*. On je dokazao da je velika verovatnoća da aproksimativna kriva odgovara realnoj krivoj u slučaju da su odstupanja ili greške tačaka od aproksimativne krive distribuirana u skladu sa normalnom raspodelom. Verovatnoća i određenost raste sa brojem tačaka. U opštem slučaju meru nepreciznosti moguće je izmeriti uzastopnim merenjem i korišćenjem statističkih metoda. Ovaj pristup pogodan je za slučaj kada postoje detaljne informacije o raspodeli grešaka, kad je moguće izvršiti velik broj merenja i kad računsko vreme nije ograničeno.

Moderan pristup za modeliranje nepreciznosti predstavljen je 1965. kada je Zadeh u radu [48] postavio temelje teorije koja podržava koncept parcijalne

pripadnosti skupu, odnosno prvi je put pomenut termin *Fazi skup*. Korištenjem fazi skupova omogućeno je modeliranje prostornih regiona sa nejasnim ili fazi granicama. Teorija i osobine fazi skupova od tada postali su predmet intenzivnog istraživanja sa ciljem praktične primene u modelima prostornih geometrijskih objekata i operacija odnosno relacija. Schneider je na primer u radu [43] analizirao problem modeliranja prostornih objekata sa nejasnim granicama. U ovom radu predstavljen je konceptualni model fazi prostornih objekata zajedno sa fazi geometrijskom unijom, presekom i razlikom. Takođe, predstavljeni su osnovni topološki predikati. Osnova za modeliranje prostora u ovom radu nije uobičajeni Euklidski prostor ili kontinualni prostor već diskretni geometrijski domen *rešetkaste podele (grid partition)* u kojem se koristi mašinski skup realnih brojeva (skupovi brojeva koje je moguće koristiti u računanjima u kojima učestvuje računar). U radu [44] isti autor predstavio je apstraktni konceptualni model fazi prostornih tipova podataka kao što su fazi tačke, fazi linije i fazi regioni. U ovom radu kao formalno sredstvo za konceptualni model iskorištena je teorija fazi skupova i fazi topoloških predikata.

Buckley i Eslami u radu [45] postavili su osnove za takozvanu fazi geometriju definišući fazi tačku kao fazi skup čija funkcija pripadnosti je konveksna (svi alfa preseki su konveksni skupovi), prekidna sa gornje strane (ako postoji tačka prekida vrednost funkcije odgovara većoj od dve moguće vrednosti) i sa jezgrom koje čini samo jedna tačka.

Većina prostornih funkcija, predikata i operacija nad ovako modeliranim geometrijskim objektima kao osnovu ima merenje rastojanja i uređenje u skupu nepreciznih rastojanja.

Jednu od najranijih definicija rastojanja između fazi skupova predstavili su Dubois i Prade u [49], Rosenfeld [50], [51] i drugi [11], [52]. Koncept *fazi metričkog prostora* predstavljen je u više radova. U opštem slučaju, izdvojile su se dve grupe pristupa. Prvi [11], [52], [53] tretira fazi metriku kao realnu funkciju (rastojanje između fazi skupova je realan broj). U drugom pristupu, koji se koristi i u ovoj disertaciji, rastojanje između fazi skupova se definiše kao fazi skup [10], [12-14], [20], [49-51], [54] ili kao fazi broj [15-17], [45]. Proces uređenja fazi rastojanja kao jedan od fundamentalnih procesa u teoriji odlučivanja razmatran je u više radova [18], [21-24], [49], [55-57].

Hassanzadeh i Rappaport u radu [38] istakli su da je problem određivanja konveksnog omotača skupa tačaka jedan od fundamentalnih problema u oblasti modelovanja prostornih podataka i predmet intenzivnih istraživanja,

ali da problem određivanja konveksnog omotača skupa nepreciznih tačaka još uvek nije u dovoljnoj meri istražen. U istom radu dali su predlog nekoliko modela za opis neodređenosti a nad tako definisanim modelima su predložili i algoritam za određivanje konveksnog omotača. Predloženi su modeli gde se umesto precizne tačke koristi duž ili pravougaonik. Stvarna pozicija tačke je onda *negde* na duži ili na pravougaoniku, a problem određivanja konveksnog omotača u tom slučaju nije jednoznačan, već se računa donja i gornja granica.

Osnovni pojmovi i definicije

U nastavku će biti dat sažet pregled pojmova i definicija koji se koriste kao osnova u ovoj doktorskoj disertaciji, kao i korišćene oznake.

Univerzalni skup – \mathcal{X} je neprazan skup koji sadrži sve moguće elemente od značaja u datom kontekstu ili određenom problemu.

Klasični skup - kolekcija objekata ili elemenata iz univerzalnog skupa koje karakteriše neka zajednička osobina.

Pripadnost skupu- element pripada nekom skupu ako poseduje osobinu koja karakteriše skup, u protivnom element ne pripada skupu.

Obeležavanje skupova:

(i) $A = \{1, 2, 3\}$ pogodno za obeležavanje skupa sa konačno mnogo elemenata

(ii) $A = \{x \mid \mathcal{A}(x)\}$ gde je $\mathcal{A}(x)$ uslov koji zadovoljavaju svi članovi skupa A npr. $\mathcal{A}(x) \equiv 0 < x \leq 17$. Pogodno za skupove sa beskonačno mnogo članova

(iii) $\mu_A: \mathcal{X} \rightarrow \{0,1\}$ funkcija pripadnosti ili karakteristična funkcija skupa npr. $x \in \mathcal{X}$ $\mu_A(x) = \begin{cases} 1 & \text{ako je } 0 < x \leq 17 \\ 0 & \text{u protivnom} \end{cases}$ karakteristična funkcija univerzalnog skupa je $\forall x \in \mathcal{X}, \mu_{\mathcal{X}}(x) = 1$.

Prebrojiv skup nazivamo *diskretan skup* a skup sa beskonačno mnogo elemenata nazivamo *kontinualan skup*.

Prazan skup je skup koji ne sadrži ni jedan element a obeležavaćemo ga sa \emptyset ili $\{\}$. Funkcija pripadnosti praznom skupu $\forall x \in \mathcal{X}, \mu_{\emptyset}(x) = 0$.

Dekartov proizvod $A_1 \times A_2 \times \dots \times A_n$ skupova $A_1, A_2, \dots, A_n, n \in \mathbb{N}$ je skup svih uređenih n -torki (x_1, x_2, \dots, x_n) takvih da je $x_1 \in A_1, x_2 \in A_2, \dots, x_n \in A_n$.

Relacija Skup $R \subseteq A_1 \times A_2 \times \dots \times A_n$ naziva se n -arna relacija. Relaciju možemo predstaviti pomoću karakteristične funkcije

$$\mu_R: X_1 \times X_2 \times \dots \times X_n \rightarrow \{0,1\}.$$

Diskrentna relacija je skup sa konačno mnogo elemenata.

Funkcija f je skup uređenih n -torki $(x_1, x_2, \dots, x_{n-1}, y)$ takvih da za svaki $(x_1, x_2, \dots, x_{n-1}) \in A_1 \times A_2 \times \dots \times A_{n-1}$ postoji samo jedan

$y \in Y$ tada funkciju pišemo kao $y = f(x_1, x_2, \dots, x_{n-1})$ odnosno

$$f: A_1 \times A_2 \times \dots \times A_{n-1} \rightarrow Y.$$

Partitivni skup $\mathcal{P}(A) = \{T | T \subseteq A\}$.

Broj elemenata (kardinalitet) konačnog skupa - $\text{card}(A) = |A| =$

$$\sum_{x \in X} \mu_A(x), \text{ za kontinualni skup } \text{card}(A) = |A| = \int_{x \in X} \mu_A(x) dx.$$

Konveksnost klasičnih skupova i relacija - kontinualna n -arna relacija $R \subseteq$

\mathbb{R}^n je konveksna ako za svaki element

$$u = (u_1, u_2, \dots, u_n) \in R \text{ i } v = (v_1, v_2, \dots, v_n) \in R \text{ važi}$$

$$\lambda u + (1 - \lambda)v \in R \quad \forall \lambda \in [0, 1].$$

Komplement skupa Skup A^c je komplement skupa A ako važi:

$$A^c = \{x | x \in X \wedge x \notin A\} \text{ ili } \mu_{A^c}(x) = \begin{cases} 1 & \text{ako je } \mu_A(x) = 0 \\ 0 & \text{ako je } \mu_A(x) = 1 \end{cases}$$

Presek skupova A i B u oznaci $A \cap B$ je skup čiji elementi pripadaju i skupu

$$A \text{ i skupu } B. \quad A \cap B = \{x | x \in A \wedge x \in B\}$$

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)).$$

Unija skupova A i B u oznaci $A \cup B$ je skup koji sadrži elemente koji

$$\text{pripadaju ili skupu } A \text{ ili skupu } B. \quad A \cup B = \{x | x \in A \vee x \in B\}$$

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)).$$

Razlika skupova A i B u oznaci $B \setminus A$ je skup svih elemenata iz B koji ne

$$\text{pripadaju skupu } A. \quad B \setminus A = \{x | x \in B \wedge x \notin A\}.$$

Kompozicija relacija $R \subseteq X_1 \times X_2$ i $S \subseteq X_2 \times X_3$ u oznaci $R \circ S$ je skup

uređenih parova $(x_1, x_3) \in X_1 \times X_3$ takvih da postoji bar jedan

element $x_2 \in X_2$ za koji važi $(x_1, x_2) \in R$ i $(x_2, x_3) \in S$

$$R \circ S = \{(x_1, x_3) | \exists x_2 \in X_2 [(x_1, x_2) \in R \wedge (x_2, x_3) \in S]\}.$$

Fazi skup je kolekcija objekata ili elemenata iz univerzalnog skupa koje karakteriše neka fazi osobina.

Obeležavanje fazi skupova

Fazi skupovi se mogu obeležavati i kao skupovi uređenih parova $\tilde{A} =$

$$\{(x, \mu_{\tilde{A}}(x)) | x \in X, \mu_{\tilde{A}}(x) \in [0, 1]\} \text{ ili } x \in \tilde{A} =_{DF} \mu_{\tilde{A}}(x)$$

Gde je $\mu_{\tilde{A}}: X \rightarrow [0, 1]$ karakteristična funkcija, odnosno funkcija pripadnosti.

Skup \tilde{A} može da bude diskretan ili kontinualan.

Fazi relacija \tilde{R} je n dimenzioni fazi skup definisan nad Dekartovim proizvodom $\tilde{R} \subseteq X_1 \times X_2 \times \dots \times X_n$. Možemo je predstaviti na sledeći način

$$\tilde{R} = \{((x_1, x_2, \dots, x_n), \mu_{\tilde{R}}(x_1, x_2, \dots, x_n)) \mid (x_1, x_2, \dots, x_n) \in X_1 \times X_2 \times \dots \times X_n \wedge \mu_{\tilde{R}}(x_1, x_2, \dots, x_n) \in [0,1]\}.$$

Fazi partitivni skup $\tilde{\mathcal{P}}(A) = \{\tilde{T} \mid \tilde{T} \subseteq A\}$ skup svih fazi podskupova \tilde{T} definisanih nad skupom A .

Visina fazi skupa $\tilde{A} \in \tilde{\mathcal{P}}(X)$ u oznaci $hgt(\tilde{A})$ ili $h(\tilde{A})$ je supremum (ili maksimum u slučaju da je univerzalni skup konačan) funkcije pripadnosti $\mu_{\tilde{A}}(x)$

$$hgt(\tilde{A}) = h(\tilde{A}) = \sup_{x \in X} \mu_{\tilde{A}}(x)$$

Normiran fazi skup za fazi skup \tilde{A} kažemo da je *normiran* ako je $h(\tilde{A}) = 1$.

Jezgro fazi skupa $\tilde{A} \in \tilde{\mathcal{P}}(X)$ u oznaci $core(\tilde{A}) = C(\tilde{A})$

je klasičan skup svih elemenata $x \in X$ takvih da im je vrednost funkcije pripadnosti 1:

$$core(\tilde{A}) = C(\tilde{A}) = \{x \in X \mid \mu_{\tilde{A}}(x) = 1\}$$

Potpora fazi skupa $\tilde{A} \in \tilde{\mathcal{P}}(X)$ u oznaci $supp(\tilde{A}) = S(\tilde{A})$ je klasičan skup elemenata $x \in X$ takvih da je za njih vrednost funkcije pripadnosti različita od 0:

$$supp(\tilde{A}) = S(\tilde{A}) = \{x \in X \mid \mu_{\tilde{A}}(x) > 0\}$$

α *preseka* $cut_{\alpha}(\tilde{A}) = [\tilde{A}]^{\alpha} = A_{\alpha}$ fazi skupa $\tilde{A} \in \tilde{\mathcal{P}}(X)$ je klasičan skup svih elemenata $x \in X$ takvih da pripadaju skupu \tilde{A} u meri ne manjoj od α gde je $\alpha \in [0,1]$:

$$cut_{\alpha}(\tilde{A}) = [\tilde{A}]^{\alpha} = \{x \in X \mid \mu_{\tilde{A}}(x) \geq \alpha\}$$

Broj elemenata fazi skupa $card(\tilde{A}) = |\tilde{A}|$ diskretnog fazi skupa $\tilde{A} \in \tilde{\mathcal{P}}(X)$ sa konačnom potporom definiše se na sledeći način

$$card(\tilde{A}) = \sum_{x \in X} \mu_{\tilde{A}}(x) = \sum_{x \in supp(\tilde{A})} \mu_{\tilde{A}}(x)$$

ako je \tilde{A} kontinualan fazi skup tada se broj elemenata računa kao

$$\text{card}(\tilde{A}) = \int_{x \in X} \mu_{\tilde{A}}(x) dx$$

Fazi skup \tilde{A} je *konveksan* ako je svaki α -presek konveksan skup u klasičnom smislu.

$$\lambda u + (1 - \lambda)v \in [A]^\alpha \quad \forall \lambda \in [0,1] \quad \forall \alpha \in [0,1]$$

Trougaona norma (t-norma) je funkcija $T: [0,1]^2 \rightarrow [0,1]$ sa osobinama asocijativnosti, komutativnosti, monotonosti i graničnim uslovima (videti [58])

- (i) $T(a, T(b, c)) = T(T(a, b), c)$
- (ii) $T(a, b) \leq T(c, d)$ ako je $a \leq c$ i $b \leq d$
- (iii) $T(a, b) = T(b, a)$
- (iv) $T(a, 1) = a$

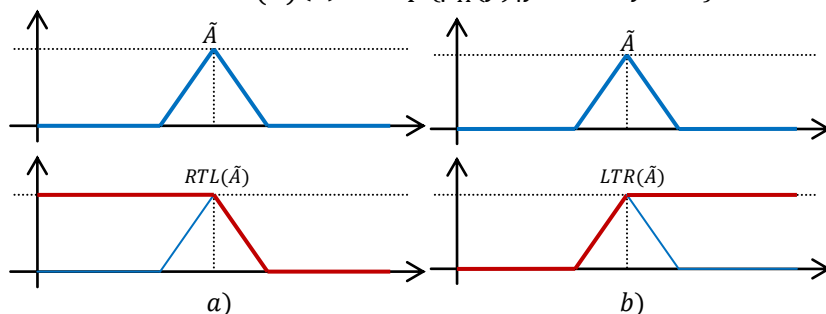
Zatvorenje fazi skupa *desno prema levo* i *levo prema desno* (slika 1.2) definišu se na sledeći način:

$RTL(\tilde{A})$ Zatvorenje fazi skupa *desno prema levo* (definisano u radu [59])

$$RTL(\tilde{A})(x) = \sup \{ \mu_{\tilde{A}}(y) \mid y \in X \wedge x \leq y \}$$

$LTR(\tilde{A})$ Zatvorenje fazi skupa *levo prema desno* (videti [59])

$$LTR(\tilde{A})(x) = \sup \{ \mu_{\tilde{A}}(y) \mid y \in X \wedge y \leq x \}$$



Slika 1.2: RTL-LTR . Zatvorenje fazi skupa a) desno prema levo b) levo prema desno

Model neprecizne tačke u ovoj doktorskoj disertaciji baziran je na klasi fazi skupova \mathcal{F}^n predstavljenoj u radovima [1], [10], [45].

Funkcije pripadnosti $u: R^n \rightarrow [0,1]$, klase fazi skupova \mathcal{F}^n zadovoljavaju sledeće osobine:

- i) $(\forall u \in \mathcal{F}^n)(\exists_1 P \in R^n) u(P) = 1,$

- ii) $(\forall X_1, X_2 \in R^n)(\lambda \in [0,1]) u(\lambda X_1 + (1 - \lambda)X_2) \geq \min(u(X_1), u(X_2))$,
- iii) funkcija u je deo po deo kontinualna a u tačkama prekida definisana je gornjoj tački,
- iv) $[u]^\alpha = \{X | X \in R^n, u(X) \geq \alpha\}$ α -presek je konveksan skup.

U slučaju da je $n = 1$, element $u \in \mathcal{F}^n$ je fazi broj. Fazi broj u , je *nenegativan* ako je definisan nad R^+ , odnosno sa \mathcal{F}^+ obeležavaćemo skup svih nenegativnih fazi brojeva.

U ovoj doktorskoj disertaciji kao osnova za prostorne funkcije korišten je pristup u kojem se rastojanje posmatra kao fazi broj a koncept fazi uređenja baziran je na konceptu uređenja baziranom na sličnostima koji je prezentovan u [55], odnosno ranije u [23]. U nastavku su date osnovne korišćene definicije.

Definicija 1.1 Neka je X neprazan skup i funkcija T je t-norma. Fazi relacija $E: X \times X \rightarrow [0,1]$, naziva se *fazi relacija ekvivalencije* (ili *T-ekvivalencije*) u odnosu na t-normu T , ako važe sledeći aksiomi za $x, y, z \in X$:

- (i) Refleksivnost: $E(x, x) = 1$.
- (ii) Simetričnost: $E(x, y) = E(y, x)$.
- (iii) T-tranzitivnost: $T(E(x, y), E(y, z)) \leq E(x, z)$.

Definicija 1.2 Neka je $E: X \times X \rightarrow [0,1]$ fazi relacija *T-ekvivalencije*. Fazi relacija $L: X \times X \rightarrow [0,1]$ naziva se *fazi uređenje* u odnosu na normu T i fazi ekvivalenciju E (kratko *T-E-uređenje*) ako za svako $x, y, z \in X$ važi sledeće:

- (i) *E-reflexivnost*: $E(x, y) \leq L(x, y)$.
- (ii) *T-E-antisimetričnost*: $T(L(x, y), L(y, x)) \leq E(x, y)$.
- (iii) *T-tranzitivnost*: $T(L(x, y), L(y, z)) \leq L(x, z)$.

Buckley i Eslami pretpostavili su u svojoj definiciji *fazi metričkog prostora* [45] da je rastojanje između dve fazi tačke koje imaju isto jezgro nenegativan fazi broj $\tilde{0}$. Kaleva i Seikkala definišu u [17] da bi samo rastojanje između istih fazi tačaka trebalo da bude $\tilde{0}$. U definiciji *fazi metričkog prostora* koja se koristi u ovom radu prihvaćen je uslov (i) iz [45] i preostali uslovi iz [17], odnosno sledeća definicija matričkog prostora

Definicija 1.3 Neka je X neprazan skup, $\tilde{d}: X \times X \rightarrow \mathcal{F}^+$, funkcije $L, R: [0,1] \times [0,1] \rightarrow [0,1]$ su simetrične, asocijativne i neopadajuće funkcije zpo oba argumenta i $L(0,0) = 0$, $R(1,1) = 1$. Uređenu četvorku

(X, \tilde{d}, L, R) nazivamo *fazi metrički prostor* a funkciju \tilde{d} *fazi metrika*, ako i samo ako važe sledeći uslovi:

- (i) $\tilde{d}(x, y) = \tilde{0} \Leftrightarrow [x]^1 = [y]^1$.
- (ii) $\tilde{d}(x, y) = \tilde{d}(y, x)$ za svako $x, y \in X$.
- (iii) Za svako $x, y \in X$:
 - a. $\tilde{d}(x, y)(s + t) \geq L(d(x, z)(s), d(z, y)(t))$ ako
 $s \leq \lambda_1(x, z) \wedge t \leq \lambda_1(z, y) \wedge s + t \leq \lambda_1(x, y)$
 - b. $\tilde{d}(x, y)(s + t) \leq R(d(x, z)(s), d(z, y)(t))$ ako
 $s \geq \lambda_1(x, z) \wedge t \geq \lambda_1(z, y) \wedge s + t \geq \lambda_1(x, y)$,

gde je fazi broja $\tilde{d}(x, y)$ definisan pomoću α –preseka

$$[\tilde{d}(x, y)]^\alpha = [\lambda_\alpha(x, y), \rho_\alpha(x, y)] \quad (x, y \in X, 0 < \alpha \leq 1).$$

Fazi broj $\tilde{0}$ zovemo *fazi nula*, i predstavlja nenegativan fazi broj takav da važi $[\tilde{0}]^1 = 0$.

Model proširivog softverskog sistema za upravljanje geoprostorom u uslovima neodređenosti

Cilj ove disertacije je formiranje proširive softverske arhitekture za upravljanje geoprostorom u uslovima neodređenosti. Kao osnova za opis nepreciznih geometrijskih objekata izabrana je teorija fazi skupova. Prilikom izbora modela za opis geometrijskih objekata potrebno je realan objekat što vernije ili detaljnije opisati kako bi greška modela bila što je moguće manja. Ovako defisane modele je potom potrebno analizirati, skladištiti i na osnovu njih izvoditi zaključke. Uzimajući u obzir da je u praktičnoj primeni količina prostornih podataka koji se analiziraju, skladište i obrađuju veoma velika, potrebno je nivo detaljnosti prilagoditi praktičnim potrebama. U radu su predloženi modeli koji predstavljaju kompromis između dva suprostavljena zahteva: što detaljniji model, kako bi odstupanje modela od realnog objekta bilo što manje i jednostavnija i upotrebljiva praktična primena ovako definisanih modela.

Prvo je izvršeno modeliranje podataka potom je dat predlog modela za opis procesa i na kraju je data softverska arhitektura kao okvir za implementaciju definisanih modela. Na taj način ostvaren je integrisani pristup u kojem se vodi računa o modelima podataka, njihovom korišćenju (procesima) i na kraju praktičnoj primeni (softverska arhitektura) kao jedinstvenoj celini.

Model podataka za opis prostorne neodređenosti

Realne prostorne objekte možemo podeliti u tri kategorije: realni objekti za čiju reprezentaciju je dovoljno poznavanje pozicije (*tačka*), realni objekti za čiju reprezentaciju je dovoljno poznavanje dužine (*kriva linija*) i realni objekti za čiju reprezentaciju je dovoljno poznavanje površine (*poligon*).

U nastavku su definisani modeli za opis *tačkastih* (fazi tačka), *pravolinijskih* (fazi duž) i *jednostavnih površinskih* (fazi trougao i fazi krug) nepreciznih geometrijskih objekata pomoću fazi skupova.

Definicija 2.1 Fazi tačka u $P \in \mathbb{R}^n$, u oznaci \tilde{P} je fazi skup čija funkcija pripadnosti $\mu_{\tilde{P}} \in \mathcal{F}^n$.

Kako je u radu [38] istaknuto, jedna od fundamentalnih osobina kolekcije tačkastih objekata je konveksni omotač. Ova osobina intenzivno se koristi u praktičnim realizacijama čitavog niza algoritama. Definicije fazi konveksnog omotača skupa fazi tačaka kao i njihove osobine jedan su od originalnih doprinosa ove doktorske teze a prezentovane su u radu [60].

Definicija 2.2 Neka je $\mathcal{A} \subset \mathcal{F}^n$. Tada se skup $CH(\alpha, \mathcal{A})$ naziva α -konveksni omotač skupa \mathcal{A} ako je $CH(\alpha, \mathcal{A})$ najmanji konveksni omotač koji sadrži sve α preseke elemenata koji pripadaju skupu \mathcal{A} .

Definicija 2.3 Neka je $\mathcal{A} \subset \mathcal{F}^n$. Tada se skup $\partial CH(\alpha, \mathcal{A}) \subset CH(\alpha, \mathcal{A})$ naziva rub α -konveksnog omotača skupa \mathcal{A} ako se $\partial CH(\alpha, \mathcal{A})$ sastoji od tačaka iz $CH(\alpha, \mathcal{A})$ takvih da sve okoline sadrže bar jednu tačku koja pripadana skupu $CH(\alpha, \mathcal{A})$ i jednu tačku izvan skupa $CH(\alpha, \mathcal{A})$.

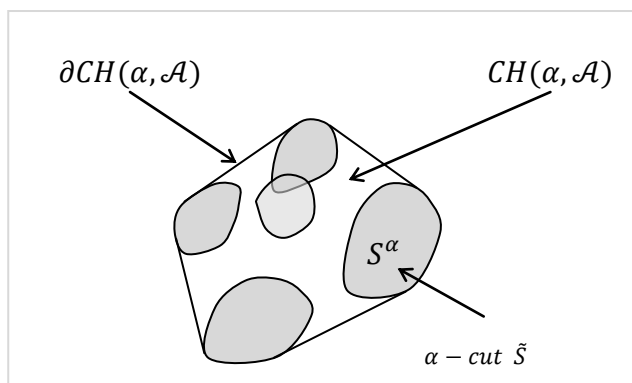
Geometrijska ilustracija definicija 2.2 i 2.3 prikazana je na slici 2.1

Definicija 2.4 Neka je $\mathcal{A} \subset \mathcal{F}^n$. Tada se minimalni podskup skupa \mathcal{A} čiji α -konveksni omotač je isti kao i α -konveksni omotač skupa \mathcal{A} naziva α -konveksni rub skupa \mathcal{A} i obeležava sa $ED(\alpha, \mathcal{A})$.

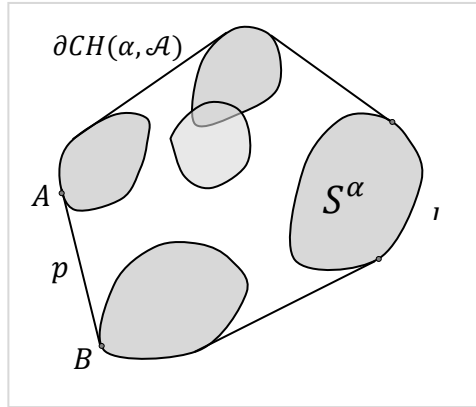
Na osnovu prethodnih definicija važi $CH(\alpha, ED(\alpha, \mathcal{A})) = CH(\alpha, \mathcal{A})$.

Definicija 2.5 Neka je $\mathcal{A} \subset \mathcal{F}^n$. Skup $\partial \mathcal{A} = ED(1, \mathcal{A})$ nazivamo konveksni rub skupa \mathcal{A} jačine m , gde je

$$m = 1 - \inf \{ \alpha \mid 0 \leq \alpha \leq 1 \wedge ED(\alpha, \mathcal{A}) = ED(1, \mathcal{A}) \}.$$



Slika 2.1 Geometrijska ilustracija skupova $CH(\alpha, \mathcal{A})$ i $\partial CH(\alpha, \mathcal{A})$



Slika 2.2: Geometrijska ilustracija teoreme 2.1

Definicija 2.6 Neka je $\mathcal{A} \subset \mathcal{F}^n$. Skup $\mathcal{A}^o = \mathcal{A} \setminus \partial\mathcal{A}$ nazivamo unutrašnjost skupa \mathcal{A} jačine m , gde je m jačina konveksnog ruba skupa \mathcal{A} .

U sledećoj teoremi dokazaćemo da α -konveksni rub skupa \mathcal{A} sadrži prave linije.

Teorema 2.1 Neka je $\mathcal{A} \subset \mathcal{F}^2$. Ako postoji tačka $C \in \partial CH(\alpha, \mathcal{A}) \subset \mathbb{R}^2$, takva da ne pripada uniji svih α -preseka tačaka koje pripadaju skupu $ED(\alpha, \mathcal{A})$, tada tačka C leži na liniji koja povezuje dve tačke sa ruba koje pripadaju α -preseku tačaka koje pripadaju skupu $ED(\alpha, \mathcal{A})$, i sem krajnjih tačaka linija na kojoj leži C ne sadrži ni jednu tačku koja pripada nekom α -preseku tačaka iz $ED(\alpha, \mathcal{A})$.

Geometrijska ilustracija teoreme 2.1 prikazana je na slici 2.2.

Dokaz:

Za dokaz teoreme 2.1 prvo ćemo dokazati sledeću lemu.

Lema 2.1 Neka je $G \subset \mathbb{R}^2$ konveksni skup i tačka C pripada rubu skupa G , i ne pripada pravoj liniji koja povezuje dve tačke iz G . Ako se tačka C ukloni iz G skup koji dobijemo je takođe konveksan.

Dokaz: Pretpostavimo da skup $G \setminus \{C\}$ nije konveksan. Tada postoji duž \overline{AB} koja povezuje tačke A i B iz $G \setminus \{C\}$ i prolazi kroz C . Kako je C rubna tačka skupa, sledi $C = A$ ili $C = B$, što je u kotradikciji sa $C \in G \setminus \{C\}$. ■

Sada možemo pristupiti dokazivanju teoreme 2.1.

Neka je $D = \bigcup_{\tilde{S} \in \mathcal{A}} [\tilde{S}]^\alpha$ unija svih α -preseka tačaka iz \mathcal{A} . Pretpostavimo da je p kriva koja povezuje tačke A i B tako da je $p \cap \partial CH(\alpha, \mathcal{A}) \cap D = \{A, B\}$ i $p \neq \overline{AB}$. Skup $CH(\alpha, \mathcal{A})$ je konveksan po definiciji iz čega sledi da je duž \overline{AB} sadržana u $CH(\alpha, \mathcal{A})$. Kako p sem tačaka A i B ne sadrži druge tačke iz D , postoji linija koja povezuje tačke A i B , tada je skup $G = \overline{AB} \cup (CH(\alpha, \mathcal{A}) \setminus p)$ takav da je $CH(\alpha, \mathcal{A}) \subset G \subset D$, i na osnovu Leme 2.1. sledi da je skup G konveksan. To je u suprotnosti sa definicijom skupa $CH(\alpha, \mathcal{A})$ kao najmanjeg konveksnog skupa koji sadrži skup D . ■

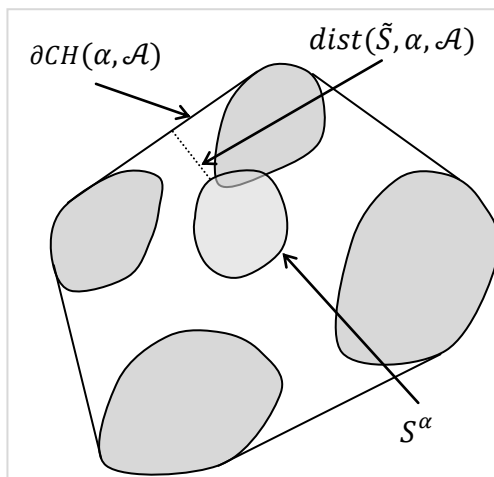
U nastavku ćemo definisati rastojanje između unutrašnje tačke i ruba.

Definicija 2.7 Rastojanje $dist(\tilde{S}, \alpha, \mathcal{A})$ između unutrašnje tačke \tilde{S} ($\tilde{S} \in \mathcal{A}$) i ruba α -konveksnog omotača $\partial CH(\alpha, \mathcal{A})$ je minimalno rastojanje između tačaka iz α -preseka fazi tačke \tilde{S} ($[\tilde{S}]^\alpha$) i ruba α -konveksnog omotača skupa \mathcal{A} ($\partial CH(\alpha, \mathcal{A})$):

$$dist(\tilde{S}, \alpha, \mathcal{A}) = \min\{d(C, P) \mid C \in [S]^\alpha, P \in \partial CH(\alpha, \mathcal{A})\}$$

gde je $d(C, P)$ Euklidsko rastojanje između tačaka iz \mathbb{R}^n .

Na slici 2.3 prikazana je geometrijska ilustracija rastojanja $dist(\tilde{S}, \alpha, \mathcal{A})$.



Slika 2.3: Geometrijska ilustracija rastojanja $dist(\tilde{S}, \alpha, \mathcal{A})$

Linearni fazi prostor - Fazi tačka

Zbog potreba analize, obrade i skladištenja fazi tačku opisanu u definiciji 2.1 odnosno njenu funkciju pripadnosti moramo da predstavimo što je moguće jednostavnije a da pri tom zadržimo osnovne osobine koje ih karakterišu. U nastavku će biti prikazan linearni fazi prostor u kojem se funkcija pripadnosti koja karakteriše fazi tačku predstavlja kao uređeni par čija prva komponenta je ista kao i kod klasične precizne tačke dok je druga komponenta realan broj koji karakteriše meru nepreciznosti. Ovako definisan linearni fazi prostor i njegove osobine takođe predstavljaju originalni doprinos teze.

Definicija 2.8 *Linearni fazi prostor* je skup $\mathcal{H}^n \subset \mathcal{F}^n$ čiji elementi su funkcije, koje pored osobina iz Definicije 2.1 imaju i sledeće osobine:

- i) Simetrične u odnosu na jezgro $S \in \mathbb{R}^n$
 $(\mu(S) = 1), \mu(V) = \mu(M) \wedge \mu(M) \neq 0 \Rightarrow d(S, V) = d(S, M)$
 gde je $d(S, M)$ rastojanje u \mathbb{R}^n .

- ii) Linearno opadajuće u odnosu na rastojanje tačke i jezgra:
 ako je $r \neq 0$

$$\mu_{\tilde{S}}(V) = \begin{cases} 1 - \frac{d(S, V)}{|r_S|} & \text{ako je } d(S, V) < r \\ 0 & \text{ako je } d(S, V) \geq r \end{cases}$$

ako je $r = 0$

$$\mu(V) = \begin{cases} 1 & \text{ako je } S = V \\ 0 & \text{ako je } S \neq V \end{cases}$$

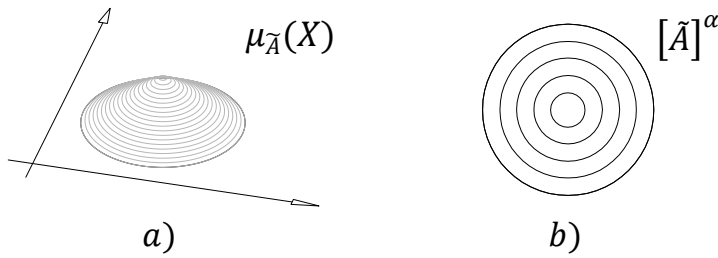
gde je $d(S, V)$ rastojanje tačke V i jezgra S ($V, S \in \mathbb{R}^n$) i $r \in \mathbb{R}$ je konstanta.

Elemente prostora \mathcal{H}^n u nastavku ćemo predstavljati kao uređene parove gde je prva komponenta element iz \mathbb{R}^n a druga komponenta iz \mathbb{R} :

$$\tilde{S} = (S, r_S) \quad S \in \mathbb{R}^n \quad r_S \in \mathbb{R}$$

Geometrijska ilustracija funkcije pripadnosti fazi tačke i α –preseka fazi tačke prikazana je na slici 2.4.

Teorema 2.2 i teorema 2.3 daju osobine α –preseka elemenata linearnog fazi prostora.



Slika 2.4: Geometrijska ilustracija
a) Funkcija pripadnosti fazi tačke b) α –preseci fazi tačke

Teorema 2.2 Neka je $\tilde{S} \in \mathcal{H}^n$ fazi tačka. Tada α –presek tačke $\tilde{S} = (S, s_r)$, u oznaci $[\tilde{S}]^\alpha$ ima sledeći oblik:

- (i) Zatvoreni interval $[S - (1 - \alpha)|r_S|, S + (1 - \alpha)|r_S|]$ ako je $n = 1$
- (ii) Krug sa centrom u $S \in \mathbb{R}^2$ i radijusom $(1 - \alpha)|r_S|$ ako je $n = 2$
- (iii) Sfera sa centrom u $S \in \mathbb{R}^3$ i radijusom $(1 - \alpha)|r_S|$ ako je $n = 3$

Za slučajeve (i), (ii) i (iii), funkcija pripadnosti kojom je okarakterisan linearni fazi prostor ima sledeći oblik:

$$\mu_S(X) = \max\left(0, 1 - \frac{d(S, X)}{|r_S|}\right)$$

gde je $d(S, X)$ metrika u prostoru \mathbb{R}^n .

Na osnovu definicije α –preseka, dobijamo

$$[\tilde{S}]^\alpha = \{X | X \in \mathbb{R}^n, 1 - \frac{d(X, S)}{|r_S|} \geq \alpha\}, \text{ i}$$

$$[\tilde{S}]^\alpha = \{X | X \in \mathbb{R}^n, d(X, S) \leq (1 - \alpha)|r_S|\}$$

Na osnovu ovoga imamo:

- (i) Za $n = 1$ primer rastojanja $d(x, y) = |x - y|$,
 $[S]^\alpha = \{x | x \in \mathbb{R}, |x - S| \leq (1 - \alpha)|r_S|\}$ iz čega sledi
 $[S]^\alpha = [S - (1 - \alpha)r_S, S + (1 - \alpha)|r_S|]$
- (ii) Za $n = 2$ primer rastojanja
 $d(X, S) = \sqrt{(x_1 - s_1)^2 + (x_2 - s_2)^2}$, gde je
tačka $X(x_1, x_2) \in \mathbb{R}^2$ a $S(s_1, s_2) \in \mathbb{R}^2$ je jezgro fazi tačke \tilde{S} .
Na osnovu prethodnog dobijamo

$$[\tilde{S}]^\alpha = \{X \mid X \in \mathbb{R}^2, \sqrt{(x_1 - s_1)^2 + (x_2 - s_2)^2} \leq (1 - \alpha)|r_S|\}$$

što predstavlja krug sa centrom u S i poluprečnikom $(1 - \alpha)|r_S|$

(iii) Za $n = 3$ primer rastojanja

$d(X, S) = \sqrt{(x_1 - s_1)^2 + (x_2 - s_2)^2 + (x_3 - s_3)^2}$ gde je tačka $X(x_1, x_2, x_3) \in \mathbb{R}^3$, a tačka $S(s_1, s_2, s_3)$ je jezgro fazi tačke \tilde{S} .

Na osnovu prethodnog dobijamo

$[S]^\alpha = \{X \mid X \in \mathbb{R}^3, \sqrt{(x_1 - s_1)^2 + (x_2 - s_2)^2 + (x_3 - s_3)^2} \leq (1 - \alpha)|r_S|\}$ što predstavlja sferu sa centrom u S i poluprečnikom $(1 - \alpha)|r_S|$.

Teorema 2.3. Neka je $\tilde{A} \in \mathcal{H}^2$ i $\alpha_1, \alpha_2 \in [0, 1]$. Tada za svako $\alpha_1 > \alpha_2$ sledi $[\tilde{A}]^{\alpha_1} \subset [\tilde{A}]^{\alpha_2}$.

Dokaz: Na osnovu teoreme 2.2 svi alfa preseki fazi tačke \tilde{A} su krugovi sa centrom u A i poluprečnikom $(1 - \alpha)|r_a|$, iz ovoga sledi da za $\alpha_1 > \alpha_2 \rightarrow (1 - \alpha_1)r_a < (1 - \alpha_2)r_a$ odnosno $[\tilde{A}]^{\alpha_1} \subset [\tilde{A}]^{\alpha_2}$ ■

U nastavku će biti definisane osnovne operacije nad linearnim fazi prostorom \mathcal{H}^2 definisanim nad poljem \mathbb{R}^2 . Pored definicija operacija biće predstavljene i njihove osobine koje će se koristiti prilikom konstruisanja složenijih fazi geometrijskih struktura.

Definicija 2.9 Neka su $\tilde{A}, \tilde{B} \in \mathcal{H}^2$. Operator $+$: $\mathcal{H}^2 \times \mathcal{H}^2 \rightarrow \mathcal{H}^2$ zovemo zbir dve fazi tačke ako važi sledeće:

$$\tilde{A} + \tilde{B} =_{DF} (A + B, r_A + r_B) = ((a_1 + b_1, a_2 + b_2), r_A + r_B)$$

gde je $A + B$ zbir dva vektora u \mathbb{R}^2 a $r_A + r_B$ je zbir dva skalara.

Teorema 2.4 Neka je \mathcal{H}^2 linearni fazi prostor i operator $+$ zbir fazi tačaka. Tada je uređen par $(\mathcal{H}^2, +)$ Abelova grupa.

Dokaz Pokazaćemo da uređen par $(\mathcal{H}^2, +)$ zadovoljava sledeće osobine:

- (i) *zatvorenost*
- (ii) *asocijativnost* $\tilde{A} + (\tilde{B} + \tilde{C}) = (\tilde{A} + \tilde{B}) + \tilde{C}$
- (iii) *neutralni element* $\tilde{o} + \tilde{A} = \tilde{A} + \tilde{o} = \tilde{A}$
- (iv) *inverzni element* $\tilde{A} + \tilde{A}^{-1} = \tilde{o}$
- (v) *komutativnost* $\tilde{A} + \tilde{B} = \tilde{B} + \tilde{A}$

(i) na osnovu definicije

$$(ii) \quad \tilde{A} + (\tilde{B} + \tilde{C}) = ((a_1, a_2), r_A) + (((b_1, b_2), r_B) + ((c_1, c_2), r_C)) = ((a_1, a_2), r_A) + ((b_1 + c_1, b_2 + c_2), r_B + r_C) = ((a_1 + b_1 + c_1, a_2 + b_2 + c_2), r_A + r_B + r_C) = (((a_1 + b_1) + c_1, (a_2 + b_2) + c_2), (r_A + r_B) + r_C) = (\tilde{A} + \tilde{B}) + \tilde{C}$$

$$(iii) \quad \tilde{0} + \tilde{A} = ((0, 0), 0) + ((a_1, a_2), r_A) = ((0 + a_1, 0 + a_2), 0 + r_A) = ((a_1 + 0, a_2 + 0), r_A + 0) = \tilde{A} + \tilde{0} = ((a_1, a_2), r_A) = \tilde{A}$$

$$(iv) \quad \tilde{A} + \tilde{X} = \tilde{0} \rightarrow ((a_1 + x_1, a_2 + x_2), r_A + r_X) = ((0, 0), 0) \rightarrow \text{dobijamo tri jednačine } a_1 + x_1 = 0 \text{ i } a_2 + x_2 = 0, r_A + r_X = 0 \rightarrow x_1 = -a_1, x_2 = -a_2, r_X = -r_A \rightarrow \tilde{X} = ((-a_1, -a_2), -r_A)$$

$$(v) \quad \tilde{A} + \tilde{B} = \tilde{B} + \tilde{A} \rightarrow ((a_1, a_2), r_A) + ((b_1, b_2), r_B) = ((a_1 + b_1, a_2 + b_2), r_A + r_B) \rightarrow ((b_1 + a_1, b_2 + a_2), r_B + r_A) \rightarrow \tilde{B} + \tilde{A}$$

Definicija 2.10 Operacije $\oplus: \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$ i $\odot: \mathbb{R}^2 \times \mathbb{R}^2 \rightarrow \mathbb{R}^2$ definišemo na sledeći način:

$$(i) \quad (\alpha_1, \alpha_2) \oplus (\beta_1, \beta_2) = (\alpha_1 + \beta_1, \alpha_2 + \beta_2)$$

$$(ii) \quad (\alpha_1, \alpha_2) \odot (\beta_1, \beta_2) = (\alpha_1 \cdot \beta_1, \alpha_2 \cdot \beta_2)$$

Teorema 2.5 Uređena trojka $(\mathbb{R}^2, \oplus, \odot)$ je polje.

Dokaz:

Na osnovu Definicije 2.10, neutralni element za sabiranje je $(0, 0)$ a neutralni element za množenje je $(1, 1)$. U nastavku ćemo ove elemente obeležavati kratko sa 0 i 1 respektivno. Ostale osobine polja se trivijalno dokazuju.

Definicija 2.11 Levi i desni proizvod uređenog para i fazi tačke

$\cdot^L: \mathbb{R}^2 \times \mathcal{H}^2 \rightarrow \mathcal{H}^2$ i $\cdot^R: \mathcal{H}^2 \times \mathbb{R}^2 \rightarrow \mathcal{H}^2$ definišemo na sledeći način:

$$(i) \quad \forall (\alpha_1, \alpha_2) \in \mathbb{R}^2, \forall \tilde{A} = (A, r_A) \in \mathcal{H}^2$$

$$(\alpha_1, \alpha_2) \cdot^L (A, r_A) = (\alpha_1 \cdot A, \alpha_2 \cdot r_A)$$

$$(ii) \quad \forall (\alpha_1, \alpha_2) \in \mathbb{R}^2, \forall \tilde{A} = (A, r_A) \in \mathcal{H}^2$$

$$(A, r_A) \cdot^R (\alpha_1, \alpha_2) = (A \cdot \alpha_1, r_A \cdot \alpha_2) = (\alpha_1 \cdot A, \alpha_2 \cdot r_A)$$

gde je \cdot množenje skalara i uređenog para (vektora)

Kako su operacije \cdot^L i \cdot^R dualne u nastavku ćemo ih kratko obeležavati sa \cdot , a zvaćemo ih proizvod uređenog para i fazi tačke.

Teorema 2.6 Neka je \mathcal{H}^2 linearni fazi prostor definisan nad \mathbb{R}^2 , operacija $+$ zbir fazi tačaka, a operacija \cdot je proizvod uređenog para i fazi tačke. Tada uređena četvorka $(\mathcal{H}^2, +, \mathbb{R}^2, \cdot)$ predstavlja *vektorski prostor*.

Dokaz Pokazaćemo da uređena četvorka $(\mathcal{H}^2, +, \mathbb{R}^2, \cdot)$ zadovoljava sledeće osobine:

- (i) $(\mathcal{H}^2, +)$ Abelova grupa
- (ii) $\forall \alpha, \beta \in \mathbb{R}^2, \forall \tilde{A} \in \mathcal{H}^2 \quad \alpha \cdot (\beta \cdot \tilde{A}) = (\alpha \odot \beta) \cdot \tilde{A}$
- (iii) $\forall \alpha \in \mathbb{R}^2, \forall \tilde{A}, \tilde{B} \in \mathcal{H}^2 \quad \alpha \cdot (\tilde{A} + \tilde{B}) = \alpha \cdot \tilde{A} + \alpha \cdot \tilde{B}$
- (iv) $\forall \alpha, \beta \in \mathbb{R}^2, \forall \tilde{A} \in \mathcal{H}^2 \quad (\alpha \oplus \beta) \cdot \tilde{A} = \alpha \cdot \tilde{A} + \beta \cdot \tilde{A}$
- (v) $\forall \tilde{A} \in \mathcal{H}^2 \quad 1 \cdot \tilde{A} = \tilde{A}$

Osobina (i) dokazana je u teoremi 2.4.

Dokaz osobine (ii): $\alpha \cdot (\beta \cdot \tilde{A}) = (\alpha_1, \alpha_2) \cdot ((\beta_1, \beta_2) \cdot (A, r_a)) = (\alpha_1, \alpha_2) \cdot (\beta_1 A, \beta_2 r_a) = (\alpha_1, \alpha_2) \cdot (\alpha_1 \beta_1 A, \alpha_2 \beta_2 r_a) = ((\alpha_1 \beta_1) A, (\alpha_2 \beta_2) r_a) = (\alpha_1 \beta_1, \alpha_2 \beta_2) \cdot (A, r_a) = (\alpha \odot \beta) \cdot \tilde{A}$.

Dokaz osobine (iii): $\alpha \cdot (\tilde{A} + \tilde{B}) = (\alpha_1, \alpha_2) \cdot ((A, r_A) + (B, r_B)) = (\alpha_1, \alpha_2) \cdot (A + B, r_A + r_B) = (\alpha_1(A + B), \alpha_2(r_A + r_B)) = (\alpha_1 A + \alpha_1 B, \alpha_2 r_A + \alpha_2 r_B) = \alpha_1 A + \alpha_1 B, \alpha_2 r_A + \alpha_2 r_B = \alpha_1 A, \alpha_2 r_A + \alpha_1 B, \alpha_2 r_B = \alpha \cdot A + \alpha \cdot B$.

Dokaz osobine (iv): $(\alpha \oplus \beta) \cdot \tilde{A} = ((\alpha_1, \alpha_2) \oplus (\beta_1, \beta_2)) \cdot (A, r_A) = (\alpha_1 + \beta_1, \alpha_2 + \beta_2) \cdot (A, r_A) = ((\alpha_1 + \beta_1) A, (\alpha_2 + \beta_2) r_A) = (\alpha_1 A + \beta_1 A, \alpha_2 r_A + \beta_2 r_A) = \alpha_1 A + \beta_1 A, \alpha_2 r_A + \beta_2 r_A = \alpha \cdot A + \beta \cdot A$

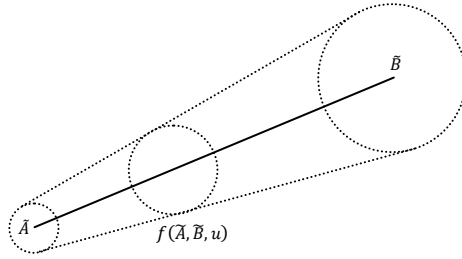
Dokaz osobine (v): $1 \cdot \tilde{A} = (1, 1) \cdot (A, r_A) = (1A, 1r_A) = (A, r_A) = \tilde{A}$ ■

Definicija 2.12 Neka je \mathcal{H}^2 linearni fazi prostor. Funkciju $f: \mathcal{H}^2 \times \mathcal{H}^2 \times [0, 1] \rightarrow \mathcal{H}^2$ nazivamo *linearna kombinacija* fazi tačaka ako važi:

$$f(\tilde{A}, \tilde{B}, u) =_{df} \tilde{A} + u \cdot (\tilde{B} - \tilde{A})$$

Napomena: *linearnu kombinaciju fazi tačaka možemo da pišemo i kao*

$$f(\tilde{A}, \tilde{B}, u) =_{df} (1 - u)\tilde{A} + u \cdot \tilde{B}$$



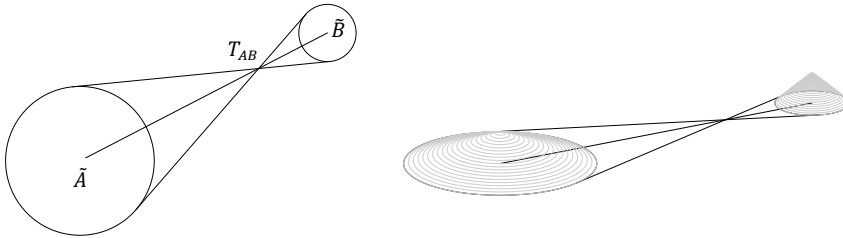
Slika 2.5: Geometrijska ilustracija linearne kombinacije dve fazi tačke
Geometrijska ilustracija linearne kombinacije dve fazi tačke prikazana je na slici 2.5.

Definicija 2.13 Neka su fazi tačke $\tilde{A}, \tilde{B} \in \mathcal{H}^2$ i $\tilde{A} \neq \tilde{B}$. Tačku $T_{AB} \in \mathbb{R}^2$ nazivamo *interni homotetički centar* ako važi sledeće:

$$T_{AB} = A + \frac{a_r}{a_r + b_r} (B - A)$$

gde je $\tilde{A} = (A, a_r)$ i $\tilde{B} = (B, b_r)$.

Geometrijska ilustracija internog homotetičkog centra prikazana je na slici 2.6.

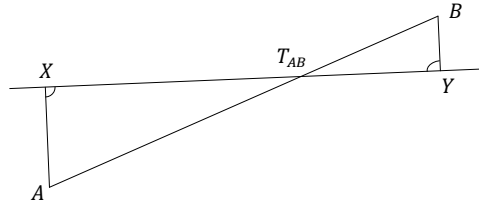


Slika 2.6: Geometrijska ilustracija internog homotetičkog centra

Teorema 2.7 Neka su tačke $\tilde{A}, \tilde{B} \in \mathcal{H}^2$, $\tilde{A} \neq \tilde{B}$ i tačka $T_{AB} \in \mathbb{R}^2$ je *interni homotetički centar*. Tada za $\forall \alpha \in [0,1]$ i $\forall X \in [\tilde{A}]^\alpha$ postoji $Y \in [\tilde{B}]^\alpha$ takvo da važi

$$Y = T_{AB} - \frac{b_r}{a_r} (T_{AB} - X)$$

Geometrijska ilustracija teoreme 2.7 prikazana je na slici 2.7.



Slika 2.7: Geometrijska ilustracija teoreme 2.7

Dokaz:

Treba pokazati da se tačka Y nalazi u $[\tilde{B}]^\alpha$, odnosno da je $d(B, Y) \leq (1 - \alpha)|r_B|$.

Na osnovu definicije *internog homotetičkog centra* i sličnosti trouglova AXT_{AB} i BYT_{AB} (Slika 2.7) sledi da je $X - T_{AB} = \frac{a_r}{b_r}(T_{AB} - Y)$ odnosno $T_{AB} = X + \frac{a_r}{a_r + b_r}(Y - X)$ ili $Y = T_{AB} - \frac{b_r}{a_r}(T_{AB} - X)$. Sad treba pokazati da tačka Y pripada $[\tilde{B}]^\alpha$ odnosno da važi $d(B, Y) \leq (1 - \alpha)b_r$. Kako tačka $X \in [\tilde{A}]^\alpha$ sledi da je $d(A, X) \leq (1 - \alpha)|a_r|$. Na osnovu sličnosti trouglova AXT_{AB} i BYT_{AB} sledi da je

$$\frac{d(A, X)}{d(A, T_{AB})} = \frac{d(B, Y)}{d(B, T_{AB})}.$$

Kako je $\frac{d(A, T_{AB})}{d(B, T_{AB})} = \frac{|a_r|}{|b_r|}$, sledi da je

$$d(B, Y) = \frac{|b_r|}{|a_r|} d(A, X). \quad (2.1)$$

$$d(A, X) \leq (1 - \alpha)|a_r| \rightarrow \frac{|b_r|}{|a_r|} d(A, X) \leq \frac{|b_r|}{|a_r|} (1 - \alpha)|a_r| \quad (2.2)$$

ako zamenimo (2.1) u (2.2) dobijamo $d(B, Y) \leq (1 - \alpha)|b_r|$ iz čega sledi da tačka Y pripada skupu $[\tilde{B}]^\alpha$ ■

Nakon definicija koje opisuju osobine pojedinačnih fazi tačaka u nastavku je dato nekoliko definicija koje se odnose na osobine skupa fazi tačaka. Ove osobine će se koristiti prilikom dokazivanja osobina složenijih geometrijskih struktura, kao što su fazi duž i fazi trougao.

Uređenje elemenata \mathbb{R}^1 linearnog fazi prostora

Prvo ćemo dati definicije kojima se uređuje \mathbb{R}^1 linearni fazi prostor.

Definicija 2.14 Neka je fazi prostor \mathcal{H} definisan nad \mathbb{R}^1 . Fazi relacije \leq^{RF} i \leq^{LF} skupa \mathcal{H} definišu sledeće funkcije pripadnosti:

$$\mu(\tilde{A} \leq^{RF} \tilde{B}) = \begin{cases} 0 & \text{ako je } A > B \\ \frac{B - A}{r_A - r_B} & \text{ako je } A \leq B \wedge A + r_A > B + r_B \\ 1 & \text{ako je } A \leq B \wedge A + r_A \leq B + r_B, \end{cases}$$

$$\mu(\tilde{A} \leq^{LF} \tilde{B}) = \begin{cases} 0 & \text{ako je } A > B \\ \frac{B - A}{r_B - r_A} & \text{ako je } A \leq B \wedge A - r_A > B - r_B \\ 1 & \text{ako je } A \leq B \wedge A - r_A \leq B - r_B, \end{cases}$$

redom, gde su $\tilde{A} = (A, r_A)$ i $\tilde{B} = (B, r_B)$ fazi tačke iz \mathcal{H} , A je jezgro tačke \tilde{A} i r_A je parameter koji određuje funkciju pripadnosti tačke \tilde{A} .

Napomena 2.1 U definiciji 2.14 korištena je notacija desno levo i levo u desno zatvorenje fazi skupa definisano u [55].

Fazi relacija \leq^{RF} predstavlja meru ili stepen sigurnosti (realan broj iz intervala $[0,1]$) sa kojim možemo da tvrdimo da je fazi broj \tilde{A} manji ili jednak fazi broju \tilde{B} , u skladu sa njihovim levo prema desno ili desno prema levo zatvorenju. Malo preciznije fazi relacija \leq^{RF} predstavlja meru skupa svih α takvih da je $[\text{RTL}(\tilde{A})]^\alpha \subseteq [\text{RTL}(\tilde{B})]^\alpha$, gde je $\text{RTL}(\tilde{A})$ najmanji fazi nadskup skupa \tilde{A} koji ima nerastuću funkciju pripadnosti.

U sledeće dve teoreme sledi dokaz da fazi relacije date u definiciji 2.14 predstavljaju fazi uređenje u skupu \mathcal{H} .

Teorema 2.8 Neka je T_M -ekvivalencija $E: \mathcal{H} \times \mathcal{H} \rightarrow [0,1]$ data sa:

$$E(\tilde{A}, \tilde{B}) = \begin{cases} 1 & \text{ako je } A = B \wedge r_A = r_B \\ 0 & \text{inače,} \end{cases}$$

i minimum T_M -norma ($T_M(a, b) = \min(a, b)$), tada je fazi relacija \leq^{RF} uređenje u skladu sa definicijom fazi uređenja.

Dokaz:

Treba da dokažemo da fazi relacija \leq^{RF} ima sledeće osobine:

- (i) E -refleksivnost $E(\tilde{A}, \tilde{B}) \leq \mu(\tilde{A} \leq^{RF} \tilde{B})$.
- (ii) T_M - E anti-simetričnost $T_M(\mu(\tilde{A} \leq^{RF} \tilde{B}), \mu(\tilde{B} \leq^{RF} \tilde{A})) \leq E(\tilde{A}, \tilde{B})$.
- (iii) T_M -tranzitivnost $T_M(\mu(\tilde{A} \leq^{RF} \tilde{B}), \mu(\tilde{B} \leq^{RF} \tilde{C})) \leq \mu(\tilde{A} \leq^{RF} \tilde{C})$.

 E -refleksivnost:

Ako je $E(\tilde{A}, \tilde{B}) = 0$ dokaz je trivijalan. U protivnom iz $E(\tilde{A}, \tilde{B}) = 1$ sledi $A = B \wedge r_A = r_B$, što ima za posledicu $\mu(\tilde{A} \leq^{RF} \tilde{B}) = 1 \Rightarrow 1 \leq 1$.

 T_M - E anti-simetrija:

Dokaz je trivijalan za $E(\tilde{A}, \tilde{B}) = 1$. U protivnom ako je $E(\tilde{A}, \tilde{B}) = 0$ treba pokazati da je $\mu(\tilde{A} \leq^{RF} \tilde{B}) = 0 \vee \mu(\tilde{B} \leq^{RF} \tilde{A}) = 0$ tačno. Pretpostavimo da važi $\mu(\tilde{A} \leq^{RF} \tilde{B}) \neq 0 \wedge \mu(\tilde{B} \leq^{RF} \tilde{A}) \neq 0$. Smenom $\mu(\tilde{A} \leq^{RF} \tilde{B}) = a$ i $\mu(\tilde{B} \leq^{RF} \tilde{A}) = b$, razlikujemo četiri slučaja.

- 1) $a = 1 \wedge b = 1$;
 $(A \leq B \wedge A + r_A \leq B + r_B) \wedge (B \leq A \wedge B + r_B \leq A + r_A) \Rightarrow A = B \wedge r_A = r_B \Rightarrow E(\tilde{A}, \tilde{B}) = 1$, što je nemoguće zbog pretpostavke da je $E(\tilde{A}, \tilde{B}) = 0$.
- 2) $a = 1 \wedge b < 1$;
 $(A \leq B \wedge A + r_A \leq B + r_B) \wedge (B \leq A \wedge B + r_B > A + r_A) \Rightarrow A = B \wedge r_A \leq r_B \wedge r_B > r_A$, što je nemoguće.
- 3) $a < 1 \wedge b = 1$; veoma slično dokazu pod 2.
- 4) $a < 1 \wedge b < 1$
 $(A \leq B \wedge A + r_A > B + r_B) \wedge (B \leq A \wedge B + r_B > A + r_A) \Rightarrow A = B \wedge r_A > r_B \wedge r_B < r_A$, što je nemoguće.

 T_M -tranzitivnost:

Treba pokazati da $\min(\mu(\tilde{A} \leq^{RF} \tilde{B}), \mu(\tilde{B} \leq^{RF} \tilde{C})) \leq \mu(\tilde{A} \leq^{RF} \tilde{C})$ važi za svako \tilde{A}, \tilde{B} i \tilde{C} koji pripadaju \mathcal{H} definisanom nad R^1 . Ako sa A, B i C obeležimo jezgra tačaka \tilde{A}, \tilde{B} i \tilde{C} , redom, a sa r_A, r_B i r_C odgovarajuće maksimalno rastojanje od jezgra za koje je funkcija pripadnosti različita od 0, tada razlikujemo sledeće slučajeve: (1.) $A > B \vee B > C$; (2.) $A \leq B \wedge B \leq C$.

1. Za $A > B \vee B > C$ dokaz je trivijalan, jer je $\mu(\tilde{A} \leq^{RF} \tilde{B}) = 0$ ili $\mu(\tilde{B} \leq^{RF} \tilde{C}) = 0$.
2. Za $A \leq B \wedge B \leq C$ razlikuju se dva slučaja (a) $A + r_A \leq C + r_C$ i (b) $A + r_A > C + r_C$.
 - (a) Trivijalan, jer je $\mu(\tilde{A} \leq^{RF} \tilde{C}) = 1$.
 - (b) Neka je $\mu(\tilde{A} \leq^{RF} \tilde{C}) = \frac{C-A}{r_A-r_C} = a < 1$. Tada je nejednakost $\min(\mu(\tilde{A} \leq^{RF} \tilde{B}), \mu(\tilde{B} \leq^{RF} \tilde{C})) \leq a$ tačna ako je $\mu(\tilde{A} \leq^{RF} \tilde{B}) \leq a$ ili $\mu(\tilde{B} \leq^{RF} \tilde{C}) \leq a$, i ponovo se izdvajaju tri različita slučaja: (i) $A + r_A \leq B + r_B$; (ii) $A + r_A > B + r_B \geq C + r_C$; (iii) $B + r_B < C + r_C$. U slučaju (i) $\mu(\tilde{A} \leq^{RF} \tilde{B}) = 1$ i (iii) $\mu(\tilde{B} \leq^{RF} \tilde{C}) = 1$. Posledica prethodnih tvrđenja su sledeća tri slučaja: (i) $1 \leq a \vee \frac{C-B}{r_B-r_C} \leq a$; (ii) $\frac{B-A}{r_A-r_B} \leq a \vee \frac{C-B}{r_B-r_C} \leq a$; (iii) $\frac{B-A}{r_A-r_B} \leq a \vee 1 \leq a$.

Nastavićemo kontradikcijom. Pretpostavimo suprotno: (i) $1 > a \wedge \frac{C-B}{r_B-r_C} > a$; (ii) $\frac{B-A}{r_A-r_B} > a \wedge \frac{C-B}{r_B-r_C} > a$; (iii) $\frac{B-A}{r_A-r_B} > a \wedge 1 > a$.

Tada za (i) važi da je $\frac{B-A}{r_A-r_B} \geq 1 > a \wedge \frac{C-B}{r_B-r_C} > a$, dok za (iii) važi $\frac{B-A}{r_A-r_B} > a \wedge \frac{C-B}{r_B-r_C} \geq 1 > a$. Slučajeve (i), (ii) i (iii) možemo predstaviti kao $\frac{B-A}{r_A-r_B} > a \wedge \frac{C-B}{r_B-r_C} > a$. Što ima za posledicu: $B - A > a(r_A - r_B) \wedge C - B > a(r_B - r_C) \Rightarrow C - A > a(r_A - r_C)$, odnosno $\frac{C-A}{r_A-r_C} > a$, što je nemoguće u skladu sa pretpostavkom da je $a = \frac{C-A}{r_A-r_C}$. ■

Teorema 2.9 Neka je T -ekvivalencija $E: \mathcal{H} \times \mathcal{H} \rightarrow [0,1]$ data sa:

$$E(\tilde{A}, \tilde{B}) = \begin{cases} 1 & \text{ako je } A = B \wedge r_A = r_B \\ 0 & \text{inače,} \end{cases}$$

i T kao minimum T_M -norma ($T_M(a, b) = \min(a, b)$), tada je fazi relacija \leq^{LF} fazi uređenje u skladu sa definicijom 1.2.

Dokaz: Analogan dokazu teoreme 2.8 ■

Napomena 2.2 Teorema 2.8 i teorema 2.9 dokazane su za minimum t-normu. Tada, u skladu sa radovima [25] i [58] sledi da su ove teoreme tačne za bilo koju drugu t-normu.

U sledeće dve definicije iskorištena je ideja o maksimalnom i minimalnom elementu na način kako je to predloženo u radovima [55], [26].

Definicija 2.15 Neka je \leq^{RF} fazi uređenje, \mathcal{H} je linearni prostor definisan nad \mathbb{R} , i $\mathcal{A} \subset \mathcal{H}$. *Maksimalni element skupa* \mathcal{A} je fazi skup određen funkcijom pripadnosti

$$\mu_G(\tilde{X}) = \min (\{\mu(\tilde{Y} \leq^{RF} \tilde{X}) \mid \forall \tilde{Y} \in \mathcal{A}\}).$$

Definicija 2.16 Neka je \leq^{LF} fazi uređenje, \mathcal{H} je linearni fazi prostor definisan nad \mathbb{R}^1 , i $\mathcal{A} \subset \mathcal{H}$. *Minimalni element skupa* \mathcal{A} je fazi skup određen funkcijom pripadnosti

$$\mu_L(\tilde{X}) = \min (\{\mu(\tilde{X} \leq^{RF} \tilde{Y}) \mid \forall \tilde{Y} \in \mathcal{A}\}).$$

Fazi duž

U prethodnim definicijama i teoremama opisali smo osnovni element linearnog fazi prostora i dali pregled njegovih elementarnih osobina. Fazi tačku koristimo da opišemo poziciju nekog realnog objekta u uslovima kad postoji određena nesigurnost da je izmerena vrednost potpuno tačna. Najčešće se ova nesigurnost u praktičnim primenama zanemaruje. Postoje primene u kojima realne objekte ne karakteriše samo pozicija već ceo niz uniformno raspoređenih tačaka. Ove tačke mogu da budu raspoređene duž krive koja ima svoj početak i kraj. Krivu koja povezuje dve tačke nazivamo putanja ili put. Najjednostavniji oblik krive je duž koja se može predstaviti kao najkraći put između dve tačke.

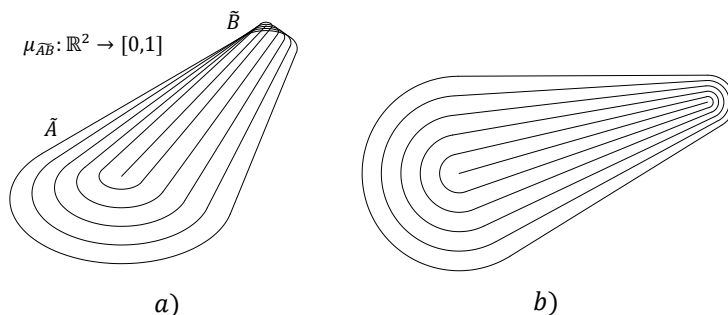
Ako su tačke koje karakterišu putanju neprecizne i sama putanja je neprecizna, odnosno potrebno ju je opisati na sličan način kao i nepreciznu tačku. U nastavku slede definicije i teoreme za reprezentaciju takvih realnih objekata ili pojava.

Definicija 2.17 Neka je \mathcal{H}^2 linearni fazi prostor a funkcija f linearna kombinacija fazi tačkaka \tilde{A} i \tilde{B} . Tada fazi skup

$$\widetilde{AB} = \bigcup_{u \in [0,1]} f(\tilde{A}, \tilde{B}, u)$$

nazivamo *fazi duž*.

Skup svih *fazi duži* definisanih nad \mathcal{H}^2 u nastavku ćemo obeležavati sa \mathcal{L}^2 . Elementi skupa \mathcal{L}^2 su uređeni parovi fazi tačkaka.



Slika 2.8: a) Funkcija pripadnosti fazi duži b) α –preseci fazi duži

Na slici 2.8 a) prikazana je geometrijska ilustracija funkcije pripadnosti fazi duži a na slici 2.8 b) prikazana je geometrijska ilustracija α preseka fazi duži \widetilde{AB} .

Teorema 2.10 Neka je \widetilde{AB} fazi duž definisana nad \mathcal{H}^2 i $[\widetilde{AB}]^\alpha$ α -presek fazi skupa \widetilde{AB} , tada je $[\widetilde{AB}]^1 = \overline{AB}$.

Dokaz: $[\widetilde{AB}]^1 = [\bigcup_{u \in [0,1]} f(\tilde{A}, \tilde{B}, u)]^1 = \bigcup_{u \in [0,1]} [f(\tilde{A}, \tilde{B}, u)]^1 = \bigcup_{u \in [0,1]} (A + u \cdot (B - A)) = \overline{AB}$ ■

Definicija 2.18 Neka je \widetilde{AB} fazi duž definisana nad \mathcal{H}^2 i $X \in \mathbb{R}^2$. Fazi tačku $\tilde{X}' \subset \widetilde{AB}$ nazivamo *fazi slika* tačke X na fazi duži \widetilde{AB} , a realan broj $u \in [0,1]$ nazivamo *karakteristična vrednost slike tačke X na fazi duži \widetilde{AB}* ako važi sledeće:

- (i) $\tilde{X}' = \tilde{A} + u(\tilde{B} - \tilde{A})$,
- (ii) $d(X, [\tilde{X}']^1) = \min \{d(X, Y) | \forall Y \in [\widetilde{AB}]^1\}$,

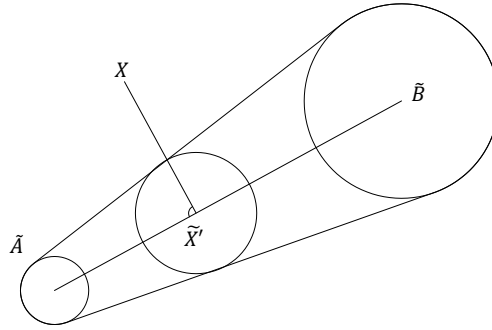
$$(iii) u = \min \left(1, \max \left(0, \frac{(x_1 - a_1)(b_1 - a_1) + (x_2 - a_2)(b_2 - a_2)}{(b_1 - a_1)^2 + (b_2 - a_2)^2} \right) \right)$$

gde je $X = (x_1, x_2)$, $\tilde{A} = ((a_1, a_2), a_r)$ i $\tilde{B} = ((b_1, b_2), b_r)$.

Napomena: Ako je karakteristična vrednost slike tačke X jednaka 0 tada je fazi slika početna tačka, a ako je karakteristična vrednost jednaka 1 tada je fazi slika krajnja tačka inače je neka tačka između.

Osobina (iii) može da se izvede iz (i) i (ii) pa bi mogla da se izdvoji u posebnu teoremu, gde bi se prilikom izvodjenja pokazalo da je skalarni proizvod vektora $(B - A)(X - X') = 0$ jer je $\cos(\beta) = 0$ odnosno ugao između vektora je $\frac{\pi}{2}$. Rešavanjem skalarnog proizvoda po nepoznatoj promenljivoj u dobija se izraz (iii).

Na slici 2.9 prikazana je geometrijska ilustracija fazi slike tačke i karakteristične vrednosti tačke.



Slika 2.9: Ilustracija definicije fazi slike i karakteristične vrednosti

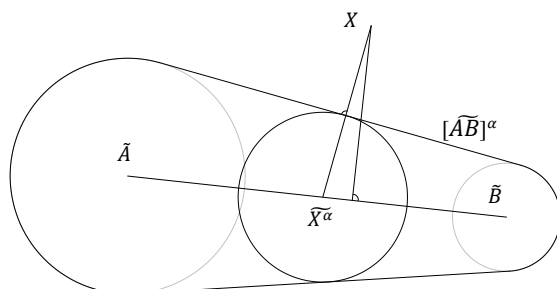
Pored definicije fazi slike i karakteristične vrednosti slike definišaćemo i njihove analogone čija vrednost zavisi od $\alpha \in [0,1]$, odnosno predstavlja projekciju tačke X na duž \overline{AB} kroz tačku sa ruba α –preseka koja je najbliža tački X .

Definicija 2.19 Neka je \overline{AB} fazi duž definisana nad \mathcal{H}^2 i $X \in \mathbb{R}^2$. Fazi tačku $\overline{X}^\alpha \subset \overline{AB}$ nazivamo α – fazi slika tačke X na fazi duži \overline{AB} , a realan broj $u^\alpha \in [0,1]$ nazivamo α –karakteristična vrednost slike tačke X na fazi duži \overline{AB} ako važi sledeće:

- (i) $\overline{X}^\alpha = \tilde{A} + u^\alpha(\tilde{B} - \tilde{A})$,
- (ii) $d(X, [\overline{X}^\alpha]^\alpha) = \min \{d(X, Y) | \forall Y \in [\overline{AB}]^\alpha\}$,
- (iii) $u^\alpha = \min \left(1, \max \left(0, \frac{(1-\alpha)(b_r - a_r)}{d(A,B)^3} + \frac{(x_1 - a_1)(b_1 - a_1) + (x_2 - a_2)(b_2 - a_2)}{d(A,B)^2} \right) \right)$

gde je $X = (x_1, x_2)$, $\tilde{A} = ((a_1, a_2), a_r)$ i $\tilde{B} = ((b_1, b_2), b_r)$.

Na slici 2.10 prikazana je geometrijska ilustracija Definicije 2.19.



Slika 2.10: Geometrijska ilustracija α – fazi slika tačke X na fazi duži $\tilde{A}\tilde{B}$

Teorema 2.11 Neka je $\tilde{X}' \in \mathcal{H}^2$ fazi slika tačke $X \in \mathbb{R}^2$ i realan broj $u \in [0,1]$ karakteristična vrednost slike tačke $X \in \mathbb{R}^2$ na fazi duži $\tilde{A}\tilde{B} \in L^2$. Tada vrednost funkcije pripadnosti fazi duži $\tilde{A}\tilde{B}$ u tački X računamo na sledeći način:

$$\mu_{\tilde{A}\tilde{B}}(X) = \begin{cases} \mu_{\tilde{A}}(X) & \text{ako je } u_m = 0 \\ \mu_{\tilde{X}'_T}(X) & \text{za } 0 < u_m < 1 \\ \mu_{\tilde{B}}(X) & \text{ako je } u_m = 1 \end{cases}$$

gde je fazi tačka $\tilde{X}'_T = \tilde{A} + u_m(\tilde{B} - \tilde{A})$ i $u_m = u + \frac{(b_r - a_r) d(X, X')^2}{x'_r d(A, B)^2}$

Dokaz

Kako je po definiciji fazi duž $\tilde{A}\tilde{B}$ unija linearnih kombinacija fazi tačaka \tilde{A} i \tilde{B} a vrednost funkcije unije dva fazi skupa $\mu_{\tilde{A} \cup \tilde{B}}(X) = \max(\mu_{\tilde{A}}(X), \mu_{\tilde{B}}(X))$ sledi da se vrednost funkcije pripadnosti tačke X fazi duži $\tilde{A}\tilde{B}$ računa kao $\mu_{\tilde{A}\tilde{B}}(X) = \max(\mu_{\tilde{Y}}(X) | \tilde{Y} \subset \tilde{A}\tilde{B})$.

$$(g^2(u))' = \frac{(2(c_2 + c_1u)^2c_1 + 2(c_4 + c_3u)^2c_3)(a_r + c_5u)^2 - ((c_2 + c_1u)^2 + (c_4 + c_3u)^2)(2(a_r + c_5u)c_5)}{(a_r + c_5u)^4}$$

$$(g^2(u))' = \frac{2(a_r + c_5u)((c_2 + c_1u)^2c_1 + (c_4 + c_3u)^2c_3)(a_r + c_5u) - ((c_2 + c_1u)^2 + (c_4 + c_3u)^2)c_5}{(a_r + c_5u)^4}$$

$$(g^2(u))' = \frac{2((c_2 + c_1u)^2c_1 + (c_4 + c_3u)^2c_3)(a_r + c_5u) - ((c_2 + c_1u)^2 + (c_4 + c_3u)^2)c_5}{(a_r + c_5u)^3}$$

$$(g^2(u_m))' = 0$$

$$((c_2 + c_1u_m)^2c_1 + (c_4 + c_3u_m)^2c_3)(a_r + c_5u_m) - ((c_2 + c_1u_m)^2 + (c_4 + c_3u_m)^2)c_5 = 0$$

$$(c_2c_1 + c_1^2u_m + c_3c_4 + c_3^2u_m)(a_r + c_5u_m) = c_5(c_2^2 + 2c_1c_2u_m + c_1^2u_m^2 + c_4^2 + 2c_3c_4u_m + c_3^2u_m^2)$$

$$\text{nakon smena } c_6 = c_1c_2 + c_3c_4, \quad c_7 = c_1^2 + c_3^2 = d(A, B)^2 \text{ i}$$

$$c_8 = c_2^2 + c_4^2 = (a_1 - x_1)^2 + (a_2 - x_2)^2 = d(A, X)^2 \text{ dobijamo}$$

$$(c_6 + c_7u_m)(a_r + c_5u_m) = c_5(c_8 + 2c_6u_m + c_7u_m^2)$$

$$c_6a_r + c_6c_5u_m + c_7a_ru_m + c_7c_5u_m^2 = c_5c_8 + 2c_6c_5u_m + c_5c_7u_m^2$$

odnosno

$$u_m = \frac{c_5c_8 - c_6a_r}{c_7c_5 - c_5c_6}$$

nakon zamene smena

$$u_m = \frac{(b_r - a_r)d(A, X)^2 - c_6a_r}{d(A, B)^2a_r - (b_r - a_r)c_6} \text{ ako imenioc i brojilac podelimo sa } a_r \text{ i napravimo}$$

$$\text{smenu } c_9 = \frac{b_r - a_r}{a_r} \text{ dobijamo}$$

$$u_m = \frac{c_9d(A, X)^2 - c_6}{d(A, B)^2 - c_9c_6} \quad (2.3)$$

U pravouglom trouglu $AX'X$ vredi $d(A, X)^2 = d(A, X')^2 + d(X, X')^2$ a kako je $d(A, X') = u \cdot d(A, B)$ sledi $d(A, X)^2 = u^2d(A, B)^2 + d(X, X')^2$ i kako je po definiciji $u = \frac{-c_6}{d(A, B)^2}$. Jednačinu (2.3) sad možemo da pišemo kao

$$u_m = \frac{c_9(u^2d(A, B)^2 + d(X, X')^2) - c_6}{d(A, B)^2 - c_9c_6} \quad (2.4)$$

Ako u jednačini (2.4) brojilac i imenilac podelimo sa $d(A, B)^2$ i izvršimo smenu $u = \frac{-c_6}{d(A, B)^2}$ dobijamo

$$u_m = \frac{c_9u^2 + \frac{c_9d(X, X')^2}{d(A, B)^2} + u}{1 + c_9u} = \frac{u(1 + c_9u) + \frac{c_9d(X, X')^2}{d(A, B)^2}}{1 + c_9u}$$

Po definiciji fazi slike tačke X dobijamo $x'_r = a_r + u(b_r - a_r)$ odnosno kako je $c_9 = \frac{b_r - a_r}{a_r}$ dobijamo da je $\frac{x'_r}{a_r} = 1 + uc_9$ odnosno $uc_9 = 1 - \frac{x'_r}{a_r} = \frac{a_r - x'_r}{a_r}$ sledi:

$$u_m = u + \frac{\frac{(b_r - a_r) d(X, X')^2}{a_r} \frac{d(A, B)^2}{x'_r}}{a_r}$$

i na kraju dobijamo

$$u_m = u + \frac{(b_r - a_r) d(X, X')^2}{x'_r d(A, B)^2}$$

■

Teorema 2.12 Neka je \mathcal{H}^2 linearni fazi prostor, fazi duž \widetilde{AB} definisana nad \mathcal{H}^2 i određena fazi tačkama $\widetilde{A}, \widetilde{B} \in \mathcal{H}^2$. Tada važi

$$\widetilde{AB} = \widetilde{BA}.$$

Dokaz: Treba dokazati da za svako $X \in \mathbb{R}^2$ važi $\mu_{\widetilde{AB}}(X) = \mu_{\widetilde{BA}}(X)$.

Kako je po definiciji

$$\mu_{\widetilde{AB}}(X) = \max_{u \in [0,1]} \left(\max \left(0, 1 - \frac{d(A + u(B - A), X)}{|a_r + u(b_r - a_r)|} \right) \right)$$

izraz možemo napisati i kao

$$\mu_{\widetilde{AB}}(X) = \max_{u \in [0,1]} \left(\max \left(0, 1 - \frac{d(B + (1 - u)(A - B), X)}{|b_r + (1 - u)(a_r - b_r)|} \right) \right)$$

nakon smene $v = 1 - u$ i kako vredi

$\forall u \in [0,1] \rightarrow (1 - u) \in [0,1]$ odnosno $v \in [0,1]$

dobijamo

$$\mu_{\widetilde{AB}}(X) = \max_{v \in [0,1]} \left(\max \left(0, 1 - \frac{d(B + v(A - B), X)}{|b_r + v(a_r - b_r)|} \right) \right)$$

$$\mu_{\widetilde{AB}}(X) = \mu_{\widetilde{BA}} \quad \blacksquare$$

Fazi trougao

U nastavku su date definicije i osobine koje poseduje jedan od najjednostavnijih geometrijski oblika, fazi trougao. Fazi trougao, za razliku od prethodnih geometrijskih oblika, poseduje novu prostornu meru a to je površina. Često se realni objekti ili pojave grupišu u odnosu na jednu ili više zajedničkih karakteristika. Ako te osobine nemaju potpuno uniformni karakter ili ako je nemoguće potpuno tačno definisati granice površine koju zauzimaju takvi objekti, (npr. granice planine, šume, reke ili područja sa velikim zagađenjem vazduha) vrlo često se opisuju klasičnim geometrijskim oblicima a da se pritom vrši zanemarivanje određenih osobina koje su po svojem karakteru neodređene. Ova zanemarivanja vrlo često imaju svoje opravdanje u praktičnoj primeni jer je rad sa objektima koji poseduju kompleksnu geometriju često nemoguć. Definicije koje slede predstavljaju originalni doprinos ove teze a određene su kao kompromis između praktične primene gde je zahtev da se koriste što jednostavniji oblici i što kvalitetnijeg opisa u kojem su zanemarivanja nepreciznosti što je moguće manja.

Na slici 2.12 prikazan je primer jednog klasičnog "preciznog" trougla (a) i "nepreciznog" trougla (b).



Slika 2.12 Klasičan trougao a) i "neprecizni" trougao b)

Definicija 2.20 Neka su tačke $\tilde{A}, \tilde{B}, \tilde{C} \in \mathcal{H}^2$ takve da im jezgra nisu međusobno kolinearna ($\tilde{A} \neq \tilde{B} \neq \tilde{C}$) i neka je funkcija f linearna kombinacija dve fazi tačke. Fazi skup \widetilde{ABC} nazivamo *fazi trougao* ako važi

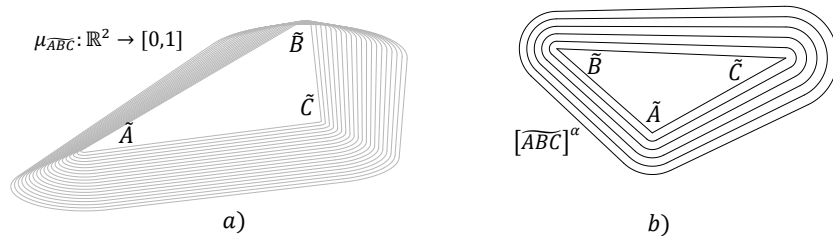
$$\widetilde{ABC} = \bigcup_{u=0}^1 f(\tilde{A}, \bigcup_{v=0}^1 f(\tilde{B}, \tilde{C}, v), u)$$

Vrednost funkcije pripadnosti tačke $X \in \mathbb{R}^2$ fazi trouglu \widetilde{ABC} označavaćemo sa $\mu_{\widetilde{ABC}}(X)$:

$$\mu_{\overline{ABC}}(X) = \max_{u \in [0,1], v \in [0,1]} \{\mu_{\tilde{Y}}(X) \mid \tilde{Y} = f(\tilde{A}, f(\tilde{B}, \tilde{C}, v), u)\}$$

Alfa presek fazi trougla \overline{ABC} označavaćemo sa $[\overline{ABC}]^\alpha$. Svaki fazi trougao jednoznačno je određen uređenom trojkom fazi tačaka $(\tilde{A}, \tilde{B}, \tilde{C})$.

Geometrijska ilustracija funkcije pripadnosti fazi trougla i α –preseka fazi trougla prikazana je na slici 1.13.



Slika 2.13: a) Funkcija pripadnosti fazi trougla b) α –preseci fazi trougla

Skup svih *fazi trouglova* definisanih nad \mathcal{H}^2 u nastavku ćemo obeležavati sa \mathcal{T}^2 . Elementi skupa \mathcal{T}^2 su uređene trojke fazi tačaka.

Definicija 2.21 Neka je fazi trougao \overline{ABC} definisan nad \mathcal{H}^2 . Fazi tačku $\tilde{X} \subset \overline{ABC}$ nazivamo *rubna tačka trougla \overline{ABC}* ako za svako $\alpha \in [0,1]$ postoji tačka $Y \in [\tilde{X}]^\alpha$ takva da svaka njezina okolina sadrži i tačke koje pripadaju skupu $[\overline{ABC}]^\alpha$ i tačke koje ne pripadaju skupu $[\overline{ABC}]^\alpha$.

Napomena:

Svi alfa preseci rubne fazi tačke dodiruju rub alfa preseka fazi trougla.

Definicija 2.22 Neka je fazi trougao \overline{ABC} definisan nad \mathcal{H}^2 . Fazi tačku $\tilde{X} \subset \overline{ABC}$ nazivamo *unutrašnja tačka trougla \overline{ABC}* ako tačka nije rubna.

Definicija 2.23 Neka je fazi trougao \overline{ABC} definisan nad \mathcal{H}^2 . Uniju svih rubnih fazi tačaka trougla \overline{ABC} zvaćemo fazi rub trougla \overline{ABC} i obeležavaćemo sa $\partial \overline{ABC}$.

Teorema 2.13 Neka je fazi trougao \widetilde{ABC} definisan nad \mathcal{H}^2 i fazi tačka $\tilde{X} \subset \partial\widetilde{ABC}$ je rubna tačka. Tada postoji jedinstven uređen par (u, v) takav da je

$$\tilde{X} = \tilde{A} + u(\tilde{B} + v(\tilde{C} - \tilde{B}) - \tilde{A})$$

i važi sledeće:

- (i) ako je $u = 0 \Leftrightarrow \tilde{X} = \tilde{A}$
- (ii) ako je $v = 0 \Leftrightarrow \tilde{X} \subset \widetilde{AB}$
- (iii) ako je $v = 1 \Leftrightarrow \tilde{X} \subset \widetilde{AC}$
- (iv) ako je $u = 1 \Leftrightarrow \tilde{X} \subset \widetilde{BC}$

Dokaz:

- (i) smer $u = 0 \Rightarrow \tilde{X} = \tilde{A}$ na osnovu definicije fazi trougla sledi $\tilde{X} = \tilde{A} + 0(\tilde{B} + v(\tilde{C} - \tilde{B}) - \tilde{A}) = \tilde{A} + 0 = \tilde{A}$
iz $\tilde{X} = \tilde{A} \Rightarrow u = 0$ sledi $\tilde{A} = \tilde{A} + u(\tilde{B} + v(\tilde{C} - \tilde{B}) - \tilde{A})$ odnosno $u(\tilde{B} + v(\tilde{C} - \tilde{B}) - \tilde{A}) = \tilde{o}$ ili $u(B + v(C - B) - A) = 0$. Kako jezgra tačkaka \tilde{A}, \tilde{B} i \tilde{C} nisu kolinearna sledi da ne postoji v takvo da se može napisati $A = B + v(C - B)$ iz čega sledi da je jednačina $u(B + v(C - B) - A) = 0$ tačna samo ako je $u = 0$.
- (ii) ako je $v = 0 \Rightarrow \tilde{X} = \tilde{A} + u(\tilde{B} + 0(\tilde{C} - \tilde{B}) - \tilde{A}) = \tilde{A} + u(\tilde{B} - \tilde{A})$
 $\tilde{X} \subset \widetilde{AB} \Rightarrow \tilde{X} = \tilde{A} + u(\tilde{B} - \tilde{A}) \Rightarrow \tilde{X} = \tilde{A} + u(\tilde{B} + \tilde{o} - \tilde{A})$
 $\tilde{o} = v(\tilde{C} - \tilde{B})$ po definiciji fazi trougla $\tilde{C} \neq \tilde{B}$ sledi da je jednačina $\tilde{o} = v(\tilde{C} - \tilde{B})$ tačna samo ako je $v = 0$.
- (iii) ako je $v = 1 \Rightarrow \tilde{X} = \tilde{A} + u(\tilde{B} + 1(\tilde{C} - \tilde{B}) - \tilde{A}) = \tilde{A} + u(\tilde{C} - \tilde{A})$
 $\tilde{X} \subset \widetilde{AC} \Rightarrow \tilde{X} = \tilde{A} + u(\tilde{C} + \tilde{o} - \tilde{A})$ kako je $\tilde{o} = \tilde{B} - \tilde{B}$ sledi
 $\tilde{X} = \tilde{A} + u(\tilde{C} + \tilde{B} - \tilde{B} - \tilde{A}) = \tilde{A} + u(\tilde{B} + \tilde{C} - \tilde{B} - \tilde{A})$ iz čega sledi $\tilde{X} = \tilde{A} + u(\tilde{B} + 1(\tilde{C} - \tilde{B}) - \tilde{A})$ odnosno $v = 1$.
- (iv) ako je $u = 1 \Rightarrow \tilde{X} = \tilde{A} + 1(\tilde{B} + v(\tilde{C} - \tilde{B}) - \tilde{A}) = \tilde{B} + v(\tilde{C} - \tilde{B})$
smer $\tilde{X} \subset \widetilde{BC} \Rightarrow u = 1$ dokazuje se analogno slučaju (iii)

Posledica prethodne teoreme je da se rubne tačke nalaze ili na duži \widetilde{AB} ili \widetilde{AC} ili \widetilde{CB} .

Teorema 2.14 Neka je fazi trougao \widehat{ABC} definisan nad \mathcal{H}^2 . Tada za svaku fazi tačku $\tilde{X} \in \partial\widehat{ABC}/\{\tilde{A}, \tilde{B}, \tilde{C}\}$ i $\alpha \in [0,1]$ važi:

- (i) postoji samo jedna tačka $T \in [\tilde{X}]^\alpha$ takva da sve njezine okoline sadrže i tačke iz $[\widehat{ABC}]^\alpha$ i tačke izvan tog skupa
- (ii) α –slika tačke T na fazi duži kojoj pripada \tilde{X} je tačka \tilde{X}

Dokaz: osobina pod (i) je direktna posledica teoreme 2.1. iz rada [60], koja govori da je konveksni omotač skupa fazi tačaka sastavljen od linija i lukova. Osobina pod (ii) je direktna posledica teoreme 2.18 iz rada [60].

Teorema 2.15 Neka je fazi trougao \widehat{ABC} definisan nad \mathcal{H}^2 . Tada za svako $X \in \mathbb{R}^2$ važi sledeće

$$\mu_{\widehat{ABC}}(X) = \mu_{\widehat{CAB}}(X) = \mu_{\widehat{BCA}}(X)$$

Dokaz: Dokaz je analogan dokazu teoreme 2.12

Posledica ove teoreme je da se za reprezentaciju fazi trougla može koristiti skup tri fazi tačke.

Fazi krug

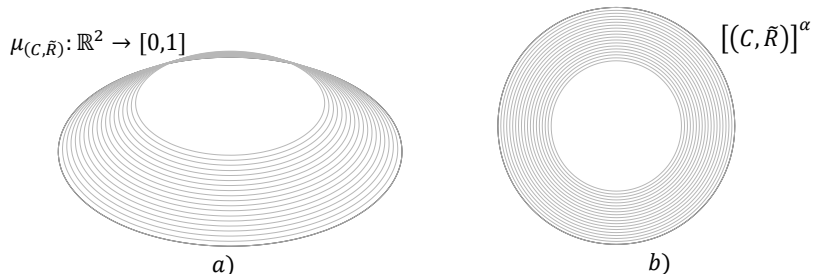
Fazi krug takođe je jedan od jednostavnijih nepreciznih geometrijskih oblika. Po analogiji sa definicijama fazi duži i fazi trougla, prilikom definisanja fazi kruga vođeno je računa da ovaj geometrijski oblik predstavlja proširenje klasičnog kruga, da ga je moguće predstaviti kao uniju skupa fazi tačaka i na kraju da ga je moguće efikasno koristiti u praktičnim primenama (broj parametara za skladištenje je minimalan).

Definicija 2.24 Neka je \mathcal{H} linearni fazi prostor definisan nad \mathbb{R} , fazi relacija \leq^{RF} je relacija uređenja elemenata iz \mathcal{H} , tačka $C \in \mathbb{R}^2$ i $\tilde{R} \in \mathcal{H}$. Tada uniju svih fazi tačaka $\tilde{A} \in \mathcal{H}^2$ takvih da važi

$$\mu(\tilde{d}(C, \tilde{A}) \leq^{RF} \tilde{R}) = 1$$

nazivamo *fazi krug* sa centrom u C i poluprečnikom \tilde{R} . U nastavku ćemo fazi krug obeležavati kao uređeni par (C, \tilde{R}) .

Na slici 2.14 data je geometrijska ilustracija funkcije pripadnosti fazi krugu i α –preseka fazi kruga.



Slika 2.14: a) Funkcija pripadnosti fazi krugu b) α –preseci fazi kruga

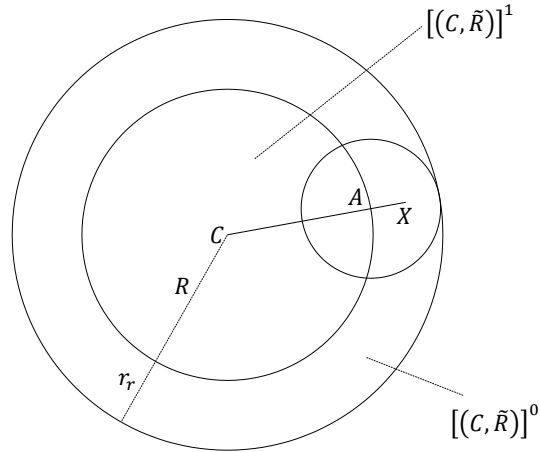
Teorema 2.16 Neka je *fazi krug* (C, \tilde{R}) definisan nad \mathcal{H}^2 . Tada vrednost funkcije pripadnosti kojom se karakteriše fazi krug računamo na sledeći način:

$$\mu_{(C, \tilde{R})}(X) = \max\left(0, \min\left(1, 1 - \frac{d(X, C) - R}{r_r}\right)\right)$$

gde je $\tilde{R} = (R, r_r)$.

Dokaz: Posmatramo tri slučaja (i) kad je $d(C, X) \leq R$, (ii) $R < d(C, X) \leq R + r_r$ i (iii) $R + r_r < d(C, X)$.

Slučaj (i) je trivijalan jer po definiciji fazi relacije \leq^{RF} postoji \tilde{A} sa jezgrom u X takvo da je $\mu(\tilde{d}(C, \tilde{A}) \leq^{RF} \tilde{R}) = 1$ iz čega sledi da je $\mu_{(C, \tilde{R})}(X) = 1$. U slučajevima (ii) i (iii) $d(C, X) = R + x$ gde je $x \geq 0$ odnosno postoji fazi tačka $\tilde{A} = (A, r_r)$ takva da je rastojanje $d(C, A) = R$, $d(A, X) = x$ i tačka A se nalazi na pravoj koja povezuje C i X (Slika). Na osnovu prethodnih iskaza $d(X, C) = d(C, A) + d(A, X) = R + x$ odnosno $x = d(X, C) - R$. Vrednost funkcije pripadnosti tačke X fazi tački \tilde{A} jednaka je $\mu_{\tilde{A}}(X) = \max\left(0, 1 - \frac{d(A, X)}{r_r}\right) = \max\left(0, 1 - \frac{d(X, C) - R}{r_r}\right)$ ■



Slika 2.15: Ilustracija teoreme 2.16

Prostorne operacije

Kao što je u prethodnim poglavljima istaknuto, za potrebe analize, skladištenja i zaključivanja potrebno je podatke obraditi. Na početku se izdvajaju njihove osnovne karakteristike u vidu *prostornih merenja* (dužina i površina). Potom, je potrebno omogućiti transformaciju putem *prostornih funkcija* (presek, unija, fazi konveksni omotač, translacija, rotacija, itd.). Poslednja grupa prostornih operacija su *prostorne relacije* (podudarnost, kolinearnost, itd.). U narednim odeljcima date su definicije i osobine osnovnih prostornih operacija grupisanih u tri grupe: prostorna merenja, prostorne funkcije i prostorne relacije. Ove definicije i osobine takođe predstavljaju originalni doprinos teze.

Prostorna merenja

Grupa funkcija koje se vrlo često koriste jesu funkcije kojima se mere rastojanja i površine. Prilikom merenja rastojanja između prostornih tipova podataka u slučaju da je jedan od argumenata iz grupe fazi podataka rezultat je nenegativan fazi broj.

Definicija 2.25 Neka je \mathcal{H}^2 linearni fazi prostor. Funkciju $\tilde{d}: \mathcal{H}^2 \times \mathcal{H}^2 \rightarrow \mathcal{H}^+$ nazivamo *fazi rastojanje* između *fazi tačaka* ako važi sledeće:

- (iv) $\tilde{d}(\tilde{X}, \tilde{Y}) = \tilde{0} \Leftrightarrow [\tilde{X}]^1 = [\tilde{Y}]^1$.
- (v) $\tilde{d}(\tilde{X}, \tilde{Y}) = \tilde{d}(\tilde{Y}, \tilde{X})$ za svako $\tilde{X}, \tilde{Y} \in \mathcal{H}^2$.
- (vi) Za svako $\tilde{X}, \tilde{Y} \in \mathcal{H}^2$:
 - a. $\tilde{d}(\tilde{X}, \tilde{Y})(s+t) \geq L(d(x,z)(s), d(z,y)(t))$ if
 $s \leq \lambda_1(x,z) \wedge t \leq \lambda_1(z,y) \wedge s+t \leq \lambda_1(x,y)$
 - b. $\tilde{d}(\tilde{X}, \tilde{Y})(s+t) \leq R(d(x,z)(s), d(z,y)(t))$ if
 $s \geq \lambda_1(x,z) \wedge t \geq \lambda_1(z,y) \wedge s+t \geq \lambda_1(x,y)$,

gde je $[\tilde{d}(\tilde{X}, \tilde{Y})]^\alpha = [\lambda_\alpha(x,y), \rho_\alpha(x,y)]$ ($x, y \in \mathbb{R}^+, 0 < \alpha \leq 1$) a $\tilde{0}$ je nenegativan fazi broj za koji važi $[\tilde{0}]^1 = 0$.

Definicija 2.26 Neka je \mathcal{H}^2 linearni fazi prostor i $\tilde{d}: \mathcal{H}^2 \times \mathcal{H}^2 \rightarrow \mathcal{F}^+$, $L, R: [0,1] \times [0,1] \rightarrow [0,1]$. Neka su funkcije L i R simetrične, asocijativne i neopadajuće funkcije po oba parametra i neka važi $L(0,0) = 0$ i $R(1,1) = 1$. Uređena četvorka $(\mathcal{H}^2, \tilde{d}, L, R)$ naziva se *fazi metrički prostor* a funkcija \tilde{d} *fazi metrika*.

Primer: Sledeće funkcije jesu *fazi metrika*

- (i) $\tilde{d}(\tilde{X}, \tilde{Y}) =_{DF} (d(X, Y), (r_X + r_Y))$
- (ii) $\tilde{d}(\tilde{X}, \tilde{Y}) =_{DF} (d(X, Y), \max(r_X, r_Y))$
- (iii) $\tilde{d}(\tilde{X}, \tilde{Y}) =_{DF} (d(X, Y), |r_X - r_Y|)$

Poslednji primer predstavlja klasičnu metriku.

U narednih nekoliko definicija definisaćemo rastojanja između različitih fazi geometrijskih objekata, npr. rastojanje između fazi tačke i fazi linije ili rastojanje između dva fazi trougla.

Definicija 2.27 Neka je \mathcal{H}^2 linearni fazi prostor, \mathcal{L}^2 skup svih *fazi duži* definisanih nad \mathcal{H}^2 , funkcija \tilde{d} rastojanje između fazi tačaka i funkcija μ_L fazi skup koji opisuje minimalni element u skupu rastojanja fazi tačaka, tada funkciju $dist: \mathcal{H}^2 \times \mathcal{L}^2 \rightarrow \mathcal{H}^+$ nazivamo rastojanje *fazi tačke i fazi linije* ako važi sledeće:

$$dist(\tilde{T}, \tilde{AB}) = \tilde{d}(\tilde{T}, \tilde{X})$$

gde je tačka $\tilde{X} \in \tilde{AB}$ i $\mu_L(\tilde{d}(\tilde{T}, \tilde{X})) = hgt(\{\tilde{d}(\tilde{T}, \tilde{Y}) | \forall \tilde{Y} \in \tilde{AB}\})$.

Definicija 2.28 Neka je \mathcal{H}^2 linearni fazi prostor, \mathcal{T}^2 skup svih *fazi trouglova* definisanih nad \mathcal{H}^2 i funkcija μ_L je fazi skup koji opisuje minimalni element u skupu rastojanja fazi tačaka, tada funkciju $dist: \mathcal{H}^2 \times \mathcal{L}^2 \rightarrow \mathcal{H}^+$ nazivamo rastojanje između *fazi tačke* i *fazi trougla* ako važi sledeće:

$$dist(\tilde{T}, \overline{ABC}) = \tilde{d}(\tilde{T}, \tilde{X})$$

gde je tačka $\tilde{X} \in \overline{ABC}$ i

$$\mu_L(\tilde{d}(\tilde{T}, \tilde{X})) = hgt(\{\tilde{d}(\tilde{T}, \tilde{Y}) | \forall \tilde{Y} \in \overline{ABC}\})$$

Definicija 2.29 Neka je \mathcal{H}^2 linearni fazi prostor, \mathcal{L}^2 skup svih *fazi duži* definisanih nad \mathcal{H}^2 i funkcija μ_L fazi skup koji opisuje minimalni element u skupu rastojanja fazi tačaka. Tada funkciju $dist: \mathcal{L}^2 \times \mathcal{L}^2 \rightarrow \mathcal{H}^+$ nazivamo rastojanje između dve fazi duži ako važi sledeće:

$$dist(\overline{AB}, \overline{CD}) = \tilde{d}(\tilde{X}, \tilde{Y})$$

gde je tačka $\tilde{X} \in \overline{AB}$ i $\tilde{Y} \in \overline{CD}$ i

$$\mu_L(\tilde{d}(\tilde{X}, \tilde{Y})) = hgt(\{\tilde{d}(\tilde{Q}, \tilde{W}) | \forall \tilde{Q} \in \overline{AB} \wedge \forall \tilde{W} \in \overline{CD}\})$$

Definicija 2.30 Neka je \mathcal{H}^2 linearni fazi prostor, \mathcal{L}^2 skup svih *fazi duži* definisanih nad \mathcal{H}^2 , \mathcal{T}^2 skup svih *fazi trouglova* i funkcija μ_L fazi skup koji opisuje minimalni element u skupu rastojanja fazi tačaka. Tada funkciju $dist: \mathcal{L}^2 \times \mathcal{T}^2 \rightarrow \mathcal{H}^+$ nazivamo rastojanje između fazi duži i fazi trougla ako važi sledeće:

$$dist(\overline{AB}, \overline{CDE}) = \tilde{d}(\tilde{X}, \tilde{Y})$$

gde je tačka $\tilde{X} \in \overline{AB}$ i $\tilde{Y} \in \overline{CDE}$ i

$$\mu_L(\tilde{d}(\tilde{X}, \tilde{Y})) = hgt(\{\tilde{d}(\tilde{Q}, \tilde{W}) | \forall \tilde{Q} \in \overline{AB} \wedge \forall \tilde{W} \in \overline{CDE}\})$$

Definicija 2.31 Neka je \mathcal{H}^2 linearni fazi prostor, \mathcal{T}^2 skup svih *fazi trouglova* definisanih nad \mathcal{H}^2 i funkcija μ_L je fazi skup koji opisuje minimalni element u skupu rastojanja fazi tačaka. Tada funkciju $dist: \mathcal{T}^2 \times \mathcal{T}^2 \rightarrow \mathcal{H}^+$ nazivamo rastojanje između dva fazi trougla ako važi sledeće:

$$dist(\overline{ABC}, \overline{DEF}) = \tilde{d}(\tilde{X}, \tilde{Y})$$

gde je tačka $\tilde{X} \in \widetilde{ABC}$ i $\tilde{Y} \in \widetilde{DEF}$ i

$$\mu_L(\tilde{d}(\tilde{X}, \tilde{Y})) = hgt(\{\tilde{d}(\tilde{Q}, \tilde{W}) \mid \forall \tilde{Q} \in \widetilde{ABC} \wedge \forall \tilde{W} \in \widetilde{DEF}\}).$$

Prostorne operacije

U ovom odeljku biće date osnovne definicije i osobine prostornih funkcija kao što su presek, unija, fazi transformacije (translacija, rotacija, zamućenje i izoštrenje) i fazi konveksni omotač [60].

Presek i unija predstavljaju osnovne operacije nad skupovima.

Definicija 2.32 Neka je \mathcal{H}^2 linearni fazi prostor definisan nad \mathbb{R}^2 . *Unija* fazi tačaka $\tilde{A} \in \mathcal{H}^2$ i $\tilde{B} \in \mathcal{H}^2$ je fazi skup čija funkcija pripadnosti ima sledeći oblik:

$$\mu_{\tilde{A} \cup \tilde{B}}(X) = \max(\mu_{\tilde{A}}(X), \mu_{\tilde{B}}(X)).$$

Definicija 2.33 Neka je \mathcal{H}^2 linearni fazi prostor definisan nad \mathbb{R}^2 . *Presek* fazi tačaka $\tilde{A} \in \mathcal{H}^2$ i $\tilde{B} \in \mathcal{H}^2$ je fazi skup čija funkcija pripadnosti ima sledeći oblik:

$$\mu_{\tilde{A} \cap \tilde{B}}(X) = \min(\mu_{\tilde{A}}(X), \mu_{\tilde{B}}(X)).$$

Fazi konveksni omotač

Određivanje fazi konveksnog omotača skupa tačkastih objekata predstavlja fundamentalni problem računarske geometrije i sastavni je deo velikog broja algoritama koji se primenjuju u analizi, obradi i zaključivanju. Iz velikog broja primera u kojima se koristi određivanje konveksnog omotača izvodićemo: određivanje dijametra skupa tačaka, rastojanje između dva skupa fazi tačaka, klasterizacija, prepoznavanje oblika i drugo.

U nastavku su date osnovne osobine fazi konveksnog omotača u dvodimenzionalnom linearnom fazi prostoru.

Lema 2 Neka je $\mathcal{A} \subset \mathcal{H}^2$, gde je \mathcal{H}^2 linearni fazi prostor definisan nad \mathbb{R}^2 i $k = |ED(\alpha, \mathcal{A})|$ je broj elemenata skupa $ED(\alpha, \mathcal{A})$, tada je skup $D = \bigcup_{S \in \mathcal{A}} [S]^\alpha$ konveksan za $k = 1$ i nekonveksan za $k > 1$.

Dokaz:

Ako je $k = 1$, tada je skup $D = [S]^\alpha$, i $[S]^\alpha$ krug, iz čega sledi da je skup D konveksan. Ako je $k > 1$, tada u skladu sa definicijom skupa $ED(\alpha, \mathcal{A})$, postoji fazi tačka $\tilde{V} \in ED(\alpha, \mathcal{A})$ takva da njezin α -presek (krug $[V]^\alpha$) sadrži tačke koje ne pripadaju skupu $D^- = \bigcup_{\tilde{S} \in \mathcal{A}, \tilde{S} \neq \tilde{V}} [S]^\alpha$, što je direktna posledica definicije skupa $ED(\alpha, \mathcal{A})$. Ako krug $[V]^\alpha$ ne sadrži ni jednu tačku iz D^- , tada skup D nije konveksan, jer postoji duž koja povezuje tačke iz $[V]^\alpha$ i tačke iz D^- , a duž nije u celosti u skupu D .

Ako krug $[V]^\alpha$ ima zajedničkih tačaka sa skupom D^- , tada postoji krug iz D^- koji ima bar jednu zajedničku rubnu tačku sa rubom kruga $[V]^\alpha$. Tada, postoji duž koja se nalazi na zajedničkoj tangenti ovih krugova i povezuje tačku iz $[V]^\alpha$ i tačku iz D^- . Ova duž ne sadrži tačke koje pripadaju skupu D , iz čega sledi da skup D nije konveksan. ■

Jedna od veoma važnih osobina skupa $ED(\alpha, \mathcal{A})$ opisana je sledećom teoremom.

Teorema 2.16 Neka skup $ED(\alpha, \mathcal{A})$ sadrži više od jednog elementa. Tada svaki luk sa ruba $\partial CH(\alpha, \mathcal{A})$ ima dve susedne duži a svaka duž ima dva susedna luka. Lukovi i duži su susedni ako imaju bar jednu zajedničku tačku.

Dokaz: Neka je skup $D = \bigcup_{\tilde{S} \in \mathcal{A}} [S]^\alpha$ unija svih α -preseka tačaka iz \mathcal{A} i neka je skup $L = \partial CH(\alpha, \mathcal{A}) \cap D$. Po lemi 2.2 skup D nije konveksan, iz čega sledi da $CH(\alpha, \mathcal{A}) \setminus D \neq \emptyset$. Po teoremi 2.1, sve tačke koje ne pripadaju uniji α -preseka a koje pripadaju skupu $\partial CH(\alpha, \mathcal{A})$, nalaze se na duži koja povezuje dve tačke iz L . Kako je $CH(\alpha, \mathcal{A}) \setminus D \neq \emptyset$ imamo i $\partial CH(\alpha, \mathcal{A}) \setminus D \neq \emptyset$ (u suprotnom slučaju skup D bi bio konveksan što nije tačno). Na osnovu prethodnih tvrđenja sledi da postoji najmanje jedna linija na $\partial CH(\alpha, \mathcal{A})$.

Za svaku tačku $\tilde{S} \in ED(\alpha, \mathcal{A})$ postoji bar jedan luk sa centrom u S i radijusom r_S koji pripada $\partial CH(\alpha, \mathcal{A})$. Ovo je direktna posledica definicije skupa $ED(\alpha, \mathcal{A})$. Kako je $|ED(\alpha, \mathcal{A})| > 1$, skup $\partial CH(\alpha, \mathcal{A})$ sadrži najmanje dva različita luka. Ovi lukovi su susedni samo ako se krug kojem pripada jedan luk nalazi unutar drugog kruga, što je u suprotnosti sa definicijom skupa $ED(\alpha, \mathcal{A})$ kao minimalnog skupa tačaka koji formiraju konveksni omotač. Ako lukovi nemaju zajednički centar i radius, tada oni ne pripadaju rubu konveksnog omotača jer postoji duž koja se nalazi na zajedničkoj tangenti oba kruga kojima lukovi pripadaju, povezuje tačke iz

$CH(\alpha, \mathcal{A})$ i isključujući dodirne tačke ne sadrži ni jednu tačku iz $CH(\alpha, \mathcal{A})$. Kao posledica, skup $CH(\alpha, \mathcal{A})$ nije konveksan što je u kontradikciji sa pretpotavkom da je $CH(\alpha, \mathcal{A})$ konveksan omotač. Rub konveksnog omotača je zatvorena kriva. Kako dva luka ne mogu da imaju zajedničke tačke sledi da mora postojati kriva koja ih povezuje a kao posledica teoreme 2.2 ta kriva je duž.

U slučaju da linija povezuje dve tačke sa istog luka, ova linije bi pripadala unutrašnjosti α -preseka fazi tačke na čijem rubu se luk nalazi, što je u suprotnosti sa definicijom ruba. Iz ovoga sledi da duž povezuje različite lukove odnosno da svaka linija ima dva susedna luka i svaki luk ima dve susedne duži. ■

U teoremi 2.17 pokazano je da je minimalno rastojanje unutrašnje tačke i duži manje ili jednako od rastojanja te tačke i lukova susednih datoj duži.

Teorema 2.17 Neka skup $ED(\alpha, \mathcal{A})$ sadrži više od jednog elementa. Tada za svaku tačku $S \in CH(1, \mathcal{A})$ i proizvoljni luk l koji pripada rubu $\partial CH(\alpha, \mathcal{A})$, postoji duž p , takva da važi

$$d(S, p) \leq d(S, l),$$

gde je $d(S, p) = \min(d(S, X) | X \in p)$, $d(S, l) = \min(d(S, Y) | Y \in l)$, p je duž sa $\partial CH(\alpha, \mathcal{A})$, $d(S, p)$ je rastojanje tačke $S \in R^2$ i duži p , $d(S, l)$ je rastojanje tačke $S \in R^2$ i luka l .

Dokaz: Neka je v prava linija koja prolazi kroz tačke S i V . Projekciju proizvoljne tačke Q sa luka l na pravu v označićem sa Q' . Tačke Q, Q', S formiraju pravougli trougao. Tačke Q, Q', V takođe formiraju pravougli trougao. Pretpostavimo da je: $d(Q, Q') = t$, $d(Q'S) = x$, $d(Q, S) = f(x)$, $d(Q', V) = x - a$, $d(Q, V) = (1 - \alpha)r_V$. Na osnovu dva trougla moguće je formirati sledeće dve jednačine:

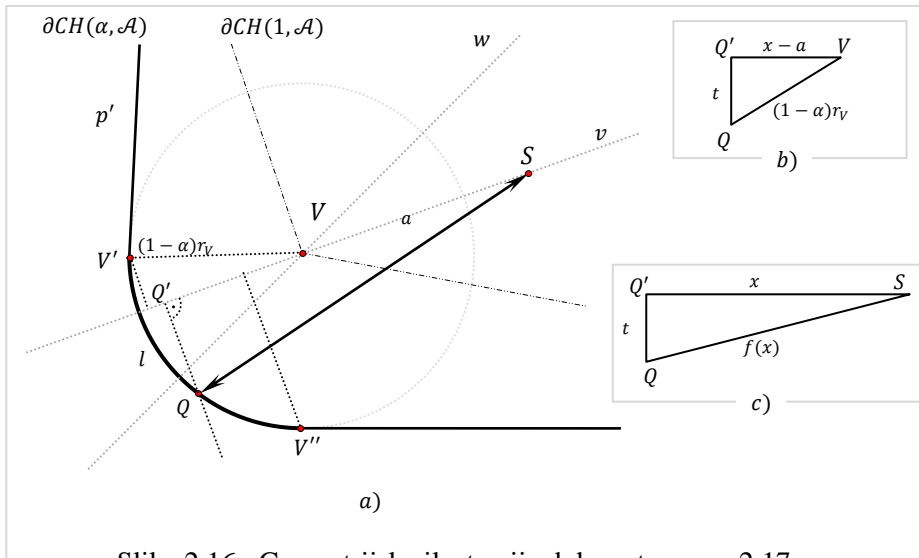
$$t^2 + x^2 = f^2(x),$$

$$t^2 + (x - a)^2 = (1 - \alpha)^2 r_V^2.$$

Iz ovih jednačina, moguće je predstaviti rastojanje $d(Q, S)$ između proizvoljne tačke koja pripada luku l i tačke S kao funkciju $f(x) = \sqrt{(1 - \alpha)^2 r_V^2 - a^2 + 2ax}$. Ova funkcija je monotonno rastuća. Iz čega sledi da je minimum funkcije za minimalno x odnosno na mestu gde je projekcija sa prave najbliža tački S . Označićemo sa w pravu liniju koja prolazi kroz tačku V i deli luk l na dva jednaka dela. Projekciju tačke S na pravu w obeležimo sa S' . Tada tačke S, V i S' formiraju ugao $\gamma = \sphericalangle(S, V, S')$ (pozitivna orijentacija ugla je u smeru suprotnom od smera kazaljki na satu). U zavisnosti od vrednosti ugla razlikujemo tri slučaja:

- a) $\gamma > 0$ rastojanje projekcije tačke V'' na w i tačke S je manje od rastojanja projekcije tačke V' na w i tačke S ,
- b) $\gamma = 0$ projekcije tačaka V'' i V' jednako su udaljene od tačke S , i
- c) rastojanje projekcije tačke V' na w i tačke S je manje od rastojanja projekcije tačke V'' na w i tačke S .

Za slučaj a) tačka koja pripada luku l koja je najbliža tački S je tačka V'' koja istovremeno pripada i duži p . Iz ovoga i na osnovu definicije rastojanja tačke i linije sledi da je $d(S, p) \leq d(S, V'')$ i $d(S, p) \leq d(S, l)$. Za slučaja b) dovoljno je posmatrati tačku V'' čime se dokaz svodi na slučaj pod a). Na kraju, slučaj pod c) u kojem je najbliža tačka V' a, kako po teoremi 2.1, tačka pripada i duži p' , sledi da je $d(S, p') \leq d(S, V')$ odnosno važi $d(S, p') \leq d(S, l)$. ■



Slika 2.16: Geometrijska ilustracija dokaza teoreme 2.17

Teoremom 2.17 je pokazano da se za određivanje rastojanja između unutrašnje tačke S i ruba $\partial CH(\alpha, \mathcal{A})$ ne moraju razmatrati rastojanja do lukova, jer za svaki luk postoji duž koja je bliža posmatranoj unutrašnjoj tački S .

Da bismo dokazali teoremu 2.20 uvodimo sledeće dve teoreme:

Teorema 2.18 Neka skup $ED(\alpha, \mathcal{A})$ sadrži više od jednog elementa. Tada se rastojanje između α -preseka $[S]^\alpha$ unutrašnje tačke $\tilde{S} \in \mathcal{A}^o$ i proizvoljne duži $p \subset \partial CH(\alpha, \mathcal{A})$ određuje po sledećoj formuli:

$$d([S]^\alpha, p) = c_1 \sqrt{1 - c_2^2 (1 - \alpha)^2} + c_4 (1 - \alpha),$$

gde je

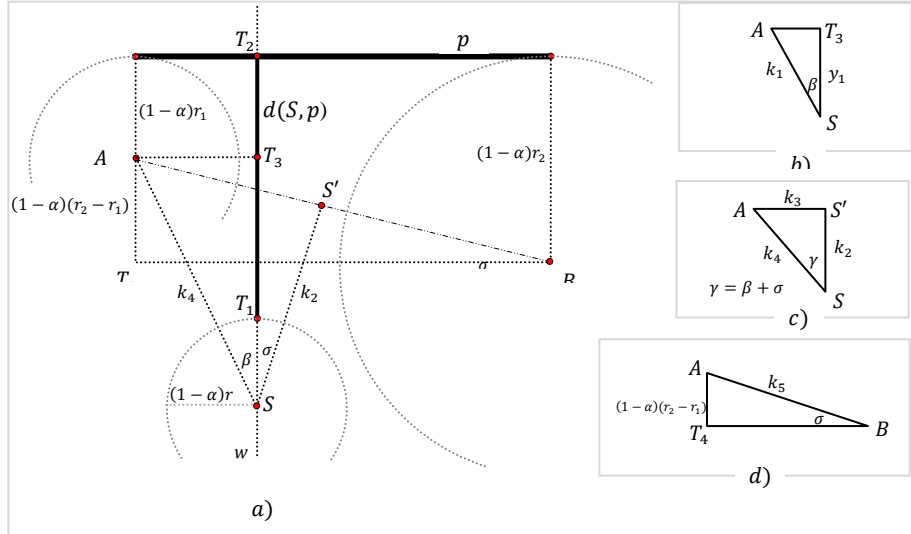
$$c_1 = \frac{k_1 k_2}{k_4}, \quad c_2 = \frac{r_2 - r_1}{k_5}, \quad c_3 = \frac{k_1 k_3}{k_4}, \quad c_4 = (c_3 c_2 + r_1 - r), \quad r_1 = \min \{r_A, r_B\},$$

$$r_2 = \max \{r_A, r_B\},$$

$k_1 = d(A, S)$, $k_2 = d(S, S')$, tačka S' je projekcija tačke S na duž \overline{BA} ,

$k_3 = d(A, S')$,

$k_4 = d(S, A)$ i $k_5 = d(A, B)$

Dokaz:

Slika 2.17: Geometrijska ilustracija dokaza teoreme 2.18

Na slici 2.17 a) prikazana je geometrijska interpretacija dokaza teoreme 2.18. Duž $p \subset \partial CH(\alpha, \mathcal{A})$ povezuje rubove α -preseka tačkaka $\tilde{A} = (A, r_A)$ i $\tilde{B} = (B, r_B)$, koje pripadaju skupu $ED(\alpha, \mathcal{A})$. Rastojanje između α -preseka unutrašnje tačke $\tilde{S} = (S, r)$ i duži p računa se na sledeći način:

$$d([S]^\alpha, p) = d(T_1, T_2) = d(S, T_2) - d(S, T_1) = d(S, T_2) - (1 - \alpha)r \quad (2.5)$$

$$d(S, T_2) = d(S, T_3) + d(T_3, T_2) = y_1 + (1 - \alpha)r_1, \quad (2.6)$$

gde je $y_1 = d(S, T_3)$. Na slici 2.17 b) prikazan je trougao $\triangle(T_3, A, S)$ na osnovu kojeg se dužina y_1 može predstaviti na sledeći način:

$$y_1 = k_1 \cos(\beta), \quad (2.7)$$

gde je $k_1 = d(A, S)$, i ne zavisi od promene parametra α .

Na osnovu uglova γ i σ (slika 2.17 c) ugao β možemo da predstavimo kao:

$$\gamma = \beta + \sigma, \quad \beta = \gamma - \sigma.$$

Pravougli trougao $\triangle(S', A, S)$ ne zavisi od promena parametra α

$$\cos(\gamma) = \frac{k_2}{k_4}, \quad \sin(\gamma) = \frac{k_3}{k_4}, \quad (2.8)$$

gde je dužina k_2, k_3 i k_4 su konstante (nezavisne od parametra α).

Na osnovu osnovnih trigonometrijskih pravila i jednačina (2.8), $\cos(\beta)$ možemo da predstavimo kao:

$$\cos(\beta) = \cos(\gamma - \sigma) = \cos(\gamma) \cos(\sigma) + \sin(\gamma) \sin(\sigma), \quad (2.9)$$

$$\cos(\beta) = \frac{k_2}{k_4} \cos(\sigma) + \frac{k_3}{k_4} \sin(\sigma).$$

Na osnovu pravouglog trougla $\triangle(T_4, B, A)$ (Slika 2.17d), $\sin(\sigma)$ može se predstaviti kao:

$$\sin(\sigma) = \frac{(1-\alpha)(r_2-r_1)}{k_5}, \quad (2.10)$$

$$\cos(\sigma) = \sqrt{1 - \sin(\sigma)^2} = \sqrt{1 - \frac{(1-\alpha)^2(r_2-r_1)^2}{k_5^2}}, \quad (2.11)$$

gde je dužina k_5 konstantna i ne zavisi od parametra α . Ako spojimo jednačine (2.11) i (2.10) sa (2.9) dobijamo:

$$\cos(\beta) = \frac{k_2}{k_4} \sqrt{1 - \frac{(1-\alpha)^2(r_2-r_1)^2}{k_5^2}} + \frac{k_3}{k_4} \frac{(1-\alpha)(r_2-r_1)}{k_5}. \quad (2.12)$$

Ubacivanjem jednačine (2.12) u jednačinu (2.7) dobijamo:

$$y_1 = k_1 \left(\frac{k_2}{k_4} \sqrt{1 - \frac{(1-\alpha)^2(r_2-r_1)^2}{k_5^2}} + \frac{k_3}{k_4} \frac{(1-\alpha)(r_2-r_1)}{k_5} \right),$$

$$y_1 = \frac{k_1 k_2}{k_4} \sqrt{1 - \frac{(1-\alpha)^2(r_2-r_1)^2}{k_5^2}} + \frac{k_1 k_3}{k_4} \frac{(1-\alpha)(r_2-r_1)}{k_5},$$

i ako izvršimo smene $c_1 = \frac{k_1 k_2}{k_4}$, $c_2 = \frac{r_2-r_1}{k_5}$, $c_3 = \frac{k_1 k_3}{k_4}$ dobijamo sledeće:

$$y_1 = c_1 \sqrt{1 - (1-\alpha)^2 c_2^2} + (1-\alpha) c_3 c_2. \quad (2.13)$$

Ako ubacimo jednačinu (2.13) u (2.6) dobijamo sledeći izraz:

$$d(S, T_2) = c_1 \sqrt{1 - (1-\alpha)^2 c_2^2} + (1-\alpha) c_3 c_2 + (1-\alpha) r_1, \quad (2.14)$$

i ubacivanjem jednačine (2.14) u jednačinu (2.5) dobijamo:

$$d([S]^\alpha, p) = c_1 \sqrt{1 - (1-\alpha)^2 c_2^2} + (1-\alpha) c_3 c_2 + (1-\alpha) r_1 - (1-\alpha) r,$$

odnosno

$$d([S]^\alpha, p) = c_1 \sqrt{1 - (1-\alpha)^2 c_2^2} + (1-\alpha)(c_3 c_2 + r_1 - r).$$

Nakon sređivanja izraza dobijamo

$$d([S]^\alpha, p) = c_1 \sqrt{1 - c_2^2 (1-\alpha)^2} + c_4 (1-\alpha),$$

gde je $c_4 = (c_3 c_2 + r_1 - r)$.

Teorema 2.19 Neka je skup $\mathcal{A} \subset \mathcal{H}^2$, i neka je skup $ED(\alpha, \mathcal{A})$ ima više od jednog elementa.

- (i) Tada je rastojanje između unutrašnje tačke $\tilde{S} \in \mathcal{A}^o$ i $\partial CH(\alpha, \mathcal{A})$ dato sledećom formulom

$$\text{dist}(\tilde{S}, \alpha, \mathcal{A}) = \min \{d(S, p) | p \in \partial CH(\alpha, \mathcal{A})\}.$$
- (ii) Funkcija $\text{dist}(\tilde{S}, \alpha, \mathcal{A})$ je konkavna za svaku unutrašnju tačku $\tilde{S} \in \mathcal{A}^o$ i $\alpha \in [0, 1]$.

Dokaz:

- (i) Ova pretpostavka je direktna posledica teoreme 2.17 i teoreme 2.18.
- (ii) Treba pokazati da je funkcija $d([S]^\alpha, p)$ prikazana u teoremi 2.18. konkavna, odnosno da je njen drugi izvod manji od 0.

Ako posmatramo formulu

$$d([S]^\alpha, p) = c_1 \sqrt{1 - c_2^2 (1 - \alpha)^2} + c_4 (1 - \alpha),$$

gde su c_1, c_2, c_3 i c_4 opisani u teoremi 2.18, prvi izvod je:

$$d'([S]^\alpha, p) = \frac{c_1 c_2^2 (1 - \alpha)}{\sqrt{1 - c_2^2 (1 - \alpha)^2}} - c_4.$$

Odnosno drugi izvod:

$$d''([S]^\alpha, p) = \frac{-c_1 c_2^2}{(1 - c_2^2 (1 - \alpha)^2) \sqrt{1 - c_2^2 (1 - \alpha)^2}}.$$

Kako je $c_1 \geq 0$ i $1 - c_2^2 (1 - \alpha)^2 > 0$ sledi da je $d''([S]^\alpha, p) \leq 0$ za svako $0 \leq \alpha \leq 1$. ■

U sledećoj teoremi dokazana je jedna veoma važna osobina α -konveksnog ruba skupa \mathcal{A} .

Teorema 2.20 Za svaki skup \mathcal{A} definisan u linearnom fazi prostoru \mathcal{H}^2 i za svaki $\lambda > 0$ važi sledeće

$$ED(\alpha, \mathcal{A}) = ED(1, \mathcal{A}) \Rightarrow ED(\alpha + \lambda, \mathcal{A}) = ED(1, \mathcal{A}).$$

Dokaz: Pretpostavimo suprotno, tj. da je $ED(\alpha, \mathcal{A}) = ED(1, \mathcal{A}) \Rightarrow ED(\alpha + \lambda, \mathcal{A}) \neq ED(1, \mathcal{A})$. Sledi da postoji tačka $\tilde{S} \in ED(\alpha + \lambda, \mathcal{A})$ takva da $\tilde{S} \notin ED(\alpha, \mathcal{A}), \tilde{S} \notin ED(1, \mathcal{A})$. U teoremi 2.19 je pokazano da rastojanje $dist(\tilde{S}, \alpha, \mathcal{A})$ između unutrašnje tačke \tilde{S} i \mathcal{A} ($dist(\tilde{S}, \alpha, \mathcal{A})$) je konkavna funkcija. Ako pretpostavimo da je $dist(\tilde{S}, 1, \mathcal{A}) > 0$ i $t(\tilde{S}, \alpha, \mathcal{A}) > 0$, dobijamo $dist(\tilde{S}, \alpha + \lambda, \mathcal{A}) > 0$ što je u suprotnosti sa $\tilde{S} \in ED(\alpha, \mathcal{A})$, jer je $dist(\tilde{T}, \alpha, \mathcal{A}) = 0$ za svaku tačku \tilde{T} iz $ED(\alpha, \mathcal{A})$.

U nastavku definišaćemo pojam dijametra nepraznog skupa fazi tačaka.

Definicija 2.34 Neka je \mathcal{H}^n linearni fazi prostor, \tilde{d} je fazi rastojanje definisano u definiciji 2.29, $\mathcal{A} \subset \mathcal{H}^n$ i $|\mathcal{A}| > 1$. Tada je *dijametar skupa* \mathcal{A} fazi skup takav da važi $\tilde{d}(\mathcal{A})$ je maksimalni element (definicija 2.15) skupa $\{\tilde{d}(\tilde{X}, \tilde{Y}) | \tilde{X}, \tilde{Y} \in \mathcal{A}\}$.

Za dokazivanje teoreme 2.21 uvodimo sledeću lemu.

Lema 2.3 Neka su X i Y tačke iz skupa $CH(\alpha, \mathcal{A}) \subset R^2$. Neka je rastojanje između tačaka X i Y jednako dijametru skupa $CH(\alpha, \mathcal{A})$, tada su tačke X i Y rubne tačke.

Dokaz: Pretpostavimo da je tačka X rubna tačka a Y je unutrašnja tačka (postoji krug sa centrom u Y i sadrži sve tačke iz $CH(\alpha, \mathcal{A})$). Tada postoji tačka Z koja se nalazi na pravoj liniji koja prolazi kroz tačke X i Y . Ova tačka pripada rubu kruga koji je u potpunosti unutar skupa $CH(\alpha, \mathcal{A})$ i važi $d(X, Z) > d(X, Y)$. Ovo je u suprotnosti sa pretpostavkom da je rastojanje između tačaka X i Y najveće rastojanje. ■

Na kraju dokazaćemo nekoliko osobina konveksnog ruba jačine m nepraznog skupa fazi tačaka.

Teorema 2.21 Neka je $\mathcal{A} \subset \mathcal{H}^2$ i $\partial\mathcal{A}$ konveksni rub skupa \mathcal{A} jačine m , tada važi sledeće:

- (i) $\tilde{d}(\mathcal{A}) = \tilde{d}(\partial\mathcal{A})$.
- (ii) $hgt(\tilde{d}(\mathcal{A})) > 0$.
- (iii) $hgt(\tilde{d}(\mathcal{A})) \geq m$.

Dokaz:

- (i) Pretpostavimo da su tačke $\tilde{P} \in \mathcal{A}^o$ i $\tilde{T} \in \partial\mathcal{A}$ takve da je $d(P, T)$ najveći element u $\{d(X, Y) | X, Y \in CH(1, \mathcal{A})\}$. Kako je $\tilde{P} \notin \partial\mathcal{A}$, po lemi 2.3 postoji tačka koja pripada rubu $CH(1, \mathcal{A})$ čije rastojanje do tačke T je veće od $d(P, T)$. Na osnovu prethodnog i definicije maksimalnog elementa (definicija 2.15) sledi da $\mu_G(\tilde{d}(\tilde{T}, \tilde{P})) = 0$. Na osnovu $\mu_G(\tilde{d}(\tilde{T}, \tilde{P})) \geq 0$ sledi da tačke $\tilde{T}, \tilde{P} \in \partial\mathcal{A}$ pripadaju rubu, dok za sve ostale tačke važi $\mu_G(\tilde{d}(\tilde{T}, \tilde{P})) = 0$.
- (ii) Neka je $Q = \{\tilde{d}(\tilde{X}, \tilde{Y}) | \tilde{X}, \tilde{Y} \in \partial\mathcal{A}\}$. Ako skup Q sadrži samo jedan element \tilde{P} , tada na osnovu refleksivnosti fazi relacije \leq^{RF} i definicije 2.15 sledi da je $\mu_G(\tilde{P}) = 1$ i $hgt(\tilde{d}(\mathcal{A})) = 1 > 0$. Ako skup Q sadrži više od jednog elementa, tada na osnovu definicije dijametra, i ako je $\tilde{X} \neq \tilde{Y}$, sledi da je jedna od vrednosti $\mu(\tilde{X} \leq^{RF} \tilde{Y}) > 0$ ili $\mu(\tilde{Y} \leq^{RF} \tilde{X}) > 0$, odnosno $hgt(\tilde{d}(\mathcal{A})) > 0$.
- (iii) Neka je $hgt(\tilde{d}(\mathcal{A})) = h > 0$ odnosno postoje tačke $\tilde{X}, \tilde{Y} \in \partial\mathcal{A}$ takve da je $\mu_G(\tilde{d}(\tilde{X}, \tilde{Y})) = h$. Tada tačke \tilde{X} i \tilde{Y} pripadaju i skupu $ED(1, \mathcal{A})$ i skupu $(1 - h, \mathcal{A})$. Na osnovu osobine (i) sledi da skup $ED(1 - (h + \varepsilon), \mathcal{A})$ ne sadrži obe tačke i \tilde{X} i \tilde{Y} jer važi $(1 - (h + \varepsilon), \mathcal{A}) \neq ED(1, \mathcal{A})$, gde je $\varepsilon > 0$. U skladu sa definicijom 2.5, tačke \tilde{X} i \tilde{Y} , koje pripadaju skupu $ED(1, \mathcal{A})$, takođe pripadaju i skupu $ED(1 - m, \mathcal{A})$. Ako pretpostavimo da je $h < m$ to znači da je $1 - m < 1 - (h + \varepsilon) < 1$ što je u suprotnosti sa teoremom 2.20, u kojoj je pokazano da važi
- $$ED(1 - m, \mathcal{A}) = ED(1 - (h + \varepsilon), \mathcal{A}) = ED(1, \mathcal{A}) \blacksquare$$

Algoritam za određivanje maksimalnog rastojanja

Jedan od važnih doprinosa ove disertacije je novi algoritam za određivanje maksimalnog rastojanja u skupu nepreciznih tačaka koji se odlikuje smanjenom računskom složenošću.

Ovaj algoritam odnosi se na određivanje dijametra skupa \mathcal{A} čiji elementi pripadaju linearnom fazi prostoru definisanom nad R^1 i R^2 . Algoritam se bazira na određivanju tačaka \tilde{S}_{min} i \tilde{S}_{max} čije rastojanje predstavlja maksimalni element u skupu svih rastojanja elemenata iz \mathcal{A} . Za tačke \tilde{S}_{min} i \tilde{S}_{max} važi

$$m = \mu_G(\tilde{d}(\tilde{S}_{max}, \tilde{S}_{min})) \geq \max(\mu_G(\tilde{d}(\tilde{S}_i, \tilde{S}_j)) | \forall \tilde{S}_i, \tilde{S}_j \in \mathcal{A}),$$

gde je funkcija μ_G definisana u definiciji 2.15.

Prvi korak algoritma je određivanje rubnih tačaka $\partial\mathcal{A}$ (na primer algoritmom Graham scan [61]). Među tačkama određenim na ovaj način nalaze se i tačke \tilde{S}_{min} i \tilde{S}_{max} (dokazano teoremom 2.21). Zatim se vrednost $m' \cong \mu_G(\tilde{d}(\tilde{S}_{max}, \tilde{S}_{min}))$ određuje metodom polovljenja.

U nastavku je dat algoritam pseudo kodom:

Funkcija: $fuzzyDiameter(\mathcal{A}, \varepsilon, \tilde{D}_M, m')$

Ulaz: Skup $\mathcal{A} = \{\tilde{S}_1, \dots, \tilde{S}_n\}$ sadrži elemente linearnog fazi prostora definisanog definicijom 2.8 ; ε - zahtevana tačnost za računanje m .

Izlaz: $\tilde{D}_M = \tilde{d}(\tilde{S}_{min}, \tilde{S}_{max})$ dijametar skupa \mathcal{A} , $m' \cong m = \mu_G(\tilde{d}(\tilde{S}_{max}, \tilde{S}_{min}))$, funkcija μ_G definisana je u definiciji 2.15.

Begin

- Odrediti skup $\partial\mathcal{A} = ED(1, \mathcal{A})$; ovaj skup sadrži tačke \tilde{S}_{min} i \tilde{S}_{max} .
 - Ako je skup $\mathcal{A} \subset \mathcal{H}^1$ tada se $\partial\mathcal{A}$ određuje trivijalno, a za slučaj $\mathcal{A} \subset \mathcal{H}^2$ skup $\partial\mathcal{A}$ određuje Graham scan algoritmom.
- Odrediti skup svih međusobnih rastojanja tačaka koje pripadaju $\partial\mathcal{A}$ $\mathcal{D} = \{\tilde{d}(\tilde{S}_i, \tilde{S}_j) | \forall \tilde{S}_i, \tilde{S}_j \in \partial\mathcal{A}\} = \{\tilde{D}_1, \dots, \tilde{D}_k\}$
- Odrediti element iz skupa \mathcal{D} sa najvećim jezgrom. Ova vrednost odgovara jezgru elementa $\tilde{D}_M = \tilde{d}(\tilde{S}_{min}, \tilde{S}_{max})$
- Odrediti vrednost $m' = mPower(\tilde{D}_M, \mathcal{D}, \varepsilon)$

End

Funkcija $mPower(\tilde{D}_M, \mathcal{D}, \varepsilon)$

Ulaz:

Fazi tačka \tilde{D}_M ; element sa najvećim jezgrom u \mathcal{D}

Skup \mathcal{D} ; sadrži sva rastojanja između rubnih tačaka

Realan broj ε ; zahtevana tačnost vrednosti $\mu_G(\tilde{d}(\tilde{S}_{max}, \tilde{S}_{min}))$

Izlaz:

Realan broj m' ; broj između 0 i 1

Begin

$\tilde{\mathcal{K}} = edge(1, \mathcal{D})$

$t = \log_2(\frac{1}{\varepsilon})$

$\alpha = 0.5, \Delta\alpha = 0.5$

for $i = 1..t$

if ($edge(\alpha, \mathcal{D}) == \tilde{\mathcal{K}}$)

$\alpha = \alpha - \Delta\alpha$

else

$\alpha = \alpha + \Delta\alpha$

end

$\Delta\alpha = \Delta\alpha/2$

$m' = 1 - \alpha$

End

Funkcija $edge(\alpha, \mathcal{D})$

Ulaz:

Realan broj α ; vrednost između 0 i 1

Skup \mathcal{D} ; skup svih rastojanja između rubnih tačaka

Izlaz:

Fazi tačka $\tilde{\mathcal{K}}$; α -presek tačke $\tilde{\mathcal{K}}$ sadrži maksimalnu tačku u uniji α -preseka svih tačaka iz \mathcal{D}

Begin

$\tilde{\mathcal{K}} = \tilde{D}_1$

For each D_i in \mathcal{D}

If $D_i + (1 - \alpha)r_{D_i} > \mathcal{K}_i + (1 - \alpha)r_{\mathcal{K}_i}$ then

$\tilde{\mathcal{K}} = \tilde{D}_i$

End

Računska složenost algoritma. Za računanje vrednosti $\mu_G(\tilde{d}(\tilde{S}_i, \tilde{S}_j))$ neophodno je izvršiti poređenje rastojanja $\tilde{d}(\tilde{S}_i, \tilde{S}_j)$ sa svim ostalim rastojanjima elemenata iz skupa \mathcal{A} . Broj ovakvih rastojanja je $t = p(p - 1)/2 \approx p^2$ gde je p broj elemenata iz skupa \mathcal{A} .

Kompleksnost procedure za poređenje svih rastojanja u slučaju fazi tačaka je

$$f_M(p) = t + t \cdot (t - 1)/2 \in O(p^4).$$

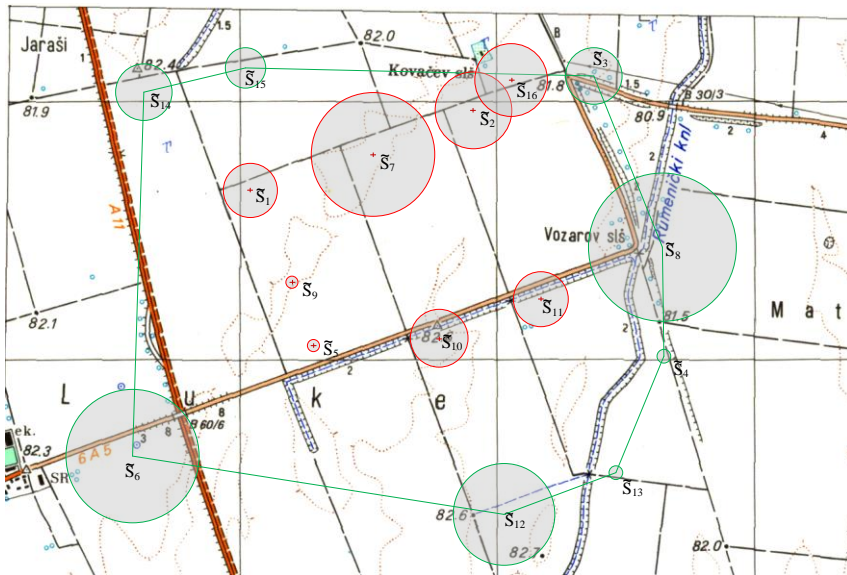
iz čega sledi da je računaska složenost ovakve procedure $O(p^4)$. U predstavljenom algoritmu za određivanje tačaka \tilde{S}_{min} i \tilde{S}_{max} i vrednosti $\mu_G(\tilde{d}(\tilde{S}_{max}, \tilde{S}_{min}))$, koristi se osobina $\tilde{d}(\mathcal{A}) = \tilde{d}(\partial\mathcal{A})$ dokazana u teoremi 2.21. U slučaju da je broj elemenata skupa $\partial\mathcal{A}$ k , $k \leq p$. Tada ako se traži tačna vrednost za m , računaska kompleksnost računanja dijametra $\tilde{d}(\mathcal{A})$ je $f_C(p, k) = p \cdot \log(p) + f_M(k) \in O(k^4)$.

Ako umesto određivanja tačne vrednosti m , određujemo njenu približnu vrednost m' , sa dozvoljenim odstupanjem ε , $|m' - m| \leq \varepsilon$, tada se računaska složenost računa po sledećoj formuli

$$f_A(p, k, \varepsilon) = p \cdot \log(p) + k \cdot (k - 1)/2 \cdot \log(1/\varepsilon) \in O(k^2).$$

Primer

U primeru je prikazana primena predstavljenog modela nepreciznih tačkastih objekata. Pozicije 16 prostornih objekata dobijene su GPS merenjem (različitim uređajima i različitim preciznostima). Tačke su prikazane na mapi na slici 2.18 a odgovarajuće fazi tačke date su u tabeli 2.1. Nepreciznost podatka predstavljena je odgovarajućim krugom (slika)/vrednosti radiusa (tabela).



Slika 2.18: GPS tačke

Fazi tačka	x	y	r
\tilde{S}_1	420.3	291.95	45.33
\tilde{S}_2	768.7	171.05	63.01
\tilde{S}_3	957.2	108.65	35.8
\tilde{S}_4	1063.8	545.45	10
\tilde{S}_5	520	530.5	10
\tilde{S}_6	244	706.5	109.69
\tilde{S}_7	615.3	242.55	91.04
\tilde{S}_8	1063.8	382.95	115.2
\tilde{S}_9	492	455.5	10
\tilde{S}_{10}	711.5	512.95	46.82
\tilde{S}_{11}	872.7	451.85	45.5
\tilde{S}_{12}	816.8	797.65	75.41
\tilde{S}_{13}	994.9	731.35	10
\tilde{S}_{14}	256	140.5	46.01
\tilde{S}_{15}	415	101.5	34.13
\tilde{S}_{16}	828	118.5	59.03

Tabela 2.1: Fazi tačke

U prvom koraku algoritma odredi se skup $\partial\mathcal{A} = \{\tilde{S}_3, \tilde{S}_4, \tilde{S}_8, \tilde{S}_{12}, \tilde{S}_{13}, \tilde{S}_{14}, \tilde{S}_{15}\}$ pomoću Graham scan algoritma.

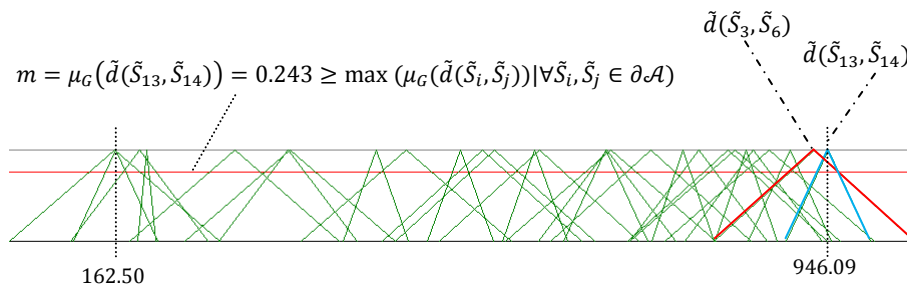
Sledeći korak je određivanje svih rastojanja između tačaka koje pripadaju skupu $\partial\mathcal{A}$. U tabeli 2.2 prikazana su fazi rastojanja kao uređeni parovi rastojanja i odgovarajućih mera nepreciznosti (vrednost poluprečnika).

	\tilde{S}_4	\tilde{S}_6	\tilde{S}_8	\tilde{S}_{12}	\tilde{S}_{13}	\tilde{S}_{14}	\tilde{S}_{15}
\tilde{S}_3	449.6,35.8	930.6,109.6	294.2,115.2	703.1,75.4	623.8,35.8	701.9,46.0	542.2,35.8
\tilde{S}_4		835.4,109.6	162.5,115.2	353.0,75.4	198.2,10.0	903.6,46.0	786.1,34.1
\tilde{S}_6			881.3,115.2	580.0,109.6	751.3,109.6	566.1,109.6	628.7,109.6
\tilde{S}_8				482.6,115.2	355.1,115.2	843.4,115.2	707.2,115.2
\tilde{S}_{12}					190.0,75.4	863.9,75.4	803.7,75.4
\tilde{S}_{13}						946.0,46.0	856.1,34.1
\tilde{S}_{14}							163.7,46.0

Tabela 2.2 Fazi rastojanja tačaka iz $\partial\mathcal{A}$

Maksimalni element u skupu svih rastojanja je $\tilde{d}(\tilde{S}_{13}, \tilde{S}_{14}) = (946.09, 46.01)$ sa vrednošću $\mu_G(\tilde{d}(\tilde{S}_{13}, \tilde{S}_{14})) = 0.243$.

Geometrijska ilustracija procedure za računanje prikazana je na slici 2.19.



Slika 2.19: Fazi rastojanja tačaka iz $\partial\mathcal{A}$

Računarska složenost algoritma za različit broj elemenata skupa \mathcal{A} i skupa $\partial\mathcal{A}$ prikazan je u tabeli 2.3, gde je p broj elemenata skupa \mathcal{A} , k je broj elemenata skupa $\partial\mathcal{A}$, a ε je dozvoljeno odstupanje od tačne vrednosti m .

	$p = 16, k = 8, \varepsilon = 0.001$	$p = 32, k = 16, \varepsilon = 0.001$
$f_M(p)$	7,260	123,256
$f_C(p, k)$	470	7,420
$f_A(p, k, \varepsilon)$	343	1,356

Tabela 2.3 Računska složenost algoritma

Translacija, rotacija, zamućenje, izošćenje

Definicija 2.35 Neka je \mathcal{H}^2 linearni fazi prostor definisan nad \mathbb{R}^2 funkciju $fMove: \mathcal{H}^2 \times \mathbb{R}^2 \rightarrow \mathcal{H}^2$ nazivamo *translacija* fazi tačke ako važi sledeće:

$$fMove(\tilde{A}, T) = (A + T, r_A)$$

ili

$$\mu_{fMove(\tilde{A}, T)}(X) = \max\left(0, 1 - \frac{d(X, A + T)}{|r_A|}\right)$$

Definicija 2.36 Neka je \mathcal{H}^2 linearni fazi prostor definisan nad \mathbb{R}^2 , funkcija $rot: \mathbb{R}^2 \times [0, \pi] \rightarrow \mathbb{R}^2$ je rotacija vektora za dati ugao. Tada funkciju $fRot: \mathcal{H}^2 \times [0, \pi] \rightarrow \mathcal{H}^2$ nazivamo *rotacija* fazi tačke za dati ugao ako važi sledeće:

$$fRot(\tilde{A}, \beta) = (rot(A, \beta), r_A)$$

ili

$$\mu_{fRot(\tilde{A}, \beta)}(X) = \max\left(0, 1 - \frac{d(X, rot(A, \beta))}{|r_A|}\right)$$

Definicija 2.37 Neka je \mathcal{H}^2 linearni fazi prostor definisan nad \mathbb{R}^2 . Tada funkciju $fBlur: \mathcal{H}^2 \times [1, \infty) \rightarrow \mathcal{H}^2$ nazivamo *zamućenje* fazi tačke ako važi sledeće:

$$fBlur(\tilde{A}, t) = (A, t \cdot r_A)$$

ili

$$\mu_{fBlur(\tilde{A}, t)}(X) = \max\left(0, 1 - \frac{d(X, A)}{t \cdot |r_A|}\right) = (1, t) \cdot \tilde{A}$$

Napomena: prilikom *zamućenja* menja se poluprečnik kupe a jezgro fazi tačke ostaje na istom mestu.

Definicija 2.38 Neka je \mathcal{H}^2 linearni fazi prostor definisan nad \mathbb{R}^2 . Tada funkciju $fSharp: \mathcal{H}^2 \times [1, \infty) \rightarrow \mathcal{H}^2$ nazivamo *izoštrenje* fazi tačke ako važi sledeće:

$$fSharp(\tilde{A}, t) = \left(A, \frac{r_A}{t}\right) = \left(1, \frac{1}{t}\right) \cdot \tilde{A}$$

ili

$$\mu_{fSharp(\tilde{A}, t)}(X) = \max\left(0, 1 - t \cdot \frac{d(X, A)}{|r_A|}\right)$$

Napomena: Ako se pusti da $t \rightarrow \infty$ $\mu_{fSharp(\tilde{A}, \infty)}(X) = 1$ samo akko je $X = A$. Odnosno dobijamo klasičnu geometrijsku tačku.

Prostorne relacije

Prostorne relacije (predikati) su grupa funkcija koji se koriste za uspostavljanje međusobnih relacija između fazi geometrijskih objekata. U ovu grupu funkcija spadaju: podudarnosti (coincidence), pripadnosti (contain) i orijentacije (kolinearnost, leva ili desna orijentacija). U nastavku će biti prikazane ove relacije.

Podudarnost (koincidencija)

Ova relacija izražava meru sa kojom se dve fazi tačke nalaze na istoj poziciji.

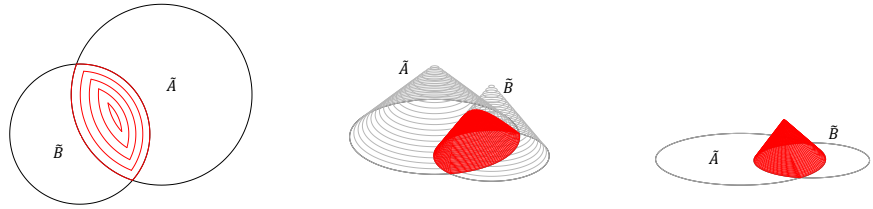
Definicija 2.39 Neka je m mera skupa i \mathcal{H}^2 linearni fazi prostor. Tada fazi relaciju *coin*: $\mathcal{H}^2 \times \mathcal{H}^2 \rightarrow [0, 1]$ nazivamo *fazi podudarnost* ako je funkcija pripadnosti definisana na sledeći način

$$\mu_{coin}(\tilde{A}, \tilde{B}) = m\{\alpha \mid [\tilde{A}]^\alpha \cap [\tilde{B}]^\alpha \neq \emptyset\}$$

Napomena: Ovu definiciju možemo u slučaju da je mera skupa m razlika između najveće i najmanje vrednosti napisati i kao:

$$\mu_{coin}(\tilde{A}, \tilde{B}) = \max\{\alpha \mid [\tilde{A}]^\alpha \cap [\tilde{B}]^\alpha \neq \emptyset\}$$

Geometrijska ilustracija Definicije 2.39 data ja na slici 2.20.



Slika 2.20: Geometrijska ilustracija Definicije 2.39

Način računanja funkcije pripadnosti za fazi relaciju *coin* dat je sledećom teoremom.

Teorema 2.22 Neka je mera skupa $m = \max$. Tada se vrednost funkcije pripadnosti fazi relacije *coin* računa na sleći način:

$$\mu_{\text{coin}}(\tilde{A}, \tilde{B}) = \begin{cases} 0 & \text{ako je } |a_r| + |b_r| = 0 \wedge d(A, B) \neq 0 \\ \max\left(0, 1 - \frac{d(A, B)}{|a_r| + |b_r|}\right) & \text{ako je } |a_r| + |b_r| \neq 0 \\ 1 & \text{ako je } |a_r| + |b_r| = 0 \wedge d(A, B) = 0 \end{cases}$$

Dokaz

Tražimo najveće α za koje $[\tilde{A}]^\alpha \cap [\tilde{B}]^\alpha \neq \emptyset$. U teoremi 2.3 je pokazano da za elemente linearnog fazi prostora važi $[\tilde{A}]^{\alpha_2} \subset [\tilde{A}]^{\alpha_1}$ akko $\alpha_2 > \alpha_1$, a iz osobine skupova da je $Q_1 \subset Q_2 \wedge T_1 \subset T_2 \rightarrow (Q_1 \cap T_1) \subset (Q_2 \cap T_2)$. Kako tražimo najveći α sledi da se on dobija za najmanji neprazan skup. Najmanji neprazan skup je skup sa samo jednim elementom a kako taj element istovremeno treba da pripada i skupu $[\tilde{A}]^\alpha$ i $[\tilde{B}]^\alpha$. $T \in [\tilde{A}]^\alpha \wedge T \in [\tilde{B}]^\alpha$. Kako je $d(T, A) \leq (1 - \alpha_m)|r_A|$ i $d(T, B) \leq (1 - \alpha_m)|r_B|$ a pošto je T jedina tačka sledi da se tačka T nalazi na dodiru dva kruga, odnosno $d(T, A) + d(T, B) = d(A, B)$. Nakon sređivanja dobijemo tri jednačine:

$$d(T, A) = (1 - \alpha_m)|r_A|$$

$$d(T, B) = (1 - \alpha_m)|r_B|$$

$$d(T, A) + d(T, B) = d(A, B)$$

Rešenje ovog sistema po α_m daje rezultat $\alpha_m = 1 - \frac{d(A,B)}{|a_r|+|b_r|}$, odnosno $\alpha_m = \max\left(0, 1 - \frac{d(A,B)}{|a_r|+|b_r|}\right)$ jer α_m ne može da bude manje od 0. U slučaju da je $|a_r| + |b_r| = 0$ sledi da su tačke \tilde{A} i \tilde{B} precizne tačke koje su podudarne samo ako je njihovo međusobno rastojanje jednako 0 odnosno konačno dobijamo

$$\mu_{coin}(\tilde{A}, \tilde{B}) = \begin{cases} 0 & \text{ako je } |a_r| + |b_r| = 0 \wedge d(A, B) \neq 0 \\ \max\left(0, 1 - \frac{d(A,B)}{|a_r|+|b_r|}\right) & \text{ako je } |a_r| + |b_r| \neq 0 \\ 1 & \text{ako je } |a_r| + |b_r| = 0 \wedge d(A, B) = 0 \end{cases} .$$

■

Pripadnost (contain)

Ova relacija predstavlja meru sa kojom fazi tačka pripada fazi duži.

Definicija 2.40 Neka je m mera skupa $[0,1]$, \mathcal{H}^2 linearni fazi prostor i \mathcal{L}^2 skup svih fazi duži definisanih nad \mathcal{H}^2 . Tada fazi relaciju *contain*: $\mathcal{H}^2 \times \mathcal{L}^2 \rightarrow [0,1]$ nazivamo *fazi pripadnost* ako je funkcija pripadnosti definisana na sledeći način

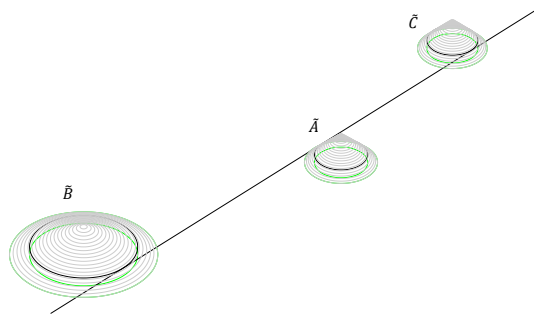
$$\mu_{contain}(\tilde{A}, \tilde{BC}) = m\{\alpha \mid [\tilde{A}]^\alpha \cap [\tilde{BC}]^\alpha \neq \emptyset\}$$

Ista definicija može se napisati i malo drugačije:

$$\mu_{contain}(\tilde{A}, \tilde{BC}) = m\{\alpha \mid \exists u \in [0,1] \wedge \exists X \in [\tilde{A}]^\alpha \wedge \exists Y, Z \in [\tilde{BC}]^\alpha \\ \wedge X = Y + u(Z - Y)\}$$

Kažemo da se fazi tačka \tilde{A} nalazi između fazi tačaka \tilde{B} i \tilde{C} u meri $\mu_{contain}(\tilde{A}, \tilde{BC})$.

Geometrijska ilustracija Definicije prikazana je na slici 2.21.

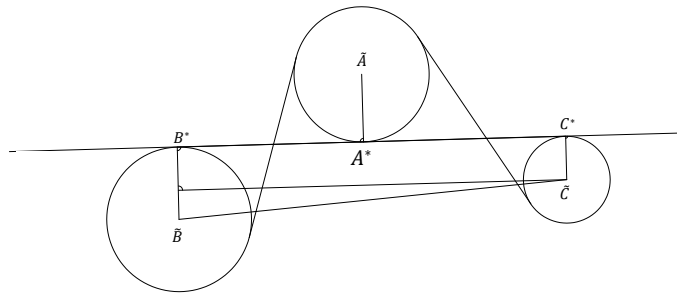
Slika 2.21: Ilustracija fazi relacije *contain*

Način određivanja funkcije pripadnosti za ovu relaciju daje Teorema 2.23 čija je geometrijska ilustracija prikazana na slici 2.22.

Teorema 2.23 Neka su $\tilde{A}, \tilde{B}, \tilde{C} \in \mathcal{H}^2$, a $u \in [0,1]$ i \tilde{A}' karakteristična vrednost slike i fazi slika tačke A na fazi duži $\tilde{A}\tilde{B}$ respektivno. T_{AB} i T_{AC} su interni homotetički centri fazi tačaka \tilde{A} i \tilde{B} i tačaka \tilde{A} i \tilde{C} . Tada meru da se fazi tačka \tilde{A} nalazi između fazi tačaka \tilde{B} i \tilde{C} računamo na sledeći način:

$$\mu_{\text{contain}}(\tilde{A}, \tilde{B}\tilde{C}) = \begin{cases} \mu_{\text{coin}}(\tilde{A}, \tilde{A}') & \text{ako je } u \in \{0,1\} \\ \mu_{\tilde{A}}(A^*) & \text{ako je } u \in (0,1) \end{cases}$$

gde je tačka A^* projekcija jezgra fazi tačke \tilde{A} na pravu koja prilazi kroz T_{AB} i T_{AC} .



Slika 2.22: Geometrijska ilustracija teoreme 2.23

Dokaz:

Slično kao i u prethodnoj teoremi tražimo α za koji je

$card([\tilde{A}]^\alpha \cap [\widetilde{BC}]^\alpha) = 1$ odnosno dva alfa preseka sadrže samo jednu zajedničku tačku. Ta tačka je na mestu gde tangenta alfa preseka tačkaka \tilde{B} i \tilde{C} dodiruje alfa presek tačke \tilde{A} , odnosno tražimo α takvo da postoji zajednička tangenta tri kružnice. To je najveće α jer ako bi postojalo veće $\alpha^* > \alpha$ sledilo bi $\emptyset \neq [\tilde{A}]^{\alpha^*} \cap [\widetilde{BC}]^{\alpha^*} \subset [\tilde{A}]^\alpha \cap [\widetilde{BC}]^\alpha$ što je nemoguće jer je $[\tilde{A}]^\alpha \cap [\widetilde{BC}]^\alpha$ najmanji neprazan skup.

Na osnovu definicije internog homotetičkog centra sledi da sve interne tangente između $[\tilde{A}]^\alpha$ i $[\tilde{B}]^\alpha$ prolaze kroz tačku T_{AB} , a sve interne tangente između $[\tilde{A}]^\alpha$ i $[\tilde{C}]^\alpha$ prolaze kroz tačku T_{AC} . Iz ovoga sledi da postoji α za koji je prava koja prolazi kroz T_{AB} i T_{AC} zajednička tangenta za $[\tilde{A}]^\alpha$, $[\tilde{B}]^\alpha$ i $[\tilde{C}]^\alpha$. Poluprečnik kruga $[\tilde{A}]^\alpha$ je $(1 - \alpha)a_r$ iz čega sledi da je rastojanje $d(A, A^*) = (1 - \alpha)|a_r|$ odnosno $= 1 - \frac{d(A, A^*)}{|a_r|}$. Kako α ne može da bude manje od 0 sledi da je $\alpha = \max\left(0, 1 - \frac{d(A, A^*)}{|a_r|}\right)$ što je po definiciji

$$\mu_{\tilde{A}}(A^*) = \alpha = \max\left(0, 1 - \frac{d(A, A^*)}{|a_r|}\right)$$

ako je $u = 0$ sledi da je tačka \tilde{B} fazi slika tačke A na fazi duži \widetilde{BC} , i tada je $[\tilde{A}]^\alpha \cap [\widetilde{BC}]^\alpha = [\tilde{A}]^\alpha \cap [\tilde{B}]^\alpha$ odnosno $\mu_{contain}(\tilde{A}, \widetilde{BC}) = \max\{\alpha \mid [\tilde{A}]^\alpha \cap [\tilde{B}]^\alpha \neq \emptyset\} = \mu_{coin}(\tilde{A}, \tilde{B})$. Analogno se dokazuje slučaj kad je $u = 1$. ■

Fazi kolinearnost

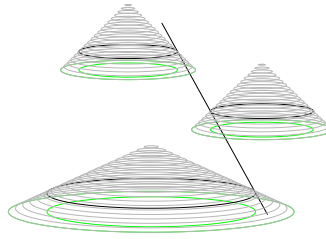
Kolinearnost je osobina koja karakteriše najmanje tri tačke koje pripadaju istoj pravoj. U slučaju da umesto klasičnih tačkaka analiziramo fazi tačke potrebno je koncept kolinearnosti posebno definisati. U nastavku sledi definicija *fazi kolinearnosti* i metod za njezino računanje.

Definicija 2.41 Fazi relaciju *coli*: $\mathcal{H}^2 \times \mathcal{H}^2 \times \mathcal{H}^2 \rightarrow [0, 1]$ nazivamo *fazi kolinearnost* između tri fazi tačke ako je funkcija pripadnosti

$$\mu_{coli}(\tilde{A}, \tilde{B}, \tilde{C}) = m\{\alpha \mid \exists u \in R \wedge \exists X \in [\tilde{A}]^\alpha \wedge \exists Y \in [\tilde{B}]^\alpha \wedge \exists Z \in [\tilde{C}]^\alpha \wedge A = B + u(C - B)\}$$

gde je funkcija m mera skupa.

Geometrijska ilustracija Definicije prikazana je na slici 2.23 a način računanja funkcije priodnosti daje Teorema 2.24.



Slika 2.23: Ilustracija Definicije 2.41

Teorema 2.24 Neka su fazi tačke $\tilde{A}, \tilde{B}, \tilde{C} \in \mathcal{H}^2$ a funkcija *coli* fazi kolinearnost. Tada se funkcija priadnosti fazi relacije računa na sledeći način:

$$\mu_{coli}(\tilde{A}, \tilde{B}, \tilde{C}) = \max\left(\mu_{contain}(\tilde{A}, \widetilde{BC}), \mu_{contain}(\tilde{B}, \widetilde{AC}), \mu_{contain}(\tilde{C}, \widetilde{AB})\right)$$

Dokaz: Alfa preseki tri fazi tačke su tri kruga. Ponovo kao i u prethodnom slučaju tražimo najveće α za koje postoji prava koja prolazi kroz tačke koje pripadaju α presecima a to je slučaj u kojem jedan od alfa preseka dodiruje alfa presek fazi duži određene preostalim dvema tačkama. ■

Modeliranje procesa

Sistem je skup entiteta združenih u interakciji sa ciljem ostvarivanja neke funkcionalnosti.

Ako prostor stanja sistema uključuje i geoprostorne osobine kažemo da je sistem prostorno determinisan.

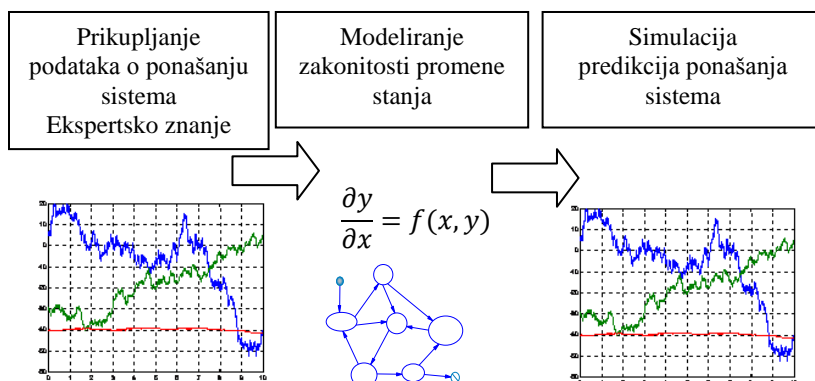
Proces je promena stanja u prostoru stanja sistema koji najčešće uključuje i vremensku komponentu.

U kontekstu predmeta ovoga rada (softverski simulacioni sistem) proces je jedan scenario izvršavanja programa.

Tehnike koje se najčešće koriste za modeliranje geoprostornih procesa su:

1. Diferencijalne jednačine
2. Automati prelaza stanja - celularni automati
3. Deklarativne metode - skupovi pravila
4. Opšte namenski jezici za modelovanje (na primer, UML)

Na slici 2.24 prikazan je tok u modeliranju procesa i kasnijoj simulaciji ponašanja sistema.



Slika 2.24: Tok modeliranja i simulacije procesa

Diferencijalne jednačine su jedan od najopštijih i najčešće korišćenih načina za predstavljanje, odnosno simulaciju prostornih procesa. U ovom slučaju moguće je koristiti poznate sisteme za rešavanje sistema diferencijalnih jednačina. Ovakav pristup jeste matematički najdosledniji međutim u praksi je veoma teško primenljiv. Prvo je neophodno odrediti zakonitost promene u obliku diferencijalnih jednačina a potom sistem

diferencijalnih jednačina koji je najčešće nelinearan je potrebno rešiti. Broj objekata od kojih se sastoji sistem proporcionalan je broju jednačina sistema diferencijalnih jednačina. Zato, ovaj pristup nije pogodan za sisteme koji se sastoje od velikog broja objekata, jer je za simulaciju potrebno rešiti sistem velikog broja diferencijalnih jednačina koji ne mora da bude linearan.

Neka se stanje sistem u vremenskom trenutku t može opisati vektorom $X = \{x_1, x_2, \dots, x_n\}$. Ponašanje sistema u opštem slučaju opisujemo skupom parcijalnih diferencijalnih jednačina:

$$f_i \left(x_1, x_2, \dots, x_n, t, \frac{\partial x_i}{\partial t}, \frac{\partial x_i}{\partial x_1}, \dots, \frac{\partial x_i}{\partial x_n} \right) = 0, i \in \{1, \dots, n\}$$

Ovaj skup opisuje složene prostorno vremenski zavisne procese.

U praksi se češće posmatraju vremenski procesi kod kojih funkcije imaju sledeći oblik

$$f_i \left(x_1, x_2, \dots, x_n, t, \frac{\partial x_i}{\partial t} \right) = 0, i \in \{1, \dots, n\}$$

ili prostorni

$$f_i \left(x_1, x_2, \dots, x_n, \frac{\partial x_i}{\partial x_1}, \dots, \frac{\partial x_i}{\partial x_n} \right) = 0, i \in \{1, \dots, n\}$$

Konačnim automatima ponašanje sistema opisujemo tako što za svaki pojedinačni objekat ili grupu objekata definišemo skup mogućih stanja i prelaze stanja. Ovaj način je pogodan za simulaciju i relativno je jednostavan za formiranje modela procesa na osnovu prikupljenih podataka. Pogodan je u slučaju konačnog broja mogućih stanja.

Kada modelujemo realan objekat svako pojedinačno stanje predstavljamo skupom mogućih vrednosti i zamenjujemo ga određenom oznakom, najčešće rednim brojem. Promene vrednosti atributa objekata tada pratimo samo u odnosu na promene stanja a u kontekstu analize procesa beležimo oznake tih stanja. Umesto klasičnog skupa mogućih vrednosti pogodnije je svako pojedinačno stanje opisati pomoću fazi skupa. U ovom slučaju objekat se u jednom trenutku može istovremeno naći u više od jednog stanja ali sa različitim merama pripadnosti, zato je potrebno sačuvati ne samo oznaku stanja već i meru sa kojom se objekat nalazi u datom stanju. U nastavku ovog odeljka date su osnovne definicije i ilustrativni primeri primeri modelovanja konačnim automatima u uslovima nedodređenosti.

Definicija 2.42 Fazi konačni automat prelaza stanja FFA je uređena sedmorka $(S, Q, I, X, \mu, \sigma, \eta)$ gde je :

- (i) S neprazan skup (*univerzalni skup stanja*),
- (ii) $Q = \{q_1, q_2, \dots, q_n\}$ neprazan skup fazi skupova (*fazi stanja*) definisanih nad univerzalnim skupom S
($\forall q \in Q) q: S \rightarrow [0,1]$),
- (iii) I neprazan skup (*univerzalni skup ulaznih simbola*),
- (iv) $X = \{x_1, x_2, \dots, x_m\}$ neprazan skup fazi skupova (*ulazni simboli*) definisanih nad skupom I ($\forall x \in X) x: I \rightarrow [0,1]$),
- (v) $\mu: Q \times Q \times X \rightarrow [0,1]$ fazi tranziciona funkcija,
- (vi) $\sigma: Q \rightarrow [0,1]$ fazi skup definisan nad Q (*inicijalno stanje*),
- (vii) $\eta: Q \rightarrow [0,1]$ fazi skup definisan nad Q (*finalno stanje*).

Za $s \in S$ i $q \in Q$ vrednost $q(s)$ predstavlja meru u kojoj se automat nalazi u stanju q . Stepeni pripadnosti $q_s = Q(s) = \{q_1(s), q_2(s), \dots, q_n(s)\} \in [0,1]^n$ predstavlja fazi vektor stanja automata s .

Za $q, p \in Q$ i $x \in X$ vrednost $\mu(q, p, x) \in [0,1]$ je mera sa kojom će automat iz stanja q za ulazni simbol x preći u stanje p . Za $q \in Q$, vrednost $\sigma(q)$ je mera sa kojom je stanje q inicijalno stanje, a $\eta(q)$ predstavlja meru da je stanje q finalno stanje.

Definicija 2.43 Neka su $q, p \in [0,1]^n$ fazi vektori stanja FFA. Tada operator $=^F: [0,1]^n \times [0,1]^n \rightarrow [0,1]$ definišemo na sledeći način:

$$q =^F p =_{DF} 1 - \frac{1}{n} \sum_{i=1}^n |q_i - p_i|$$

Ovaj operator predstavlja fazi relaciju sličnosti između fazi stanja.

Definicija 2.44 Neka je $a \in [0,1]$, $b \in [0,1]^n$ fazi vektor stanja FFA i T jedna t-norma. Tada operator $*$: $[0,1] \times [0,1]^n \rightarrow [0,1]^n$

$$a * b = a * (b_1, b_2, \dots, b_n) =_{DF} (T(a, b_1), T(a, b_2), \dots, T(a, b_n))$$

nazivamo *operator fazi implikacije*.

Operator fazi implikacije koristi se prilikom evaluacije fazi pravila $p \rightarrow q$ gde su p i q fazi izrazi. Jedna od osnovnih osobina fazi implikacije je da ako sa $\tau(p)$ i $\tau(q)$ obeležimo tačnost fazi izraza p i q redom vredi $\tau(p) \leq \tau(q)$.

Primer: Neka je dato fazi pravilo f_1 : if $(A =^F s_1$ and $B =^F s_2)$ then $(A$ is $s_3)$

fazi pravilo može da se napiše i kao $f_1: (A =^F s_1 \text{ and } B =^F s_2) \rightarrow (A \text{ is } s_3)$

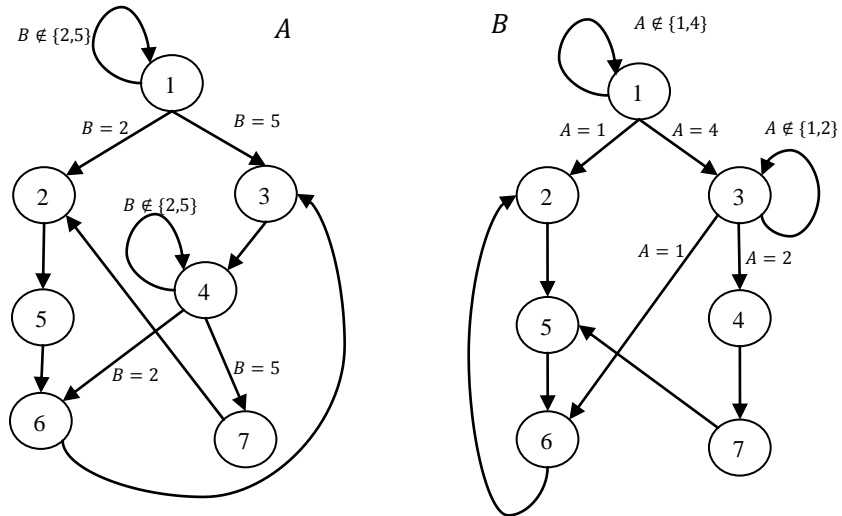
Pravilo se sastoji od pretpostavke $p: (A =^F s_1 \text{ and } B =^F s_2)$ i zaključaka $q: (A \text{ is } s_3)$. Proces evaluacije fazi pravila izvodi se tako što se prvo utvrdi tačnost pretpostavke $\tau(p) = a$ nakon čega se primeni operator fazi implikacije i izvrši dodeljivanje vrednosti $A \text{ is } a * s_3$.

Definicija 2.45 Neka je \hat{P} konačan skup fazi konačnih automata prelaza stanja (FFA) u nastavku ćemo ih zvati *fazi ćelije*, $\lambda: \hat{P} \times \hat{P} \rightarrow [0,1]$ je fazi relacija povezanosti fazi ćelija, $*$ je operator fazi implikacije i $P = (S_P, Q_P, X_P, \mu_P, \sigma_P, \eta_P) \in \hat{P}$. Tada uređeni par (\hat{P}, λ) nazivamo fazi celularni automat (FCA) ako

$$\forall P \in \hat{P} \rightarrow X_P = \bigcup_{T \in \hat{P}, T \neq P} (\lambda(P, T) * Q_T).$$

Skup ulaznih simbola X_P ćelije P definišemo kao uniju fazi stanja povezanih ćelija kada se nad njihovim stanjem primeni operator fazi implikacije uticaja povezanih ćelija na ćeliju P .

Na slici 2.25 je prikazan jedan primer modela ponašanja dva objekta. Prelazi između stanja opisani su logičkim iskazima u kojima učestvuju stanja preostalih objekata od kojih je sastavljen sistem. Simulacija u ovom slučaju predstavlja sekvencijalni niz uzastopnih prelaza stanja svakog pojedinačnog objekta.



Slika 2.25 Dijagram prelaza stanja automata A i B

Skup pravila koji određuju prelaze između stanja objekata A i B dat je tabelom 2.4

	IF	THEN
$k1$	$A_i = 1 \ \& \ B_i = 2$	$A_{i+1} = 2$
$k2$	$A_i = 1 \ \& \ B_i = 5$	$A_{i+1} = 3$
$k3$	$A_i = 1 \ \& \ B_i \notin \{2,5\}$	$A_{i+1} = 1$
$k4$	$A_i = 2$	$A_{i+1} = 5$
$k5$	$A_i = 3$	$A_{i+1} = 4$
$k6$	$A_i = 4 \ \& \ B_i = 2$	$A_{i+1} = 6$
$k7$	$A_i = 4 \ \& \ B_i = 5$	$A_{i+1} = 7$
$k8$	$A_i = 4 \ \& \ B_i \notin \{2,5\}$	$A_{i+1} = 4$
$k9$	$A_i = 5$	$A_{i+1} = 6$
$k10$	$A_i = 6$	$A_{i+1} = 3$
$k11$	$A_i = 7$	$A_{i+1} = 2$
$k12$	$B_i = 1 \ \& \ A_i = 1$	$B_{i+1} = 2$
$k13$	$B_i = 1 \ \& \ A_i = 4$	$B_{i+1} = 3$
$k14$	$B_i = 1 \ \& \ A_i \notin \{1,4\}$	$B_{i+1} = 1$
$k15$	$B_i = 2$	$B_{i+1} = 5$
$k16$	$B_i = 3 \ \& \ A_i = 1$	$B_{i+1} = 6$
$k17$	$B_i = 3 \ \& \ A_i = 2$	$B_{i+1} = 4$
$k18$	$B_i = 3 \ \& \ A_i \notin \{1,2\}$	$B_{i+1} = 3$
$k19$	$B_i = 4$	$B_{i+1} = 7$
$k20$	$B_i = 5$	$B_{i+1} = 6$
$k21$	$B_i = 6$	$B_{i+1} = 2$
$k22$	$B_i = 7$	$B_{i+1} = 5$

Tabela 2.4 Skupa pravila kojima su određeni prelazi stanja objekata A i B

Ovako definisan model sistema za početna stanja $A = 1$ i $B = 5$ daje procese predstavljene tabelom 2.5

i	A							B						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
1	1											1		
2			1											1
3				1				1						
4						1						1		
5			1											1
6				1				1						
7						1						1		
8			1											1
9				1				1						
10						1						1		

Tabela 2.5 Procesi promene stanja objekata A i B

Za ovako definisan model i data početna stanja objekti prolaze *sigurno* kroz data stanja. Ako ne možemo sa sigurnošću da tvrdimo u kojem je

stanju objekat A i B tada umesto klasičnih pravila koristimo fazi pravila i dobijamo ponašanje prikazano tabelom 2.6

i	A							B						
	1	2	3	4	5	6	7	1	2	3	4	5	6	7
1	0.8	0.2									0.1	0.8	0.1	
2	0.17		0.67		0.17				0.1				0.8	0.1
3	0.15	0.08		0.62		0.15			0.89			0.11		
4	0.11	0.11	0.11	0.11	0.06	0.44	0.06					0.89	0.11	
5	0.11	0.06	0.44	0.11	0.11	0.06	0.11		0.11					0.89
6	0.13	0.13	0.06	0.5	0.06	0.13			0.89			0.11		
7	0.11	0.11	0.11	0.11	0.11	0.42	0.05					0.89	0.11	
8	0.11	0.05	0.42	0.11	0.11	0.11	0.11		0.11				0.89	
9	0.12	0.12	0.12	0.47	0.06	0.12			0.89			0.11		
10	0.11	0.11	0.11	0.11	0.11	0.42	0.05					0.89	0.11	

Tabela 2.6 Procesi promene stanja nepreciznih objekata A i B

Specijalan slučaj ovog pristupa je *cellular automaton* u kojem se prostor podeli na konačan broj ćelija u koje su raspoređeni objekti a skupom pravila definišu se moguće promene stanja koje najčešće zavise od stanja same ćelije i stanja susednih ćelija.

Ponašanje sistema možemo opisati **deklarativno** pomoću skupa pravila. Za razliku od prethodnog slučaja, deklarativnim pristupom moguće je opisati složena pravila za opis promene stanja. Ovaj pristup je pogodan za simulaciju, međutim izvođenje pravila na osnovu prikupljenih podataka je relativno teško. U slučaju da postoji ekspertsko znanje o ponašanju sistema ono se može koristiti umesto prikupljanja podataka. U radu [62] opisano je proširenje interpretara XProlog fazi unifikacijom koja može da se koristi za opisivanje pravila prelaza stanja. Primer opisan tabelom 2.4. može se u fazi XProlog proširenju predstaviti sledećim programom.

```

stanje(s1, [1, 0, 0, 0, 0, 0, 0]).
stanje(s2, [0, 1, 0, 0, 0, 0, 0]).
stanje(s3, [0, 0, 1, 0, 0, 0, 0]).
stanje(s4, [0, 0, 0, 1, 0, 0, 0]).
stanje(s5, [0, 0, 0, 0, 1, 0, 0]).
stanje(s6, [0, 0, 0, 0, 0, 1, 0]).
stanje(s7, [0, 0, 0, 0, 0, 0, 1]).

prelazA(A, Anext, B):-stanje(s1, X1), stanje(s2, X2),
    A=X1, B=X2, Anext is X2.
prelazA(A, Anext, B):-stanje(s1, X1), stanje(s3, X3),
    stanje(s5, X5)
    A=X1, B=X5, Anext is X3.
prelazA(A, Anext, B):-stanje(s1, X1), stanje(s2, X2),

```

```

stanje(s5, X5)
A=X1, B!=X5, B!=X2, Anext is X1.
....
simulacija(PA, PB, 0, Rez, Rez) .
simulacija(PA, PB, T, Z, Rez) :-prelazA(PA, NA, PB) ,
                                prelazB(PB, NB, PA) ,
                                T1 is T-1,
                                Z1 is [(T, NA, NB) | Z],
                                simulacija(NA, NB, T1, Z1, Rez) .

simulacija(A, B, T, Rez) :-simulacija(A, B, T, [(T, A, B)], Rez) .

```

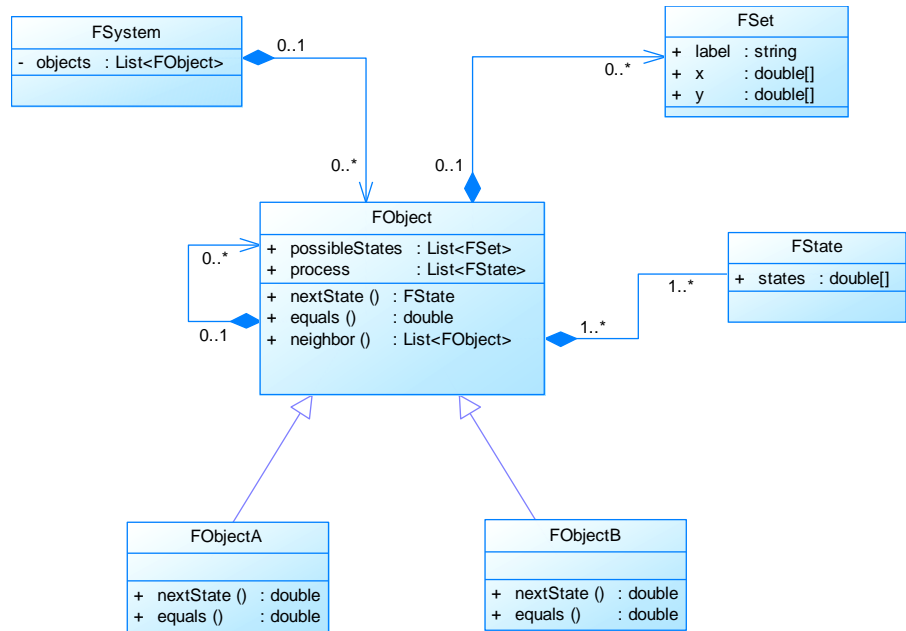
Primer pokretanja simulacije:

```
?-simulacija([0.8,0.2,0,0,0,0],[0,0,0,0.1,0.8,0.1,0],10,Rez) .
```

Rezultat ove simulacije je sadržaj prethodno prikazane Tabele 2.3.

Pristup koji koristi **opšte namenske jezike za modelovanje** je takođe pogodan u slučaju ekspertskeg znanja i najjednostavniji je za simulaciju pošto ga je moguće primeniti direktno na virtualnoj mašini bez nekog posebnog posrednika. Ovaj način nije pogodan ako posedujemo samo prikupljene podatke o sistemu. Kod proceduralnog načina modelovanja procesa promene stanja objekata sistema koristimo funkcije fazi geometrije kao što su rotacija, translacija, izošćenje, zamućenje itd. Za modelovanje i specifikaciju proceduralnog načina opisivanja geoprstornih procesa izabran je UML jezik da bi u implementaciji bilo moguće koristiti bilo koji od raspoloživih objektnih jezika.

Predloženim modelom moguće je opisati fazi proces kao automat prelaza stanja, celularni automat ili proceduralno. Osnova za kreiranje procesa su klase *FSystem*, *FObject* i *FState*. Klasa *FSystem* reprezentuje model sistema čije ponašanje se simulira i sadrži skup objekata klase *FObject*. Klasa *FObject* predstavlja klasu koja sadrži definicije svakog pojedinačnog stanja u obliku fazi skupova, trenutno stanje (mere pripadnosti za svaki pojedinačni fazi skup) i prethodne promene stanja objekta. Jedna od najznačajnijih metoda klase *FObject* je operator (\Rightarrow) poređenja dva stanja koji koristimo da utvrdimo u kojoj meri su dva stanja ista odnosno slična. Ovaj operator se koristi u funkciji prelaza stanja *nextState* u kojoj fazi pravilima definišu prelazi između stanja. Na slici 2.23 prikazan je dijagram klase za modelovanje ovako opisanih fazi procesa. Klase *FObjectA* i *FObjectB* su konkretne implementacije ponašanja dva različita objekta.



Slika 2.26: Dijagram osnovnih klasa za modeliranje fazi procesa

Simulacija ponašanja procesa vrši se uzastopnim pozivanjem metode *nextState* odgovarajuće klase. U ovoj metodi izračunava se prelaz u sledeće stanje na osnovu prethodnih stanja i na osnovu stanja susednih objekata.

Dakle na raspolaganju imamo četiri različita načina za opisivanje procesa koje je moguće koristiti u različitim situacijama. U nastavku je data arhitektura proširivog sistema koji je u mogućnosti da podrži sva četiri načina modeliranja prostornih procesa u uslovima neodređenosti.

Softverska arhitektura i prototip sistema

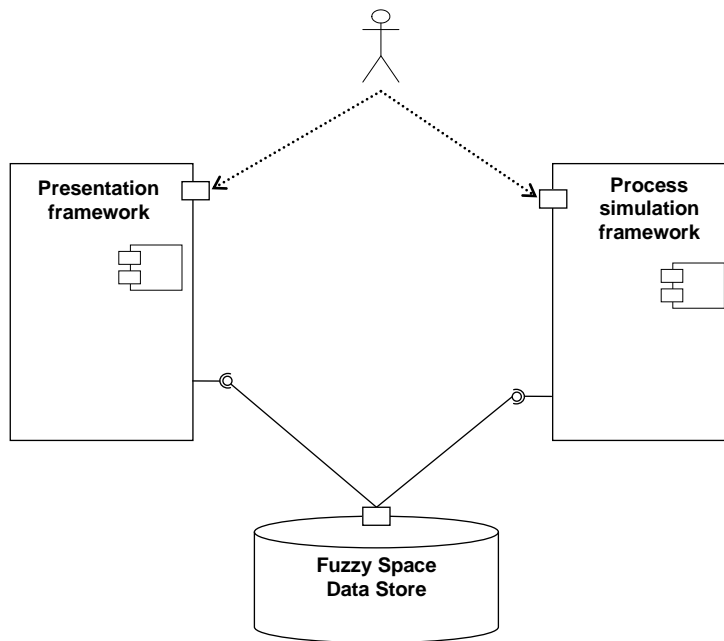
U ovom poglavlju prikazana je softverska arhitektura sistema i implementacija prototipa.

Softverska arhitektura sistema

Specifikacija softverske arhitekture proširivog simulacionog sistema za upravljanje geoprostornim procesima u uslovima neodređenosti data je UML jezikom. Sistem obuhvata osnovne podsysteme:

- skladištenje nepreciznih geoprostornih podataka,
- podsistem za simulaciju i izvršavanje geoprostornih procesa i
- podsistem za prikaz stanja geoprostornog procesa.

Na slici 3.1 prikazan je funkcionalni izgled arhitekture proširivog simulacionog sistema.

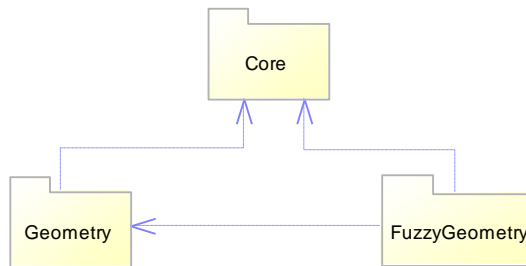


Slika 3.1: Funkcionalni izgled arhitekture proširivog simulacionog sistema

Podsistem za skladištenje podataka

Podsistem za skladištenje je prikazan kao jedna celina (Fuzzy Space Data Store) koja u sebi sadrži proizvoljan broj nezavisnih sistema za upravljanje geoprostornim podacima. Ovi sistemi treba da obezbede podatke koji će se koristiti u procesima, i da omoguće proširivost sa aspekta novih prostornih funkcija i novih prostornih tipova podataka. U ovom radu kao osnova za implementaciju podsistema za skladištenja izabran je PostgreSQL sistem za rukovanje podacima. PostgreSQL je razvijen kao proširiv sistem u kojem se nove funkcije i tipovi podataka jednostavno dodaju. Prostorno proširenje PostgreSQL je PostGIS, koji se sastoji od više prostornih tipova i preko 700 prostornih funkcija. U radu [63] je opisano proširenje PostGIS-a u kojem je implementirana podrška radu sa nepreciznim prostornim podacima. Ovim podacima moguće je pristupiti na dva standardna načina za pristup relacionim podacima JDBC i ODBC čime je omogućeno da programska okruženja u kojima će se koristiti ovakvi podaci budu nezavisna od sistema za skladištenje. Takođe, omogućeno je da klijentski programi budu napisani na različitim platformama i distribuirani na više računara.

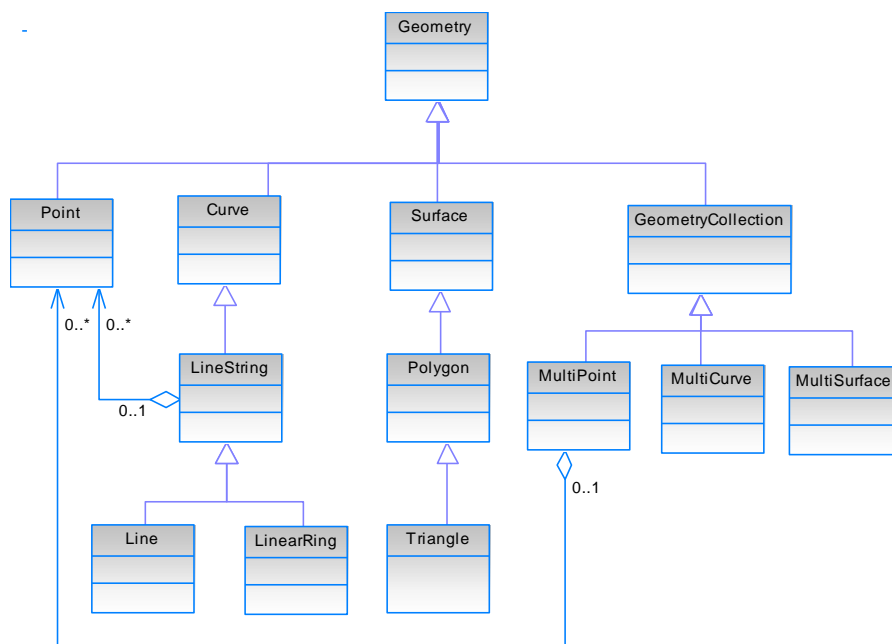
Neprecizni prostorni podaci kao i osnovne funkcije modelirani su kao proširenje PostGIS sistema za rukovanje geoprostornim podacima (vidi [63]) i kao biblioteke klasa u programskim jezicima C# i Java. Ovim pristupom omogućeno je korišćenje osnovnih elemenata i funkcija u definisanju složenih geoprostornih procesa kao Java ili C# klase tako da se oni izvršavaju u sva tri dela sistema (prezentacionom, simulacionom i delu za skladištenje podataka). Na slici 3.2 prikazan je dijagram osnovnih paketa.



Slika 3.2: Dijagram povezanosti osnovnih paketa

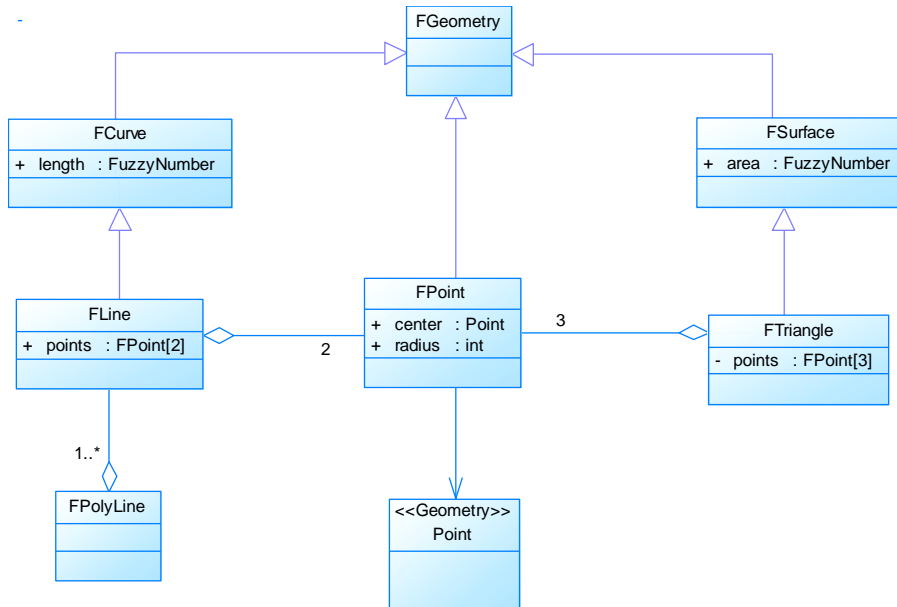
Kako su C# i Java programski jezici potpuno objektno orijentisani implementacija *.java* i *.net* biblioteka izvršena je u skladu sa prikazanim UML modelom.

Paket *Core* između ostalih sadrži klase *SpatialReferenceSystem* i *Matrix*. Prva klasa je zadužena za različite transformacije koordinatnih i referentnih prostornih sistema. Klasa *Matrix* enkapsulira matricu i standardne matrice operacije. Dijagram klasa paketa *Geometry* prikazana je na slici 3.3. Paket *Geometry* sadrži osnovne klasične geometrijske objekte kao i osnovne operacije poput prostornih merenja, transformacije i prostornih relacija.



Slika 3.3: Dijagram povezanosti klasa paketa *Geometry*

U paketu *FuzzyGeometry* implementiran je objektni model nepreciznih geometrijskih objekata opisanih u prethodnom poglavlju.



Slika 3.4: Dijagram klasa paketa *FuzzyGeometry*

Na slici 3.4 je prikazan dijagram klasa paketa *FuzzyGeometry*. Osnovna klasa je *FPoint* koja predstavlja implementaciju matematičkog modela predstavljenog definicijom 2.1.

Prezentacioni podsistem

Prikaz geoprostornih podataka predstavlja veoma kompleksan zadatak jer treba da omogući integraciju podataka iz više izvora, njihov prikaz često zavisi od posmatranog konteksta i međusobnih odnosa samih objekata. Realni objekti obično imaju kompleksan 3D izgled i potrebno je omogućiti jednostavnu manipulaciju između više različitih pogleda i različitih nivoa detaljnosti kao i vremenskih promena stanja posmatranih objekata. Prilikom izbora elemenata podsistema za prikaz podataka potrebno je voditi računa o primeni i odnosu cene i potrebnih funkcija. U ovom radu korišteno je i analizirano nekoliko postojećih rešenja koje ćemo posmatrati u odnosu na sledeće: dostupnost izvornog koda, proširivost, podrška za prikaz podataka i podrška za izmenu podataka. U nastavku je dat pregled osnovnih osobina sledećih programskih paketa: UDIG, Google Earth, Web Open Layer, AutoCAD

UDIG (User-friendly Desktop Internet GIS) je GIS okruženje razvijeno za Eclipse razvojno okruženje i predstavlja jednostavno grafičko okruženje za prikaz i manipulaciju skoro svim vrstama standardnih GIS podataka. Programski paket je proširiv i portabilan na različite programske i računarske platforme, a razvijen je kao sistem otvorenog koda. Jedan od ozbiljnih nedostataka sistema je nedostatak kvalitetne tehničke dokumentacije. Na primer, osnova samog paketa sastoji se od više hiljada klasa za koje ne postoji tehnička dokumentacija, ili ako i postoji ona je veoma skromna. Drugi problem je što ne postoji sistemsko rešenje za proširenje sistema, odnosno za dodavanje novih funkcija.

Google Earth programski paket namenjen je samo za prikaz geoprostornih podataka sačuvanih u KML ili KMZ formatu koji je u velikom broju slučajeva podržan od sistema za rukovanje prostornim podacima. Prikaz detalja kao i navigacija je veoma jednostavna i brza. Ovaj sistem nije razvijen tako da omogući proširenje novim funkcijama i nije razvijen kao sistem otvorenog koda. Osnovna verzija je besplatna i ne zahteva posebnu licencu za korišćenje. Nije omogućena programska izmena stilova za prikazivanje pojedinačnih elemenata.

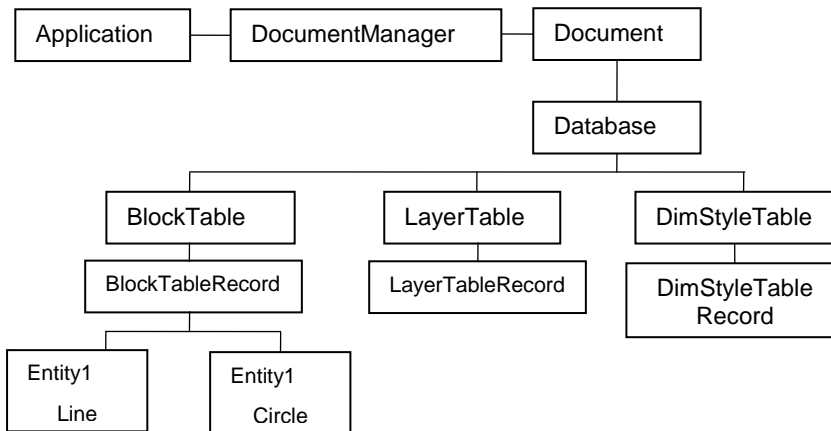
OpenLayer je programski paket ili biblioteka klasa kojom se omogućuje prikaz dinamičkih geoprostornih podataka na web stranici. Razvijen je u skladu sa BSD licencom, otvorenog je koda, besplatan je i poseduje interfejs klase preko kojih je moguća integracija u bogatije web bazirane geoprostorne aplikacije.

AutoCAD je jedan od vodećih i najčešće korištenih softverskih paketa za projektovanje i dizajn. Manipulacija prostornim pa i geoprostornim objektima je veoma jednostavna a sistemskim rešenjem za proširenje osnovnih funkcija razvijeno je nekoliko specijalizovanih programskih paketa poput Autodesk - AutoCAD Civil, AutoCAD Architecture, AutoCAD Inventor itd. Razvoj novih funkcija omogućen je u nekoliko programskih okruženja (Lisp, C++, C#, Visual Basic).

Pristupom koji podržava AutoCAD moguće je implementirati funkcije ili procese i vršiti simulacije u kojima je neophodna intenzivna interakcija između procesa i operatora. Na primer, proces vektorizacije rasterskih mapa je u priličnoj meri zavisano od interakcije sa operaterom.

AutoCAD platforma se na ovaj način pokazala i kao deo prezentacionog podsistema i kao deo simulacionog podsistema, a kako poseduje deo u kojem je moguće skladištiti veliku količinu prostornih podataka, i kao deo podsistema za skladištenje podataka.

Za implementaciju proširenja koriste se *AutoCAD.Net API* i *ObjectDBX.Net*. Na slici 3.5 prikazana je hijerarhija osnovnih objekata biblioteke *AutoCAD.Net API*.



Slika 3.5: Dijagram povezanosti osnovnih objekata *AutoCAD.Net API*

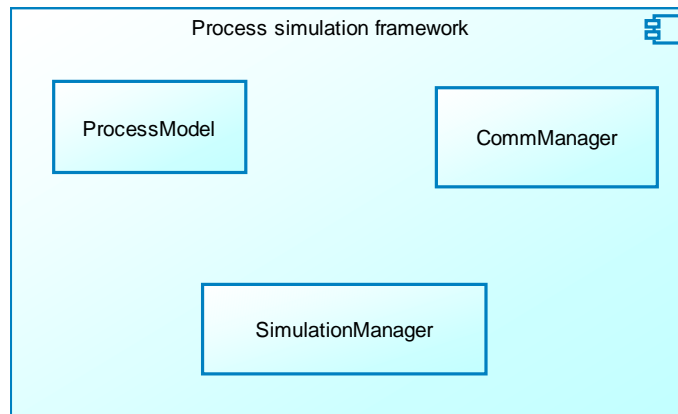
AutoCAD.Net API i *ObjectDBX.Net* distribuiraju se zajedno sa osnovnim programskim paketom u obliku dve dinamičke biblioteke klasa *acmgd.dll* i *acdbmgd.dll*. Proširenje se implementira kao dinamička biblioteka klasa. Učitava se komandom *netload* a komandom *installComponent* se aplikacija instalira tako da se svaki sledeći put automatski učita. Ovako implementirana proširenja mogu da pristupaju sistemu za rukovanje geoprostornim podacima direktno preko ODBC standarda ili indirektno preko *geoservera* korištenjem WFS i ostalih OpenGIS standarda (vidi [64]).

Zbog svojih osobina, AutoCAD je izabran kao jedno od rešenja za implementaciju prototipa. Za implementaciju je izabran programski jezik C# zbog svoje objektno orijentisane prirode a razvojno okruženje Visual Studio 2009. zbog jednostavnosti implementacije.

Podsistem za simulaciju procesa

Podsistem za simulaciju geoprostornih procesa (Slika 3.6) sastoji se iz tri dela: jezgro sistema (*ProcessModel*) u kojem su implementirani objekti za podršku modeliranju procesa, podrška za komunikaciju sa podsistemom za skladištenje geoprostornih podataka (*CommManager*) i deo u kojem se procesi izvršavaju i u kojem se upravlja izvršavanjem procesa

(*SimulationManager*). Podsystem za simulaciju implementiran je tako da omogućuje proširenja u nekoliko pravaca: jednostavno dodavanje novih procesa, konfigurabilnost načina definisanja procesa i konfigurabilnost načina opisivanja procesa.



Slika 3.6: Komponenta sistema za upravljanje simulacijama

U radu [65] implementirano je web bazirano okruženje za rad sa prostornim podacima i aplikacijama. Proširenjem predloženog rešenja delom za modeliranje procesa i delom za pristup podacima (rad [64]) dobijamo proširiv sistem za simulaciju ponašanja geoprostornih sistema.

Implementacija prototipa

Prototip proširivog simulacionog sistema implementiran je u skladu sa modelima predstavljenim u prethodnom poglavlju. Osnovna karakteristika implementiranog prototipa je proširivost i interoperabilnost sa drugim sistemima za rad sa geoprostornim podacima.

Komponente sistema razvijene su tako da se mogu izvršavati na različitim programskim platformama u distribuiranom okruženju. Distribuirano okruženje u ovom slučaju znači da u simulaciji istovremeno može da učestvuje više procesa pod uslovom da se izvršavaju na računarima koji su međusobno povezani TCP/IP protokolom i da razmenjuju podatke u skladu sa OpenGis standardima.

Implementirani proširivi simulacioni sistem sastoji se od sledećih podsistema (Slika 3.7).

Biblioteka osnovnih geometrijski tipova podataka. U ovoj biblioteci su klase kojima se opisuju neprecizni prostorni podaci. Implementirane su klase (Java, C# i PL/SQL implementacija) za sledeće tipove:

- Fazi tačka
- Fazi duž
- Fazi trougao

Biblioteka funkcije za analizu i obradu nepreciznih prostornih podataka. U ovoj biblioteci su klase u kojima su implementirane prostorne operacije, funkcije i relacije definisane u prethodnom poglavlju. Implementirane su klase u sledećim programskim jezicima: Java, C# i PL/SQL, a obuhvataju sledeće operacije, funkcije o relacije:

- rastojanja
- koincidencija, kolinearnosti i preklapanja
- izoštrjenje, zamućenje, rotacija, translacija
- konveksni omotač, voronoi dijagrami, delaunay triangulacija

Biblioteka klase za modeliranje procesa u skladu sa tri načina reprezentacije i izvršavanje simulacije (Java i C#).

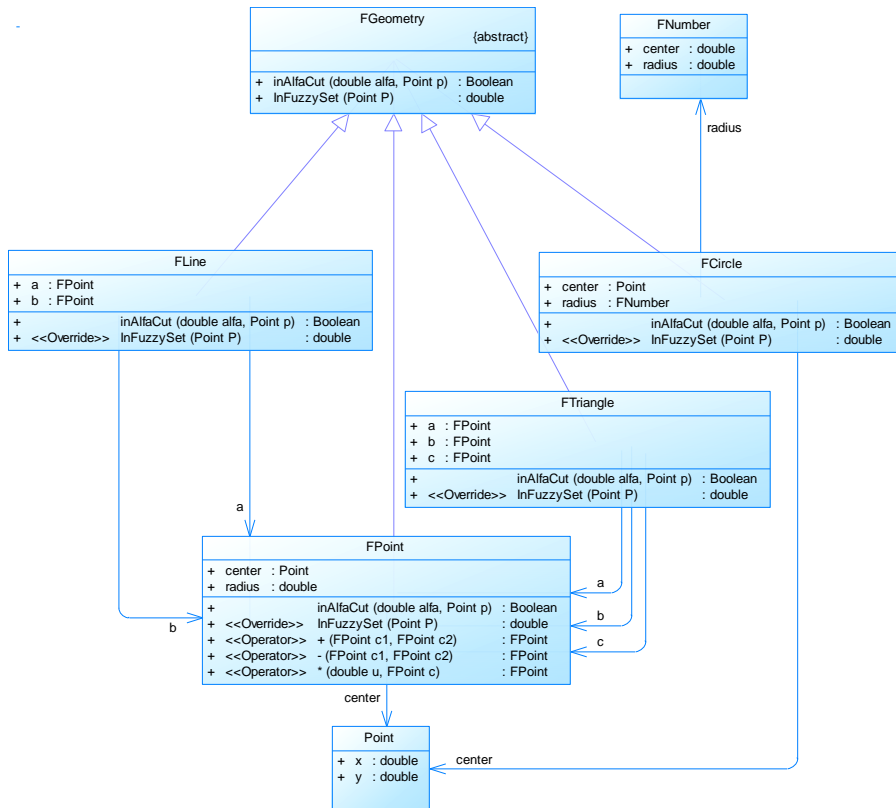
Proširenje PostGIS sistema za upravljanje geoprostornim podacima delom koji omogućuje rad sa nepreciznim prostornim podacima.

Proširenje AutoCAD programskog paketa bibliotekama za podršku prikazu nepreciznih geoprostornih podataka, modeliranju i simulaciji izvršavanja geoprostornih procesa. (C#)

Biblioteka osnovnih tipova i funkcija

U ovom odeljku prikazana je biblioteka osnovnih tipova i funkcija. Modeliranje je izvršeno korišćenjem UML notacije a objektna implementacija u programskim jezicima Java i C# izvršena je u skladu sa modelima opisanim u prethodnom poglavlju.

Na slici 3.8 je prikazan dijagram osnovnih klasa paketa *FGeometryCore*.



Slika 3.8: Dijagram klasa paketa *FGeometryCore*

Klase *FPoint*, *FLine*, *FTriangle* i *FCircle* nastale su nasleđivanjem apstraktne klase *FGeometry*. U klasama koje nasleđuju *FGeometry* potrebno je implementirati metode *inAlfaCut* i *inFuzzySet*. Svi geometrijski objekti paketa *FGeometryCore* predstavljaju fazi skup definisan nad skupom R^2 . Funkcija pripadnosti koja ih karakteriše imeplementirana je

preko funkcije *inFuzzySet* koji svaka od klasa treba da implementira. Druga funkcija koju je neophodno implementirati je funkcija *inAlfaCut*. Ova funkcija se koristi za proveru da li se precizna tačka nalazi unutar α –preseka. Klasa *Fpoint* prikazana je na slici 3.9.

FPoint		
+	center	: Point
+	radius	: double
+	inAlfaCut (double alfa, Point X)	: Boolean
+	<<Override>> InFuzzySet (Point X)	: double
+	<<Operator>> + (FPoint c1, FPoint c2)	: FPoint
+	<<Operator>> - (FPoint c1, FPoint c2)	: FPoint
+	<<Operator>> * (double u, FPoint c)	: FPoint
+	distance (FPoint Y)	: FNumber

Slika 3.9: Klasa FPoint

Funkcija koja proverava da li je tačka unutar alfa preseka implementirana je na način dat u listingu 3.1.:

```
public Boolean inAlfaCut(double alfa, Point X)
{
    double d = center.distance(X);
    return (d <= (1-alfa)*radius);
}
```

Listing 3.1: Funkcija *inAlfaCut*

Funkcija *inFuzzySet* prikazana je listingom 3.2.

```
public double InFuzzySet(Point X){
    double d = center.distance(X);
    if (radius > 0)
        return Math.Max(0, 1 - d / radius);
    else {
        if (d == 0)
            return 1;
        else
            return 0;
    }
}
```

Listing 3.2: Funkcija *inFuzzySet*

Implementacija zbira dve fazi tačke (definicija 2.9) prikazana je listingom 3.3.

```
public static FPoint operator +(FPoint c1, FPoint c2)
{
    FPoint T = new FPoint();
    T.x = c1.center.x + c2.center.x;
    T.y = c1.center.y + c2.center.y;
    T.radius = c1.radius + c2.radius;
    return T;
}
```

Listing 3.3: Operator zbira dve fazi tačke

Proizvod skalara i fazi tačke (definicija 2.11) prikazana je listingom 3.4.

```
public static FPoint operator *(double u, FPoint c)
{
    FPoint T = new FPoint();
    T.center.x = u*c.center.x;
    T.center.y = u*c.center.y;
    T.radius = u*c.radius;
    return T;
}
```

Listing 3.4: Proizvod skalara i fazi tačke

Ovako definisani operatori i fazi tačka omogućuju veoma jednostavno i intuitivno implementiranje linearne kombinacije dve fazi tačke (definicija 2.12.) na način dat u listingu 3.5:

```
FPoint A = new FPoint(10, 20, 3);
FPoint B = new FPoint(2, 4, 1);
FPoint C = A + 0.3 * (B - A);
```

Listing 3.5: Linearna kombinacija dve fazi tačke

Pored ovih funkcija implementirana je i funkcija za računanje rastojanja između dve fazi tačke.

U klasi *FuzzyOrdering* (slika 3.10) implementirane su funkcije za računanje fazi relacije poretka (definicija 1.2) definisane između *FNumber* objekata.

FuzzyOrdering	
+ leRF (FNumber A, FNumber B)	: double
+ leLF (FNumber A, FNumber B)	: double
+ maximal (FNumber A, List<FNumber> set)	: double
+ minimal (FNumber A, List<FNumber> set)	: double

Slika 3.10: Klasa *FuzzyOrdering*

Implementacija fazi relacije maksimalni element u skupu fazi tačaka u programskom jeziku C# prikazana je u listingu 3.6:

```
public double maximal(FNumber A, List<FNumber> set) {
    double m = 1;
    foreach (FNumber Y in set) {
        m = Math.Min(m, FuzzyOrdering.leRF(Y, A));
    }
    return m;
}
```

Listing 3.6: Funkcija pripadnosti fazi skupu maksimal

U klasi *FLine* (Slika 3.11) implementiran je geometrijski oblik fazi duž.

FLine	
+ a : FPoint	
+ b : FPoint	
+ inAlfaCut (double alfa, Point p)	: Boolean
+ <<Override>> InFuzzySet (Point P)	: double
+ fuzzyImage (Point X)	: FPoint
+ fuzzyImageValue (Point X)	: double

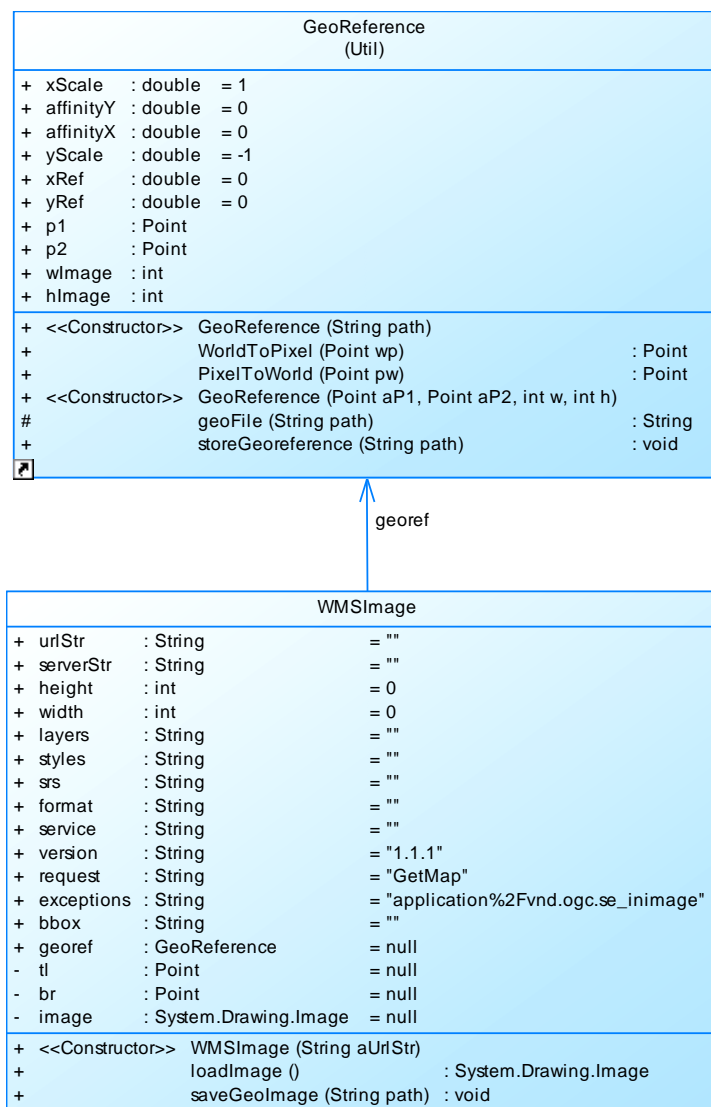
Slika 3.11: Klasa *FLine*

U klasi *FuzzyGeometryFunctions* (Slika 3.12) implementirane su prostorne funkcije opisane u prethodnom poglavlju.

FuzzyGeometryFunctions	
+ internalHomoteticCenter (FPoint f1, FPoint f2)	: Point
+ fMove (FPoint A, Point T)	: FPoint
+ fRotate (FPoint A, double angle)	: FPoint
+ fBlur (FPoint A, double t)	: FPoint
+ fSharp (FPoint A, double t)	: FPoint
+ coin (FPoint A, FPoint B)	: double
+ contain (FPoint A, FLine BC)	: double
+ coli (FPoint A, FPoint B, FPoint C)	: double
+ pos (FPoint A, FPoint B, FPoint C)	: double
+ neg (FPoint A, FPoint B, FPoint C)	: double

Slika 3.12: Klasa *FuzzyGeometryFunctions*

Sastavni deo čine i klase *WMSImage* i *Georeference* iz paketa *fGeometryCore.util* koje se koriste kao pomoćne klase u komunikaciji sa *geoserverom*. Dijagram je prikazan na slici 3.13.



Slika 3.13: Dijagram klasa paketa *fGeometryCore.util*

Konstruktor klase *WMSImage* prima kao parametar url adresu mape, ta adresa odgovara WMS GetMap formatu. Primer jednog wms getMap url-a dat je listingom 3.7.

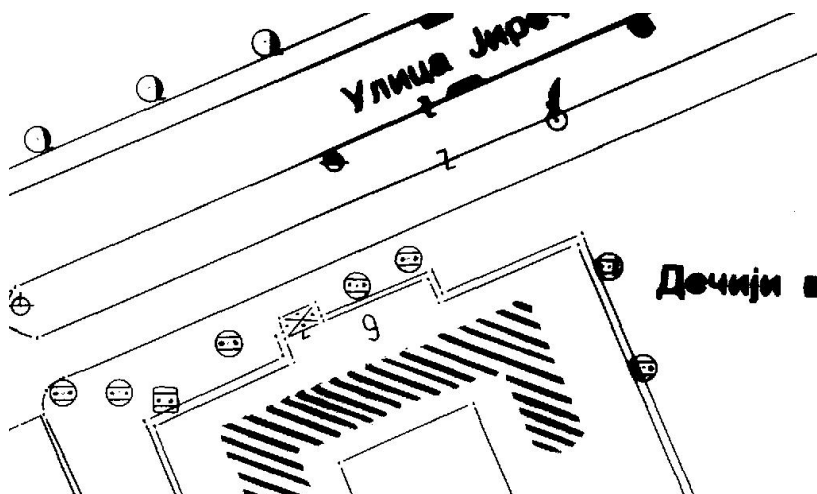
```

http://localhost:8080/geoserver/wms?
HEIGHT=610&WIDTH=1024&LAYERS=cite:NS1-052&
STYLES=&SRS=EPSG:4326&FORMAT=image/jpeg&
SERVICE=WMS&VERSION=1.1.1&REQUEST=GetMap&
EXCEPTIONS=application/vnd.ogc.se_inimage&
BBOX=7410128.39760462,5011526.50800929,7410193.21487959,501156
5.1886694

```

Listing 3.7: Primer WMS url-a

Metoda *loadImage* koristi se za preuzimanje mape u obliku slike. Primer slike koja se dobija na osnovu url-a datog u listingu 3.7 prikazan je na slici 3.14.



Slika 3.14: Mapa koja se dobije pozivom url-a datog listingom 3.7

Metodom *saveGeoImage* sliku je moguće sačuvati u georeferenciranom obliku. Datoteka za georeferenciranje slike 3.8 prikazana je listingom 3.8.

```

0.063298120087893 (veličina piksela u smeru x ose)
0 (stepen rotacije za vrste)
0 (stepen rotacije za kolone)
-0.0634109182130606 (veličina piksela u smeru z ose)
7410128.39760462 (x kordinata centra gornjeg levog piksela)
5011565.1886694 (y kordinata centra gornjeg levog piksela)

```

Listing 3.8: Sadržaj *.jgw* datoteke za georeferenciranje

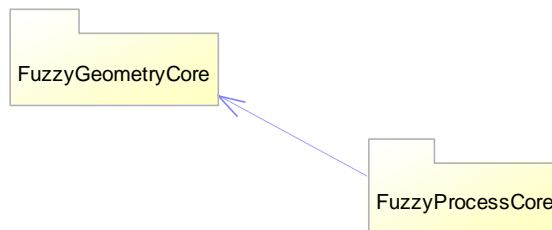
Klasa *FuzzyGeometryFunctions* sadrži većinu implementiranih funkcija sistema. Primer funkcije *contain* dat je listingom 3.9.

```
public static double contain(FPoint A, FLine BC)
{
    Point B = BC.b;
    Point C = BC.a;
    double u = A.projekcija(B, C);
    if (0 <= u && u <= 1)
    {
        Point ba = internalHomoteticCenter(B, A);
        Point ac = internalHomoteticCenter(A, C);
        Point aP = A.center.projekcijaNaDuz(ba, ac);
        double rA = Math.Abs(A.radius);
        return Math.Max(0, 1 - aP.distance(A.center) / rA);
    }
    else if (u < 0)
    {
        return coin(a, b);
    }
    else
    {
        // if (u > 1) {
        return coin(a, c);
        }
    }
    return 0;
}
```

Listing 3.9: Funkcija *contain* klase *FuzzyGeometryFunctions*

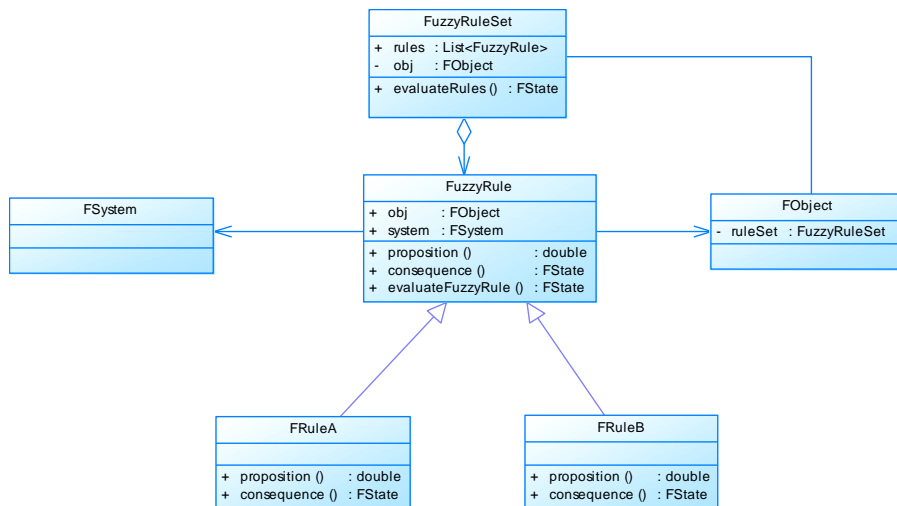
Biblioteke klasa za modeliranje procesa

Biblioteka za modeliranje fazi procesa implementirana je kao dinamička biblioteka. Na slici 3.15 je prikazan dijagram povezanosti između paketa *FuzzyGeometryCore* i *FuzzyProcessCore* u kojem je implementirana podrška za upravljanje i simulaciju geoprostornih procesa.



Slika 3.15: Dijagram povezanosti paketa *FuzzyGeometryCore* i *FuzzyProcessCore*

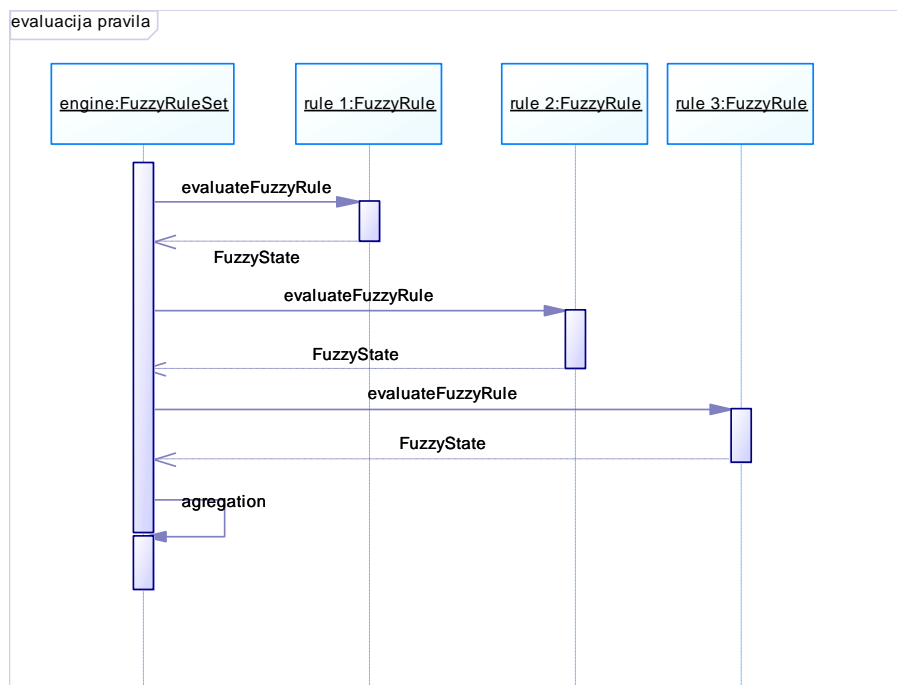
Osnovu paketa *FuzzyProcessCore* čine klase *FObject*, *FState* i *FSet*. Pored ovih klasa implementirane su klase za modeliranje i evaluaciju fazi pravila (*FuzzyRule*) i klase *FuzzyRuleSet* koja predstavlja skup fazi pravila čiji zadatak je da izvrši agregaciju evaluiranih fazi pravila. Dijagram klasa paketa *FuzzyProcessCore* prikazan je na slici 3.16.



Slika 3.16: Dijagram klasa za modeliranje fazi pravila

Klasa *FuzzyRuleSet* agregira više instanci klase *FuzzyRule*.

Evaluacija pravila koja pripadaju klasi *FuzzyRuleSet* vrši se pozivom metode *evaluateRules* u kojoj se vrši agregacija evaluiranih fazi pravila.



Slika 3.17: Dijagram redosleda akcija prilikom evaluacije fazi pravila
Implementacija klase *FuzzyRuleSet* data je listingom 3.10.

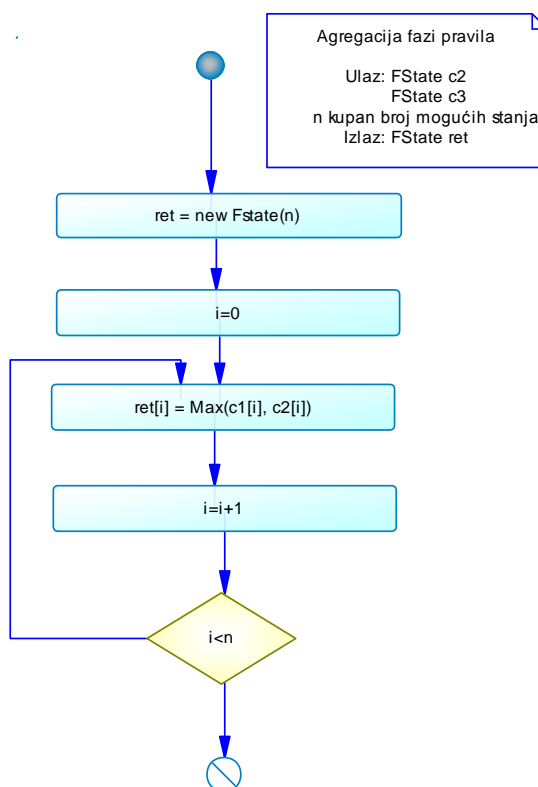
```

public class FuzzyRuleSet
{
    public List<FuzzyRule> rules;
    public FObject obj;

    public FState evaluateRules() {
        FState ret = new FState();
        foreach (FuzzyRule fr in rules) {
            fr.obj = obj;
            FState ff = fr.evaluateFuzzyRule();
            ret = ret + ff;
        }
        return ret;
    }
}
  
```

Listing 3.10: Klasa *FuzzyRuleSet*

Na slici 3.18 prikazan je dijagram aktivnosti kojim je implementiran operator agregacije + između dva fazi stanja.



Slika 3.18: Dijagram aktivnosti za agregaciju dva fazi pravila

Podsistem za skladištenje podataka

Kao osnova za skladištenje geoprostornih podataka izabran je PostGIS sistem za upravljanje podacima. PostGIS sistem istovremeno omogućuje i podršku radu sa prostornim i alfanumeričkim podacima a poseduje interfejse za jednostavno povezivanje preko ODBC i JDBC standarda. Takođe jednostavna je integracija podataka iz PostGIS baze sa Geoserverom koji predstavlja standardni integrator prostornih podataka.

Prvi korak u proširenju PostGIS sistema je implementacija osnovnog *nepreciznog* pozicionog tipa u skladu sa definicijom 2.1.

```
CREATE TYPE fuzzyPoint AS (
    core geometry,
    supportRadius float
);
```

Listing 3.11: SQL izraz za kreiranje *fuzzyPoint* tipa

Osnovne funkcije za rad sa *fuzzyPoint* tipom:

```
CREATE OR REPLACE FUNCTION
fPointCore(a fuzzyPoint) RETURNS geometry AS
$BODY$
    BEGIN
        return a.core;
    END;
$BODY$ LANGUAGE 'plpgsql';

CREATE OR REPLACE FUNCTION
fPointRadius(a fuzzyPoint) RETURNS geometry AS
$BODY$
    BEGIN
        return a.core;
    END;
$BODY$ LANGUAGE 'plpgsql';

CREATE OR REPLACE FUNCTION
fPoint(core geometry, support float)
RETURNS fuzzyPoint AS
$BODY$
DECLARE
    retVal fuzzyPoint;
BEGIN
    retVal.core = core;
    retVal.supportRadius = support;
    return retVal;
END;
$BODY$ LANGUAGE 'plpgsql';

CREATE OR REPLACE FUNCTION
aseWKT(a fuzzyPoint) RETURNS text AS
$BODY$
BEGIN
    return 'FPOINT('||x(a.core)||
        ' '||y(a.core)||
        ' '||a.supportRadius||')';
END;
$BODY$ LANGUAGE 'plpgsql';
```

Listing 3.12: Osnovne funkcije za rad sa fuzzy point objektom

Zajedno sa definicijom novog prostornog tipa potrebno je kreirati tabelu koja služi kao registar svih tabela u kojima se koristi *fuzzyPoint* tip.

```

CREATE TABLE fuzzy_point_columns
(
  f_table_catalog VARCHAR(256) NOT NULL,
  f_table_schema VARCHAR(256) NOT NULL,
  f_table_name VARCHAR(256) NOT NULL,
  f_geometry_column VARCHAR(256) NOT NULL,
  support_radius_column VARCHAR(256) NOT NULL,
  srid integer NOT NULL,
  CONSTRAINT fuzzy_point_columns_pk PRIMARY KEY(
    f_table_catalog,
    f_table_schema,
    f_table_name,
    f_geometry_column)
)

```

Listing 3.13: SQL izraz za kreiranje tabele *fuzzy_point_columns*

Primer korišćenje *fuzzyPoint* tipa dat je sledećom naredbom:

```

CREATE TABLE "Tacka_op_fuzzy"
(
  gid serial NOT NULL,
  komentar character varying(32),
  max_pdop double precision,
  gps_date date,
  gps_time character varying(10),
  unfilt_pos numeric(10,0),
  gps_height double precision,
  f_position fuzzyPoint
)

```

Listing 3.14: SQL izraz za kreiranje tabele *Tacka_op_fuzzy*

Kao što se vidi u primeru tabela pored klasičnih relacionih tipova podataka koristi i novi tip *fuzzyPoint*. Atributom *f_position* predstavljen je model prostornog podataka koji u sebi sadrži i informaciju o nepreciznosti.

Po definiciji rastojanja u uslovima neodređenosti, rastojanje je nenegativan fazi broj. U nastavku je data deklaracija tipa za reprezentaciju nenegativnog fazi broja.

```

CREATE TYPE nonNegativeLinearFuzzyNumber As (
  core float,
  supportRadius float
);

```

Listing 3.15: SQL izraz za kreiranje nenegativnog fazi broja

Funkcija za računanje rastojanja između objekata čija pozicija se predstavlja *fuzzyPoint* prostornim tipom tada se definiše na sledeći način:

```

CREATE OR REPLACE FUNCTION
  f_distance(a fuzzyPoint, b fuzzyPoint)
  RETURNS nonNegativeLinearFuzzyNumber AS
$BODY$
DECLARE
  retVal nonNegativeLinearFuzzyNumber;
BEGIN
  retVal.core = distance(a.core, b.core);
  retVal.supportRadius =
    max(a.supportRadius, b.supportRadius);
  return retVal;
END;
$BODY$
LANGUAGE 'plpgsql';

```

Listing 3.16: SQL funkcija za računanje fazi rastojanja

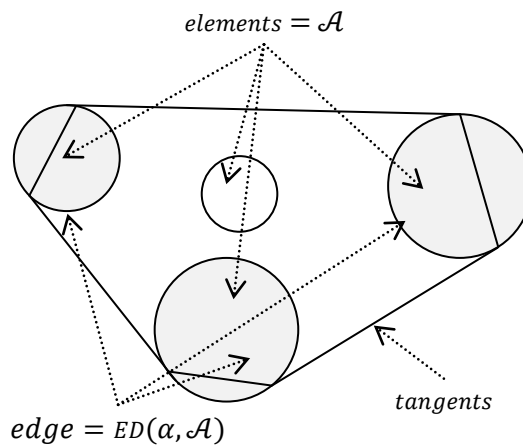
α – konveksni omotač skupa \mathcal{A} (definicija 2.7) u PostGISu deklariramo kao novi tip podataka *alfaConvexHullPolygon* (slika 3.19). Atribut *elements* sadrži ceo skup \mathcal{A} , *alfa* predstavlja vrednost parametra α , *tangents* je tipa *polygon* i sadrži sve tangente između α preseka i na kraju *edge* sadrži sve tačke iz skupa \mathcal{A} koje dodiruju neku od tangenti iz skupa *tangents*.

```

CREATE TYPE alfaConvexHullPolygon AS (
  elements _fuzzyPoint,
  alfa float,
  tangents polygon,
  edge _fuzzyPoint
);

```

Listing 3.17: SQL izraz za kreiranje alfa konveksnog poligona



Slika 3.19: Ilustracija tipa *alfaConvexHullPolygon*

Jedna od veoma važnih funkcija je $f_insideCH$ koja služi za proveru da li se tačka nalazi unutar alfa konveksnog omotača. Funkcija je implementirana u Pl/PqSQL jeziku:

```
CREATE OR REPLACE FUNCTION
  f_insideCH(p point, ch alfaConvexHullPolygon)
  RETURNS boolean AS
$BODY$
DECLARE
  retVal boolean;
  d float;
  alfa float;
  fp fuzzyPoint;
  n int;
BEGIN
  alfa = ch.alfa;
  retVal = st_contains(p, ch.tangents);
  if (retVal == false) then
    n := array_upper(ch.edge, 1);
    FOR i IN 1..n LOOP
      fp = ch.edge[i];
      d = distance(p, fp.core);
      if d < fp.supportRadius*(1-alfa) then
        return true;
      end if;
    end loop;
  end if;
  return retVal;
END;
$BODY$
LANGUAGE 'plpgsql';
```

Listing 3.18: Funkcija za proveru pripadnosti tačke alfa konveksnom omotaču

Funkcija $f_fuzzyConvexHull$ određuje alfa konveksni omotač skupa fazi tačaka. Ulazni parametri su realan broj $alfa$ (α) i niz fazi tačaka $elements$ (skup \mathcal{A}), povratna vrednost je struktura $alfaConvexHullPolygon$. Deklaracija funkcije data je u nastavku:

```
CREATE OR REPLACE FUNCTION
  f_fuzzyConvexHull(alfa float, elements _fuzzyPoint)
  RETURNS alfaConvexHullPolygon AS
$BODY$
DECLARE
  retVal alfaConvexHullPolygon;
  tanPoints _point;
  nElements int;
  nEdge int;
  fp fuzzyPoint;
  polygonExtended boolean;
BEGIN
  retVal.elements = elements;
  retVal.alfa = alfa;
  nElements := array_upper(elements, 1);
  retVal.edge[1] = elements[1];
  nEdge = 1;
```

```
FOR i IN 2..nElements LOOP
  fp = elements[i];
  polygonExtended = false;
  FOR j IN 1..nEdge LOOP
    tanPoints = simpleConvexHull(fp,
                                retVal.edge[j], alfa);
    if(!f_insideCH(tanPoints[1], retVal))
      then
        f_extendConvexPolygon(retVal,
                              tanPoints[1],
                              tanPoints[2]);
        polygonExtended = true;
      end if;
    if(!f_insideCH(tanPoints[3], retVal))
      then
        f_extendConvexPolygon(retVal,
                              tanPoints[3],
                              tanPoints[4]);
        polygonExtended = true;
      end if;
    END LOOP;
  IF(polygonExtended==true) THEN
    retVal.edge[nEdge+1] = fp;
    nEdge := nEdge + 1;
  END IF;
END LOOP;
return retVal;
END;
$BODY$
LANGUAGE 'plpgsql';
```

Listing 3.19: Implementacija funkcije za određivanje alfa konveksnog omotača

Ovako definisane tipove i funkcije pogodnije je koristiti u nekom od objektnih okruženja.

FGeoIS i FGeoISProcess podsistemi

FGeoIS podsistem predstavlja okruženje za rad sa procesima kao prostornim resursima. Implementacija je izvršena u Java programskom okruženju. Serverski deo je implementiran kao web aplikacija sa implementiranom podrškom za rad web servisa u skladu sa J2EE standardom. ImageJ paket je iskorišćen za rad sa različitim formatima slika. Procesi su implementirani kao *jar* paketi u kojima jedna od klasa opisuje ponašanje glavnog procesa a web servisi implementiraju se kao java klase. Web servisi predstavljaju interfejs preko kojeg se upravlja izvršavanjem serverskih procesa.

FGeoIS programski paket implementiran je kao web aplikacija čiji osnovni zadatak je da obezbedi podršku uređenju kataloga prostornih procesa, servisa, njihovo izvršavanje i povezivanje sa geoprostornim resursima poput *geoservera* i *FPostGIS* servera. Na slici 3.20 prikazan je dijagram korišćenja podsistema *FGeoIS*. Podsistem koriste dve grupe korisnika: *administratori* i *programeri*. Prva grupa (*administratori*) obavljaju poslove uređenja svih kataloga. Druga grupa (*programeri*) mogu da obavljaju ograničen skupo poslova poput kreiranja i dodavanja procesa, aktivacije i pokretanja procesa. Svi katalogi na *FGeoIS* serveru sastoje se od drugih podkataloga i elemenata. Elementi kataloga mogu da budu:

- (i) web servisi,
- (ii) procesi,
- (iii) proxy klase,
- (iv) dokumenti.

Web servisi nalaze se u katalogu */servisi/* pripremaju se u *FGeoISProcess* podsistemu u obliku *java* datoteka a prebacuju na server preko web interfejsa.

Procesi se nalaze u katalogu */tasks/* ili */processes/*, pripremaju se u *FGeoISProcess* podsistemu od strane programera u obliku *jar* arhive koja može da sadrži vrlo kompleksne modele procesa. Na server se prebacuju preko web interfejsa.

Klase za komunikaciju predstavljaju vezu između korisničkih aplikacija i samih procesa, kreiraju se posebno razvijenim programskim paketom na osnovu *wSDL* definicije web servisa.



Slika 3.20: Dijagrami korišćenja podsistema *FGEOIS*

Slučaj korišćenja ***Uređivanje kataloga web servisa***

Opis: Ovaj slučaj korišćenja obuhvata osnovne elementarne operacije poput, dodavanja, arhiviranja i preuzimanja (kopiranje i prebacivanje) web servisa.

Korisnici: Administrator i programer.

Okidač: Pokretanje preko naredbi korisničkog interfejsa u obliku dugmadi za pokretanje i kontekstnog menija.

Završni uslovi: Uređen katalog web servisa.

Osnovni tok: Korisnici u ovom slučaju mogu da dodaju novi web servis u obliku *java* klase. Postojeći web servis mogu da svrstaju u drugu kategoriju, arhiviraju servis ili ga preuzmu radi dalje distribucije na drugi server.

Slučaj korišćenja Uređivanje kataloga procesa

- Opis:* Ovaj slučaj korišćenja obuhvata osnovne elementarne operacije poput, dodavanja novog procesa, arhiviranja i preuzimanja (kopiranje i prebacivanje) procesa.
- Korisnici:* Administrator i programer.
- Okidač:* Pokretanje preko naredbi korisničkog interfejsa u obliku dugmadi za pokretanje i kontekstnog menija.
- Završni uslovi:* Uređen katalog geoprostornih procesa.
- Osnovni tok:* Korisnici u ovom slučaju mogu da dodaju novi proces u obliku *.jar* arhive. Postojeći proces mogu da prebace u drugu kategoriju, arhiviraju proces ili ga preuzmu radi dalje distribucije na drugi server.

Slučaj korišćenja Uređivanje kataloga komponenti

- Opis:* Komponente predstavljaju elemente za proširenje klijentskih aplikacija ili biblioteke klasa za povezivanje *FGeoIS* servera i ostalih delova sistema. Ovaj slučaj korišćenja obuhvata osnovne elementarne operacije poput, dodavanja, arhiviranja, preuzimanja, uređivanja (kopiranje i prebacivanje) komponenti raspoređenih u kategorije:
- (i) komponente za povezivanje web servisa
 - (ii) komponente za povezivanje sa FPostGIS sistemom (rezultat izvršavanja GmlToJava i GmlToCSharp podsistema)
 - (iii) gotovi dodaci za AutoCAD
 - (iv) osnovni elementi za proširenje AutoCAD programskog paketa
- Korisnici:* Administrator i programer.
- Okidač:* Pokretanje preko naredbi korisničkog interfejsa u obliku dugmadi za pokretanje i kontekstnog menija.
- Završni uslovi:* Uređen katalog komponenti.
- Osnovni tok:* Komponente neophodne za implementaciju složenih klijentskih aplikacija razvijenih alatima poput generatora klasa za povezivanje sa web servisima,

generatora POJO klasa na osnovu WFS specifikacije objekta ili generisanih klasa za pristup sistemu za skladištenje geoprostornih podataka.

Slučaj korišćenja ***Uređivanje kataloga dokumenata***

Opis: Dokumenti u ovom kontekstu predstavljaju tehničke specifikacije implementiranih geoprostornih procesa, modele razvijenih komponenti, georeferencirane mape, *shape* datoteke i drugo.

Korisnici: Administrator

Okidač: Pokretanje preko naredbi korisničkog interfejsa u obliku dugmadi za pokretanje i kontekstnog menija.

Završni uslovi: Uređen katalog dokumenata.

Osnovni tok: Tehničke specifikacije razvijenih komponenti ili klijentskih aplikacija prebacuju se na server u odgovarajući katalog. Katalog je moguće vremenom dopunjavati, pojedinačne elemente arhivirati ili preuzimati i prebacivati na drugi FGeoIS server.

Slučaj korišćenja ***Uređivanje korisničkih podataka***

Opis: Ovaj slučaj korišćenja obuhvata elementarne operacije dodavanja novog korisnika, promena osnovnih podataka postojećeg korisnika i brisanje korisnika.

Korisnici: Administrator

Okidač: Pokretanje preko naredbi korisničkog interfejsa u obliku dugmadi za pokretanje i kontekstnog menija.

Završni uslovi: Korisnički podaci ažurni.

Osnovni tok: Korisnici sistema moraju pre korišćenja aplikacije biti registrovani u katalogu korisnika. Ovaj posao izvršava administrator.

Slučaj korišćenja **Generisanje klasa za komunikaciju**

Opis: Ovaj slučaj korišćenja obuhvata poslove automatizovanog generisanja klasa u C# ili Java programskim jezicima a čijim korišćenjem se omogućuje jednostavna komunikacija između web servisa i korisničkih aplikacija.

Korisnici: Administrator i programer

Okidač: Pokretanje preko naredbi korisničkog interfejsa u obliku dugmadi za pokretanje i kontekstnog menija.

Završni uslovi: Generisane klase za komunikaciju.

Osnovni tok: Programer izvrši automatizovano generisanje klase za komunikaciju na osnovu WSDL opisa web servisa. Ovako generisanu klasu može da iskoristi za kreiranje komunikacionog sloja klijentske aplikacije ili da istu pošalje na server u katalog komponenti za deljenu upotrebu (ostali programeri mogu preko slučaja korišćenja *Uređivanje kataloga komponenti* preuzimati klasu bez generisanja)

Slučaj korišćenja **Izvršavanje procesa**

Opis: Ovaj slučaj korišćenja obuhvata posao pokretanja i praćenja statusa procesa. Programer preko klasa za komunikaciju pokreće proces koji se izvršava na serveru. Izvršavanja procesa i praćenje rezultata vrši se preko klijentske aplikacije. Korisnik u svakomtrenutku može da zaustavi izvršavanje procesa.

Korisnici: Programer

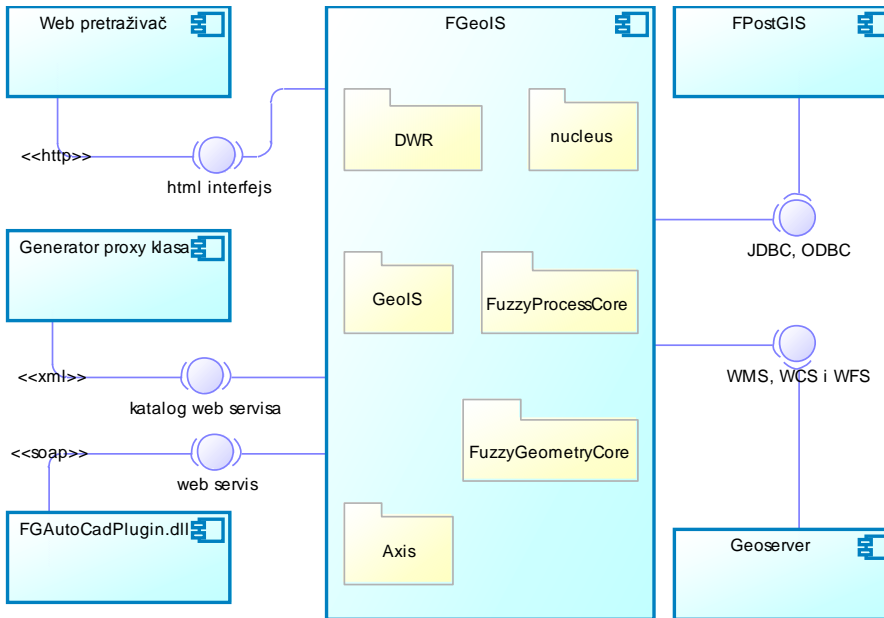
Okidač: Pokretanje preko naredbi korisničkog interfejsa u obliku dugmadi za pokretanje i kontekstnog menija.

Završni uslovi: Proces izvršen.

Osnovni tok: Programer pokrene izvršavanje procesa na serveru preko klase za komunikaciju. Ova klasa prethodno je preuzeta iz kataloga komponenti i uvrštena u klijentsku aplikaciju. U toku izvršavanja procesa

programer može da privremeno zaustavi izvršavanje procesa.

Na slici 3.21 prikazan je dijagram komponenti odnosno globalna arhitektura sistema. Centralni deo sistema je *FGeoIS* WEB aplikacija zadužena za koordinaciju i integraciju svih delova sistema.



Slika 3.21: Dijagram komponenti podsistema FGeoIS

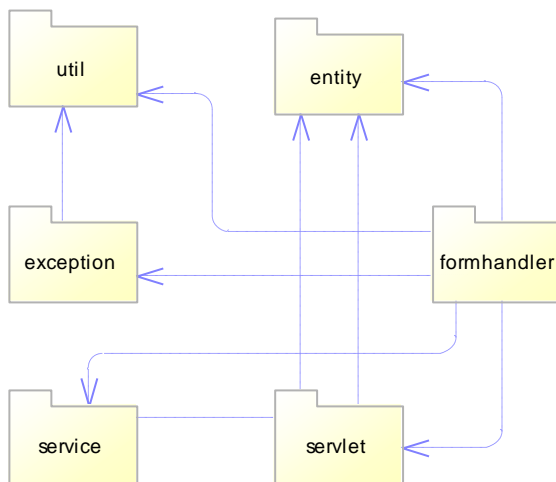
FGeoIS web aplikacija implementirana je uz korišćenje dva gotova paketa klasa *DWR* i *Apache Axis*.

Paket *DWR* (*Direct Web Remoting*) koristi se za obezbeđenje podrške asinhronom javascript XML (*ajax*) komunikaciji servera i klijenta. *DWR* je paket otvorenog koda. Osnovna prednost korišćenja *ajax* tehnologije je mogućnost razvoja kompleksnih web aplikacija u kojoj postoji dvosmerna komunikacija između klijenta i servera.

Paket *Apache Axis* je implementacija SOAP-u (*Simple Object Access Protocol*). *Apache Axis* paket, pripada grupi paketa otvorenog koda.

U nastavku je dat statički model *GeoIS* i *nucleus* paketa kojima su imeplemirane osnovne funkcije *FGeoIS* web servera.

Nucleus predstavlja jezgro web aplikacije a koristi se u više aplikacija [65-67], i sadrži pakete za pristup sistemima za rukovanje podacima (JDBC standard), biblioteku *tagova* kojim se obezbeđuje prikaz svih elemenata, pakete za rad sa XML datotekama, pakete za *backup* i *restore* baze u XML datoteku, pakete koji su zaduženi za rad sa *logovima*, pakete koji su zaduženi za rad sa konfiguracionim parametrima.



Slika 3.22: Dijagram povezanosti paketa *nucleus* podsistema

Osnovu *nucleus* podsistema (Slika 3.22) čine klase koje se koriste za generisanje korisničkog interfejsa, dela za obradu *http* zahteva i dela za rad sa komponentama. Sve komponente koje se u ovom okruženju koriste umesto da se definišu u *jsp:useBean* tagu na početku svake stranice kako je definisano JSP standardom navode se u *components.xml* datoteci a *nucleus* dalje upravlja pristupu komponentama. Primer *comonentes.xml* datoteke dat je u listingu 3.20.

```

<components>
  <component id="user" scope="session"
    class="com.gint.se.formhandler.UserHandler"/>
  ....
</components>

```

Listing 3.20: Primer *components.xml* datoteka

Ovim pristupom izbegnuto je da ponavljanje definisanja komponenti na svakoj strani. Svaka *jsp* strana web aplikacije na početku treba da ima samo sledeće dve linije:

```
<%@ taglib uri="tags" prefix="util" %>
<util:nucleus/>
```

Nucleus tag je implementiran tako da se sva obrada http zahteva obavi centralizovano i time olakša razvoj kompleksnih web aplikacija. Primer korišćenja i pristupa komponentama na *.jsp* stranicma prikazan je u sledećem listingu

```
<form name="loginForm" method="post">
  Korisnicko ime:
  <util:input type="text" bean="user.userName"/><br>
  Lozinka:
  <util:input type="password" bean="user.password"/><br>
  <input type="submit"
    name="bean:user.login" value="index.jsp">
</form>
```

Listing 3.21: Primer pristupa atributima komponenti

U ovom primeru prikazana je forma sa dva tekst polja i jednim submit dugmetom. Prvo tekst polje će biti upisano u atribut *userName* komponente *user* tako što će se posredstvom *nucleus* taga pozvati metoda *setUserName* komponente *user*. Parametar "*bean:user.login*" će kao rezultat imati pozivanje metode *executeLogin* komponente *user*. *Nucleus* tag će za svaki parametar proveriti da li postoji metoda koja odgovara imenu parametra koja počinje sa *beforeSet*, *set* i *execute*, i ako postoji poziva se sa vrednošću parametra prethodno konvertovanog u odgovarajući tip. Prvo će se izvršiti sve *beforeSet* metode, nakon toga *set* i na kraju *execute* metode. Zato se *beforeSet* metode koriste za inicijalizaciju, *set* za postavljanje vrednosti, a *execute* za obradu, jer redosled slanja parametara forme nije standardom definisan i različiti *web klijenti* parametre šalju u drugačijem redosledu.

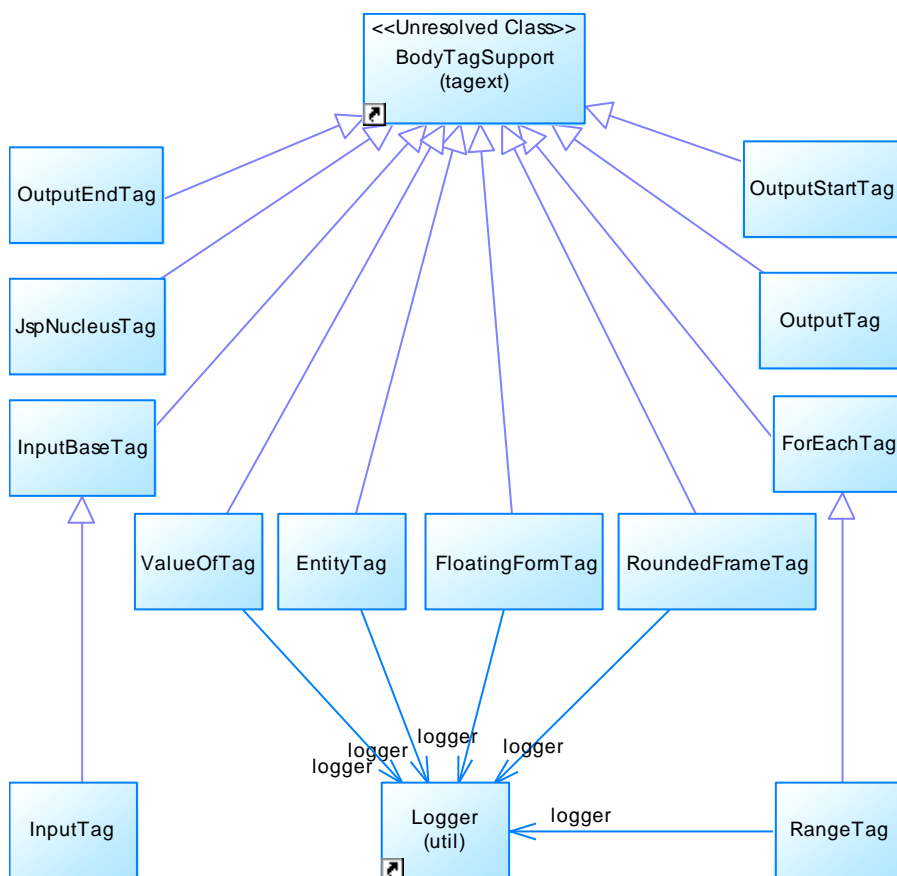
Za prikaz vrednosti određenih komponenti koriste se sledeći tagovi: *valueof*, *forEach*, *range*, *switch* itd. Na primer, ako želimo da prikažemo vrednost atributa *userName* komponente *user* na *jsp* stranici pišemo sledeće: `<util:valueof bean="user.userName"/>` ili ako želimo da prikažemo imana osoba koja se nalaze u vektoru *all* komponente *user* to ćemo uraditi na sledeći način:

```
<util:forEach rows="bean:user.all" sortBy="prezime">
  <util:output>
    <util:valueof bean="element.prezime"/><br>
  </util:output>
</util:forEach>
```

stranično prikazivanje obezbeđeno je *range* tagom:

```
<util:range rows="bean:user.all" start="1" howMany="5">
<util:output>
<util:valueof bean="element.prezime"/><br>
</util:output>
</util:range>
```

Dijagram osnovnih klasa paketa *util.tag* prikazan je na slici 3.23.



Slika 3.23: Dijagram osnovnih klasa paketa *util.tag*

Dve najznačajnije klase paketa *util* su klase *JspNucleus* i *ContextParser*. Ove dve klase zadužene su za obradu zahteva i za registraciju komponenti na osnovu *components.xml* datoteke. Pored ovih klasa paket *util* sadrži i pakete *date*, *db*, *html*, *mail* i *xml*. U paketu *db* nalazi se *ConnectionPool* klasa, koja predstavlja skup konenkcija sa bazom. Paket *xml* sadrži klase koje se koriste za rad sa xml datotekama (parsiranje i modifikaciju).

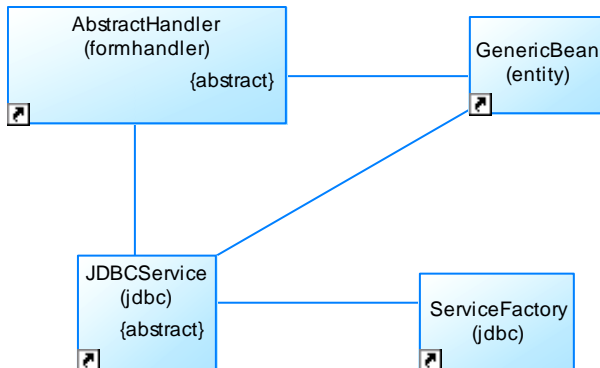
Na slici 3.24 prikazan je model klase *ContextParser* koja je zadužena za analizu parametara http zahteva koji se razmenjuje između klijentske i serverske strane.

ContextParser		
-	logger	: Logger = new Logger(ContextParser.class)
#	multiRequest	: Hashtable = null
+	getDeclaredMethods (Class cl, Hashtable ht)	: Hashtable
+	getField (PageContext pageContext, Object object, String field)	: Object
+	parse (String methodName, Object temp, int index)	: Object
+	parseBean (PageContext pageContext, String paramStr)	: Object
+	setMultiRequest (Hashtable aMultiRequest)	: void
+	parseParam (PageContext pageContext, String paramStr)	: Object
+	parseBeanValue (PageContext pageContext, String paramStr)	: Object
+	setBeanValue (PageContext pageContext, String propertyName, String propertyValue, String type)	: void
#	invokeMethod (PageContext pageContext, String methodName, Object obj, Method ms, String propertyValue)	: void
+	push (PageContext pageContext, String componentName, Object obj)	: void
+	pop (PageContext pageContext, String componentName)	: void
+	elementAtLevel (PageContext pageContext, String componentName, int level)	: Object

Slika 3.24: Dijagram klase *ContextParser*

Klasa *JspNucleus* zadužena je za dinamičko kreiranje komponenti registrovanih u *components.xml* konfiguracionoj datoteci.

Osnova za kreiranje komponenti predstavlja realizacija abstraktnih klasa prikazanih na slici 3.25.



Slika 3.25: Dijagram povezanosti klasa za implementaciju komponenti

Svaka pojedinačna klasa predstavlja reprezent paketa u kojem se nalazi. Tako na primer klasa *AbstractHandler* predstavlja osnovu za implementaciju svih klasa u kojima je opisana poslovna logika, *GenericBean* je osnova za kreiranje entiteta a *JDBCService* klasa je osnova za impementaciju klasa koje pomažu prilikom skladištenja i pretraživanja *entity* objekata. FormHandler klase nastale implementacijom klase *AbstractHandler* koriste se kao komponente na jsp stranama.

Korišćenjem *DWR* paketa moguće je uspostaviti direktnu vezu između elemenata korisničkog interfejsa i komponenti registrovanih u

components.xml. Konfiguracija DWR paketa vrši se preko *dwr.xml* datoteke smeštene u `<webapp>/WEB-INF/` direktorijumu.

Primer *dwr.xml* klase prikazan je u listingu 3.22.

```
<dwr>
  <allow>
    <convert converter="bean"
      match="com.gint.geoIs.entity.FileInfo"/>
    <create creator="new"
      javascript="FileSystem"
      scope="session">
      <param name="class"
        value="com.gint.geoIs.formhandler.FileSystemHandler"/>
    </create>
    <create creator="new"
      javascript="FileSystemDialog"
      scope="session">
      <param name="class"
        value="com.gint.geoIs.formhandler.FileSystemHandler"/>
    </create>
  </allow>
</dwr>
```

Listing 3.22: Konfiguraciona datoteka *dwr* paketa

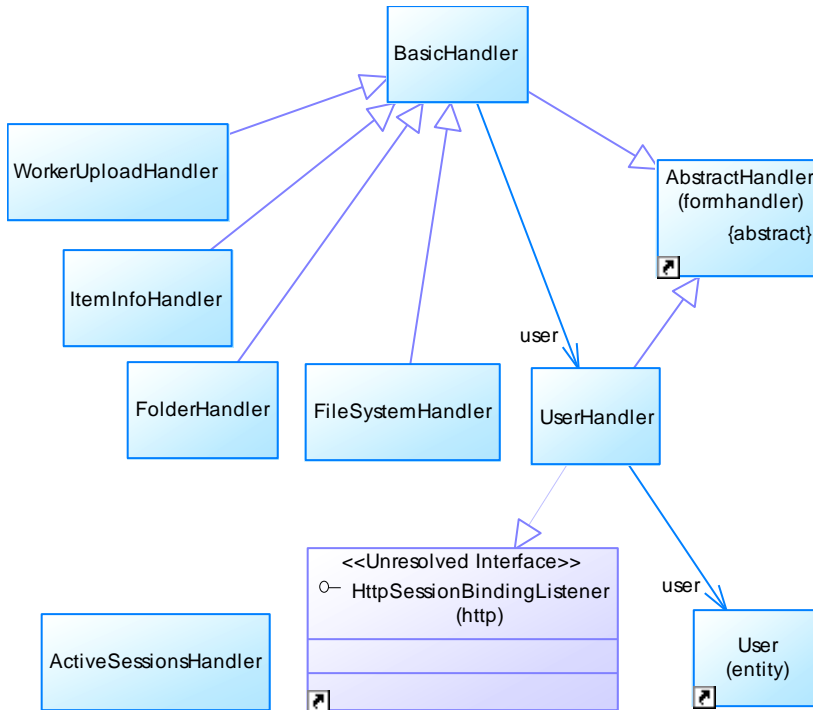
Osnova za podršku radu sa web servisima implementirana je preko *Apache Axis* paketa odnosno odgovarajućeg *ServicesEngineServlet* servleta. Konfiguracija *Apache Axis* paketa navodi se kao deo `<webapp>/WEB-INF/web.xml` registracijom servleta *ServicesEngineServlet* na sledeći način:

```
<servlet-mapping>
  <servlet-name>ServicesEngineServlet</servlet-name>
  <url-pattern>/services/*</url-pattern>
</servlet-mapping>
```

GeoIs paket predstavlja grupu paketa u kojima je izvršena implementacija osnovnih funkcionalnosti samog sistema. Paket se sastoji od tri logičke celine: *entity*, *formhandler* i *service*. Prva celina opisuje jednostavne klase koje predstavljaju osnovne entitete sistema. Druga celina je zadužena za implementaciju klasa koji treba da obezbede osnovnu interakciju sa korisnikom, kao i izvršavanje kompleksnijih procesa. U poslednjoj celini implementirane su klase za pristup sistemu za rukovanje podacima.

Na slici 3.26 prikazan je dijagram klasa paketa *com.gint.geoIs.formhandler*. Klasa *ItemInfoHandler* predstavlja vezu između korisnika, okruženja i procesa. Preko ove klase korisnik može da izvrši upload paketa u kojem se nalazi definicija procesa. Može da registruje paket i da ga izvrši.

Klasa *FolderHandler* izvršava poslove katalogizacije elemenata servera.



Slika 3.26: Paket *com.gint.geols.formhandler*

Na slici 3.26 prikazan je dijagram sekvenci izvršavanja jednog procesa na server-u.

Klasa *FileSystemHandler* u kojoj su implementirane osnovne funkcije sistema prikazana je na slici 3.27. Ova klasa predstavlja komponentu koja se registruje u *components.xml* i *dwr* ajax komponentu u *dwr.xml*.

Implementirane metode koriste se za pregled kataloga tekućeg direktorijuma, dodavanje novih elemenata, uređivanje, brisanje i prikaz postojećih elemenata.

Implementirane su i metode za kopiranje i premeštanje elemenata kao i metode za aktiviranje nove biblioteke zajedno sa pokretanjem procesa.

Klasa je implementirana tako da ju je moguće koristiti u asinhronom režimu a za sinhroni režim implementirane su klase *FolderHandler* i *ItemInfoHandler*.

Metoda *uploadImportFile* koristi se za prebacivanje datoteke sa klijenta na server i tom prilikom java klasu koja predstavlja java web servis konvertuje

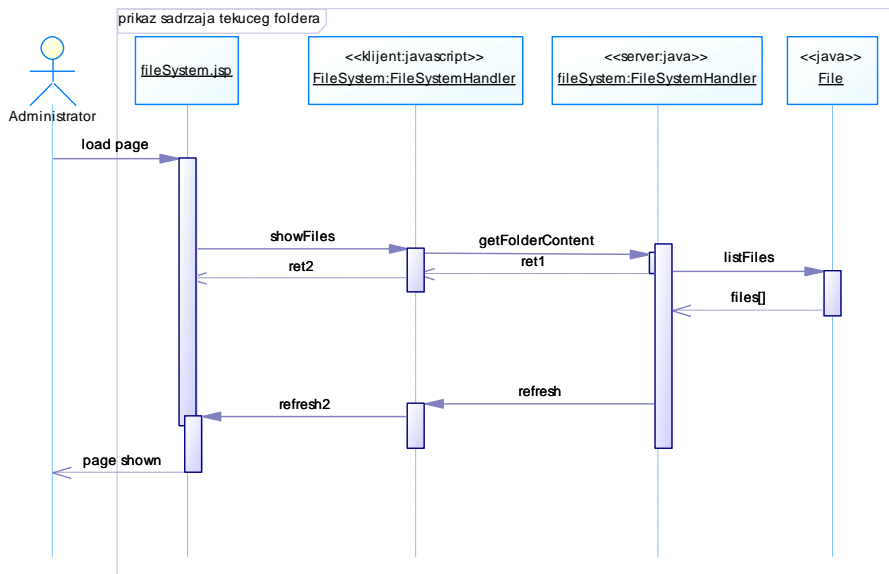
u *.jws* datoteku koju kasnije *Apache Axis* paket ostatku sistema izlaže kao web servis.

FileSystemHandler	
# editorBody	: String = ""
# appRoot	: String = ""
# rootName	: String = "files"
# rootFolderName	: String = "files"
# currFolder	: File = null
# fileName	: String = ""
# targetFolder	: String = ""
# clipboard	: Vector = new Vector()
# cutOrCopy	: String = ""
# status	: String = ""
# filter	: String = ""
# repository	: Vector = new Vector()
+ <<Constructor>>	FileSystemHandler ()
+ getPoruka ()	: String
+ uploadImportFile (String aFileName)	: void
+ executeDolImportFile (String dummy)	: void
# zameniEkstenziju (String fileName, String ne)	: String
# ekstenzija (String fileName)	: String
+ setFilter (String aFilter)	: void
+ getFolderContent ()	: FileInfo[]
+ getCurrentPath ()	: String
+ selectRootFolder (String rootFolderName)	: FileInfo[]
+ gotoRootFolder ()	: FileInfo[]
+ selectFolder (String folderName)	: FileInfo[]
+ getParentFolder ()	: FileInfo[]
+ rename (String oldName, String newName)	: FileInfo[]
+ createNewFolder (String newName)	: FileInfo[]
+ createNewFile (String newName)	: FileInfo[]
+ zipFiles (String fileName, String fileNamees[])	: FileInfo[]
+ unzipFile (String fileName)	: FileInfo[]
+ deleteFiles (String fileNamees[])	: FileInfo[]
+ copyFiles (String fileNamees[])	: void
+ cutFiles (String fileNamees[])	: void
+ pasteFiles ()	: FileInfo[]
+ saveFile (String fileName, String content)	: void
+ makeFullBackup ()	: String
+ runJar (String path)	: String
+ activateJar (String path)	: String
+ makeFullRestore (String fileName)	: String
+ getFileName ()	: String
+ getTargetFolder ()	: String
+ setTargetFolder (String targetFolder)	: void
+ getCurrFolder ()	: File
+ getRootFolderName ()	: String
+ getEditorBody ()	: String
+ setEditorBody (String editorBody)	: void

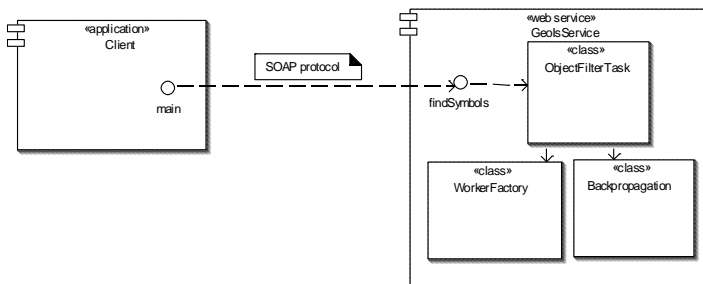
Slika 3.27: Klasa *FileSystemHandler*

U prezentacionom delu na stranicama koristi se tag *floatingForm* kao i posebno razvijene kontrole poput *fileSystem.jsp contextMenu.jsp yesNoDialog.jsp*. U datoteci *contextMenu.jsp* implementirane su funkcije i korisnički interfejs preko kojeg se uspostavlja veza između klijentskog i serverskog dela.

Na slici 3.28 prikazan je dijagram sekvenci za prikaz sadržaja tekućeg foldera. Administrator započinje proces učitavanjem stranice *fileSystem.jsp*. Prilikom učitavanja stranice kreira se javascript objekat *FileSystem* kojim je opisana veza sa odgovarajućim objektom na serverskom delu aplikacije. Za generisanje javascript objekta odgovoran je dwr paket koji na osnovu konfiguracione datoteke *dwr.xml* datoteke http protokolom šalje definiciju objekta *FileSystem*. Serverska komponenta *fileSystemHandler* preko *File* klase funkcijom *listFiles* preuzima sadržaj tekućeg foldera. Lista datoteka i foldera zatim se pripremi u obliku stringa i vrati javascript objektu. Poziv funkcije *getFolderContent* je asinhron pa je potrebno kao parametar proslediti i adresu funkcije koja će se pozvati po završetku izvršavanja funkcije koja se izvršava na serveru. Za to vreme klijentu se učitava ostatak stranice i za svaki sledeći poziv učitava se samo deo stranice sa podacima.



Slika 3.28: Dijagram sekvenci za pregled sadržaja tekućeg foldera



Slika 3.29: Dijagram razmeštaja podsistema za podršku webservisima

Klijentska aplikacija u ovom primeru implementirana je kao Java Aplikacija čiji je zadatak da pozove metodu *webservisa* i da prikaže rezultat izvršavanja. Klijentska aplikacija bi mogla da bude pisana u bilo kojem programskom jeziku i mogla bi da se izvršava na bilo kojoj platformi. SOAP protokol je zasnovan na XML jeziku i http protokolu. Sa klijentske strane formira se soap poruka u kojoj se nalazi poziv metode web servisa. Axis paket prihvata poruku i putem lanca *Handlera* vrši obradu poruke. po završetku obrade instancira se klasa *GeoIsService* čije metode predstavljaju metode web servisa.

Web servis prisupa resursima pomoću klase *WMSImage* koja preko WMS standarda preuzima mape sa *geoservera*.

U nastavku je dat deo koda kojim se opisuje metoda web servisa (listing 3.23). Na primeru se vidi na koji način servis pristupa resursima koji se nalaza na serveru.

```

public String findSymbols(String map, String params[])
{
    String retVal = "";
    try {
        prepareWorkerFactory();
        ObjectFilterTask task = new
            ObjectFilterTask();
        retVal = task.findSymbols(map, params);
    } catch (Exception e) {
        e.printStackTrace();
    }
    return retVal;
}

```

Listing 3.23: Metoda web servisa

Metoda klijenta kojom se poziva izvršavanje metode web servisa prikazana je na listingu 3.24.

```

String endpoint = "http://localhost:8080/geoIs/GeoIsService.jws";
String map = "maps/6_znakovi_2.bmp";

```

```

String znakovi[] = {"marks/25_znak1.bmp",
                   "marks/25_znak2.bmp"};
Service service = new Service();
Call call = (Call) service.createCall();
call.setTargetEndpointAddress( new
                               java.net.URL(endpoint) );
call.setOperationName( "findSymbols" );
call.addParameter( "op1", XMLType.XSD_STRING,
                  ParameterMode.IN );
call.addParameter( "op2", XMLType.XSD_ANYTYPE,
                  ParameterMode.IN );
call.setReturnType( XMLType.XSD_STRING );
String ret = (String) call.invoke( new Object []
                                  { map, znakovi });
System.out.println("Got result : " + ret);

```

Listing 3.24: Metoda klijenta kojom se poziva web servis

Ako je klasi *Task* prilikom izvršavanja neophodno da izvrši neki od servisa definisanih klasama *Worker* klasa *Task* prvo preuzme instancu servisa koji treba da se izvrši i potom pokrene njegovu *start* metodu.

Primer konfiguracione datoteke koja se odnosi na rad sa bazom podataka dat je u nastavku

```

#####
#           MySQL
#####
rdbms=MySQL
jdbcDriver=com.mysql.jdbc.Driver
jdbcURL=jdbc:mysql://localhost/geois?user=geois&password=geois&autoRe
connect=true&useUnicode=true&characterEncoding=UTF-8
#####
#           Oracle
#####
#rdbms=Oracle
#jdbcDriver=oracle.jdbc.driver.OracleDriver
#jdbcURL=jdbc:oracle:thin:@obrad:1521:GEOIS
#####
#           SAP DB
#####
#rdbms=SapDB
#jdbcDriver=com.sap.dbtech.jdbc.DriverSapDB
#jdbcURL=jdbc:sapdb://localhost/GEOIS?unicode=yes&timeout=0

#####
#           Common Db properties
#####
username=geois
password=geois
maxConnections=10
preconnectCount=5
maxIdleConnections=5

```

Listing 3.25: Konfiguraciona datoteka *Environment.properties*

Podsistem za rad sa prostornim procesima i simulacijama implementiran je kao proširiv sistem u kojem je moguće dodavati definicije procesa kao posebne *.jar* biblioteke klasa. Implementirani podsistem ne zavisi od korištene platforme, aplikativnog servera niti od korištenog sistema za rukovanje podacima.

GMLToJava i GMLToCSharp podsistem

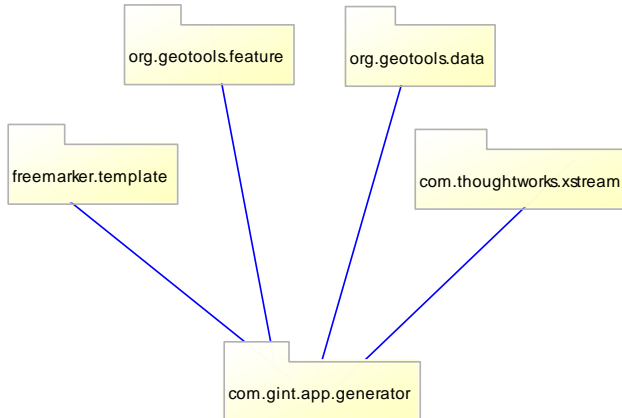
U geografskim informacionim sistemima integracija podataka je veoma čest problem. Različiti sistemi koriste po pravilu različite načine reprezentacije geoprostornih podataka. Razvojem infrastrukture prostornih podataka postoji veoma izražena potreba da se geoprostorni podaci učine dostupnijima. Razvoj standarda u ovoj oblasti pomaže da se implementiraju alati za jedinstven pristup ovoj vrsti podataka kao i servisima koji podatke obezbeđuju. GML (*Geography Markup Language*) i WFS (*Web Feature Service*) standardi predstavljaju osnovu za integraciju više sistema koji koriste geoprostorne podatke. Međutim i dalje postoji problem veoma kompleksnih procedura za transformaciju tako opisanih objekata u Java i C# objekte, odnosno objekte pomoću kojih je moguće izvršiti implementaciju softvera za opis složenih geoprostornih procesa.

Postoje dva načina transformacije: (i) transformacija podataka u generički objekat (*Feature*) koji ima dinamički definisane atribute; (ii) konkretna (*POJO-Plain Old Java Object*) Java klasa sa statički definisanim atributima.

Osnovna transformacija podrazumeva transformaciju geoprostornih podataka opisanih GML gramatikom u Java objekat. Ova transformacija odvija se posredstvom klasa biblioteke *geotools* u kojima se kreira instanca klase *Feature*. Ona sadrži dinamičke atribute pojedinačnih klasa geoprostornih podataka. Pored ovog objekta kreira se i *FeatureType* objekat koji u sebi sadrži definiciju i opise pojedinačnih atributa.

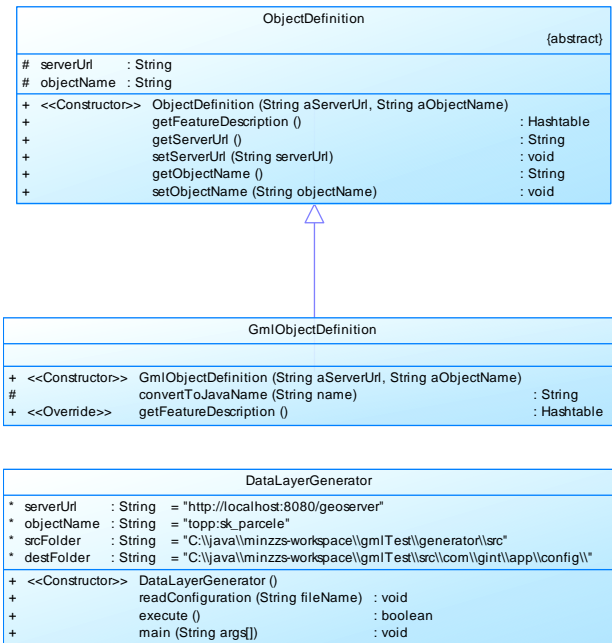
Podsistem *GMLToJava* koristi *FreeMarker* i *XStream* biblioteke. Pored generisanja POJO Java klase generiše se i *JavaService* klasa koja vrši transformaciju *Feature* objekta na POJO Java objekat.

Osnovni paket podsistema za generisanje je *com.gint.app.generator* (Slika 3.30) koji koristi pakete *freemarker.template*, *org.geotools.feature*, *org.geotools.data* i *com.thoughtworks.xstream*.



Slika 3.30: Dijagram paketa generatora

Na slici 3.31 prikazan je dijagram osnovnih klasa paketa *com.gint.app.generator*.

Slika 3.31: Dijagram klasa paketa *com.gint.app.generator*

ObjectDefinition klasa je apstraktna klasa koja služi kao osnova za implementaciju pojedinačnih konkretnih klasa za preuzimanje definicije objekata. *GmlObjectDefinition* klasa implementira metodu

`getFeatureDescription` u kojem se vrši priprema modela podataka potrebnih klasi `DataLayerGenerator` koja poziva `freemarker` biblioteku i generiše za svaki objekat po jednu POJO Java klasu i `Service` Java klasu. U metodi `readConfiguration` preuzima se konfiguracija iz XML konfiguracione datoteke koja sadrži informacije o url adresi konkretnog geoservera (`serverUrl`), naziv grupe geoprostornih podataka (`objectName`), direktorijum u kojem se nalaze šabloni (`srcFolder`) i direktorijum u koji će se generisati rezultujuće datoteke (`destFolder`).

Primer korištenja

Na listingu 3.26. dat je deo koda metode `execute` u kojoj se vrši generisanje POJO i `Service` Java klasa.

```
// inicijalizacija modela podataka
GmlObjectDefinition gml = new
    GmlObjectDefinition(serverUrl, objectName);
Hashtable ht = gml.getFeatureDescription();
Configuration cfg = new Configuration();

cfg.setDirectoryForTemplateLoading( new File(srcFolder));
// inicijalizacija POJO sablona
Template temp = cfg.getTemplate("GmlEntity.java");

String nazivKlase = (String)ht.get("nazivKlase");
File destFolderFile = new
    File(destFolder+nazivKlase+".java");
FileOutputStream fo = new
    FileOutputStream(destFolderFile);
Writer out = new OutputStreamWriter(fo);
//generisanje POJO Java klase
temp.process(ht, out);
out.flush();
fo.close();

// inicijalizacija Service sablona
temp = cfg.getTemplate("GmlEntityService.java");
File srcFolderFile = new
    File(destFolder+nazivKlase+"Service.java");
fo = new FileOutputStream(srcFolderFile);
out = new OutputStreamWriter(fo);
// generisanje Service Java klase
temp.process(ht, out);
out.flush();
fo.close();
```

Listing 3.26: Deo metode *execute* u kojoj se vrši generisanje POJO i Service Java klasa

GML opis geoprostornog podatka *sk_parcele* dat je u listingu 3.27.

```
<xsd:schema
    elementFormDefault="qualified"
    targetNamespace =
        "http://www.openplans.org/topp">
<xsd:import
    namespace="http://www.opengis.net/gml"
    schemaLocation="http://localhost:8080/geoserver/schemas/gml/3.
1.1/base/gml.xsd"/>
<xsd:complexType name="sk_parceleType">
<xsd:complexContent>
<xsd:extension
    base="gml:AbstractFeatureType">
<xsd:sequence>
<xsd:element
    maxOccurs="1" minOccurs="0"
    name="wkb_geometry"
    nillable="true"
    type="gml:GeometryPropertyType"/>
<xsd:element maxOccurs="1" minOccurs="0"
    name="area" nillable="true"
    type="xsd:double"/>
<xsd:element
    maxOccurs="1" minOccurs="0"
    name="perimeter" nillable="true"
    type="xsd:double"/>
<xsd:element
    maxOccurs="1" minOccurs="0"
    name="id" nillable="true"
    type="xsd:double"/>
<xsd:element
    maxOccurs="1" minOccurs="0"
    name="kultura" nillable="true"
    type="xsd:string"/>
</xsd:sequence>
</xsd:extension>
</xsd:complexContent>
</xsd:complexType>
<xsd:element name="sk_parcele"
    substitutionGroup="gml:_Feature"
    type="topp:sk_parceleType"/>
</xsd:schema>
```

Listing 3.27: GML opis *sk_parcele* objekta

Na slici 3.32 prikazan je dijagram klasa koje se kreiraju na osnovu GML opisa datog u listingu 3.27.

SkParcele		
+ fullName	: String	= "topp:sk_parcele"
- ID	: String	
- wkb_geometry	: Geometry	
- area	: Double	
- perimeter	: Double	
- id	: Double	
- kultura	: String	
+ <<Getter>>	getID ()	: String
+ <<Setter>>	setID (String aID)	: void
+	getWkb_geometry ()	: Geometry
+	setWkb_geometry (Geometry wkb_geometry)	: void
+	getArea ()	: Double
+	setArea (Double area)	: void
+	getPerimeter ()	: Double
+	setPerimeter (Double perimeter)	: void
+	getId ()	: Double
+	setId (Double id)	: void
+	getKultura ()	: String
+	setKultura (String kultura)	: void

SkParceleService		
* serverUrl	: String	= "http://localhost:8080/geoserver"
* objectName	: String	= "topp:sk_parcele"
+ findAll ()		: Vector<SkParcele>
+ create (SkParcele obj)		: void
+ update (SkParcele obj)		: void
+ delete (SkParcele obj)		: void
+ getServerUrl ()		: String
+ setServerUrl (String serverUrl)		: void
+ getObjectname ()		: String
+ setObjectName (String objectName)		: void

Slika 3.32: Dijagram generisanih klasa

U listingu 3.28 prikazan je deo u kojem se vrši transformacija `Feature` objekta u `SkParcele` objekat.

```
Feature feature = ft.next();
SkParcele sk = new SkParcele();
sk.setID(feature.getID());
sk.setWkb_geometry((Geometry)
    feature.getAttribute("wkb_geometry"));
sk.setArea((Double) feature.getAttribute("area"));
sk.setPerimeter((Double)
    feature.getAttribute("perimeter"));
sk.setId((Double) feature.getAttribute("id"));
sk.setKultura((String) feature.getAttribute("kultura"));
```

Listing 3.28. deo izvornog koda u kojem se vrši mapiranje

Deo šablona na osnovu kojeg je nastao ovaj kod prikazan je u listingu 3.29.

```
Feature feature = ft.next();
${nazivKlase} sk = new ${nazivKlase}();

<#list atributi as bp>
sk.set${bp.naziv?cap_first} (${bp.tip}
    feature.getAttribute("${bp.naziv}"));
</#list>
```

Listing 3.29. Deo šablona na osnovu kojeg je generisan listing 3.30

Primer korišćenja generisanih klasa u sloju poslovne logike dat je u listingu 3.30.

```
SkParceleService service = new
    SkParceleService();
// preuzimanje svih elemenata iz
// sloja za pristup podacima
Vector<SkParcele> all = service.findAll();
for (SkParcele skParcele : all)
{
    String id = skParcele.getID();
    String kultura = skParcele.getKultura();

    System.out.println(id+ " "+kultura);

    // promena vrednosti i snimanje
    skParcele.setKultura("nova vrednost");
    service.update(skParcele);
}
```

Listing 3.30. Primer korišćenja generisanih klasa

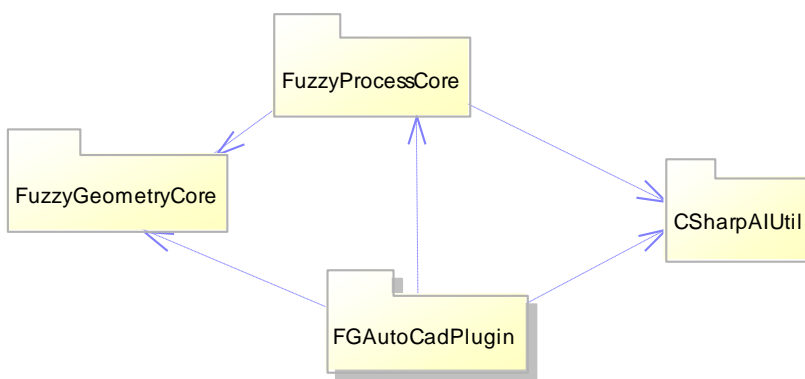
Korišćenjem podsistem za transformaciju geoprostornih podataka u objekte implementacija modela procesa postaje nezavisna od korištene tehnologije za obezbeđenje geoprostornih podataka. Klase koje se nalaze u delu za modeliranje procesa umesto da koriste kompleksan *Feature* objekat i preko metode *getAttribute* preuzimaju vrednosti, rade sa POJO Java objektima i pozivanjem *get* metoda preuzimaju konkretne vrednosti. Ovim pristupom se prilikom kompajliranja obezbeđuje kontrola tipova. Podsistem poseduje jednostavne mehanizme za uključivanje novih modela podataka. Samo dodavanjem novih šablona mogu se dodavati nove funkcionalnosti kao što je kreiranje klasa za automatizovano testiranje i dokumentovanje. Odgovarajućom implementacijom apstraktne klase *ObjectDefinition* generatoru se mogu obezbediti različiti modeli podataka, odnosno jednostavno rukovanje podacima iz različitih izvora.

AutoCad proširenje

Implementacija AutoCad proširenja za rad sa nepreciznim geoprostornim podacima izvršena je u Visual Studio 2010 okruženju u programskom jeziku C# u skladu sa specifikacijom prikazanom u prethodnom poglavlju. Programski jezik C# izabran je zbog svoje objektne prirode i jednostavne integracije sa postojećim paketima kao što su *AutoCAD.Net API* i *ObjectDBX.Net*.

AutoCad je zamišljen kao samostalan sistem sa svojim sistemom za rukovanje podacima interfejsom za prikazivanje i okruženjem u kojem je moguće inkorporirati nove funkcionalnosti. Realizovani su alati za import i eksport podataka iz raznih izvora geometrijskih podataka i u različita odredišta. AutoCADov mehanizam za čuvanje podataka nije namenjen uslovima u kojima više korisnika radi nad istim podacima pa je neophodno tu funkcionalnost implementirati.

Podrška za proširenje AutoCad programskog paketa delom za upravljanje izvršavanjem geoprostornog procesa implementirano je paketom *FGAutoCadPlugin*. Ovaj paket koristi prethodno implementirane pakete *FuzzyGeometryCore*, *FProcessCore* i *CSharpAIUtil*. Pored ovih paketa koji su u direktnoj vezi koristi se i podsistem *GMLToCSharp* kao pomoć prilikom povezivanja sa podsistemom za skladištenje podataka. Proširenje AutoCAD-a moguće je na dva načina: (i) implementiranjem skupa komandi koje se izvršavaju preko komandne linije i (ii) implementacija palete akcija preko koje se funkcije mogu pozivati i izvršavati.

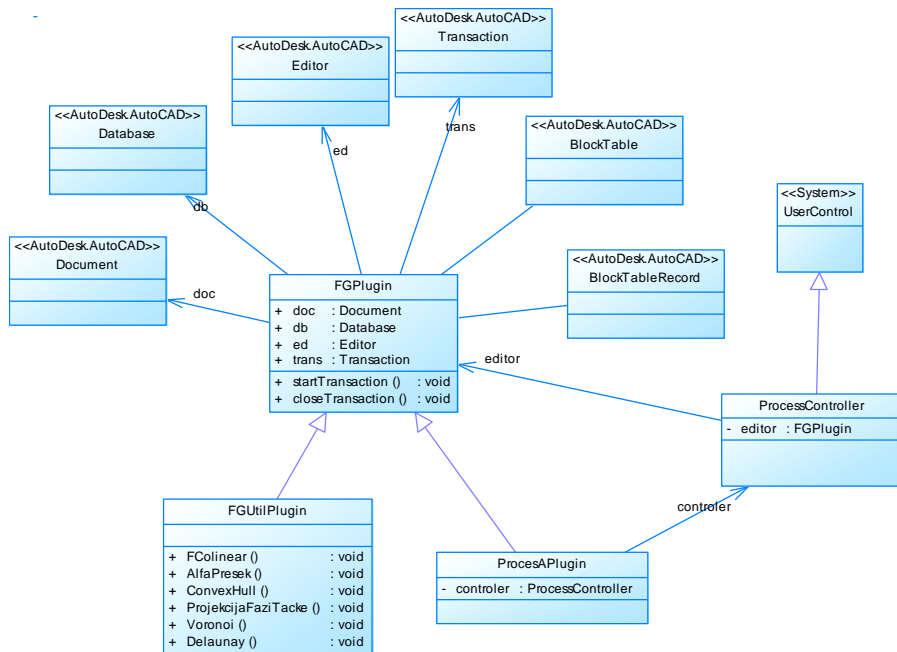


Slika 3.33: Dijagram povezanosti paketa *FGAutoCadPlugin*

Podsystem *GMLToCSharp* iskorišten je za generisanje dela aplikacije koji je namenjen za preuzimanje podataka iz PostGIS baze, kreiranje forme za uređivanje sadržaja i automatsko kreiranje *bloka* za reprezentaciju samog realnog objekta.

Prvi korak prilikom implementacije AutoCAD proširenja je da se podsystemom *GMLToCSharp* izgeneriše skup klasa koje grubo predstavljaju geometrijski objekat sačuvan u PostGIS bazi.

Nakon toga potrebno je ručno prilagoditi generisane klase konkretnoj primeni ili problemu i na kraju povezati ga sa ostatkom *AutoCadPlugin-a*.



Slika 3.34: Dijagram povezanosti klasa paketa *FGAutoCadPlugin*

Dijagram povezanosti klasa paketa *FGAutoCadPlugin* prikazana je na slici 3.34. Osnovu proširenja čini klasa *FGPlugin*. osnovni atributi klase su:

- *doc:Document* - veza sa AutoCAD osnovnim objektom *Document*.
- *db:Database* - interfejs prema AutoCad internoj bazi podataka
- *editor:Editor*-interfejs prema osnovnim funkcijama ekrana
- *trans:Transaction* -interfejs prema trenutnoj transakciji

Sve funkcije koje čine proširenje AutoCAD-a imaju veoma sličan obrazac ponašanja koji se svodi na sledeće:

- prvo se inicijalizuje *doc* promenjiva i preko nje se uspostavi veza sa AutoCAD-om
- nakon toga se preuzmu *Database* i *Editor* reference na odgovarajuće objekte
- pozivom funkcije *startTransaction* kreira se transakcija unutar koje moraju da se izvrše sve operacije sa AutoCAD internom bazom
- funkcija ili izvrši akcije dodavanja i menjanja objekta iz baze
- po završetku pozove se funkcija *closeTransaction* nakon čega se sve promene prikažu u AutoCAD editoru.

Klasa *ProcessController* je klasa koja ima funkciju palete alata a preko nje i korisničke forme za interakciju između složenih procesa i korisnika. Klasa je implementirana kao proširenje *UserControl* klase na isti način kao što se kreiraju bilo koje Windows korisničke kontrole sa dodatkom veze prema *FGPlugin* objektom. Preko ove klase *ProcessController* mogućnosti kao i ostali plugin objekti.

FGUtilPlugin	
+ FColinear ()	: void
+ AlfaPresek ()	: void
+ ProjekcijaFaziTacke ()	: void
+ ConvexHull ()	: void
+ Voronoi ()	: void
+ Delaunay ()	: void

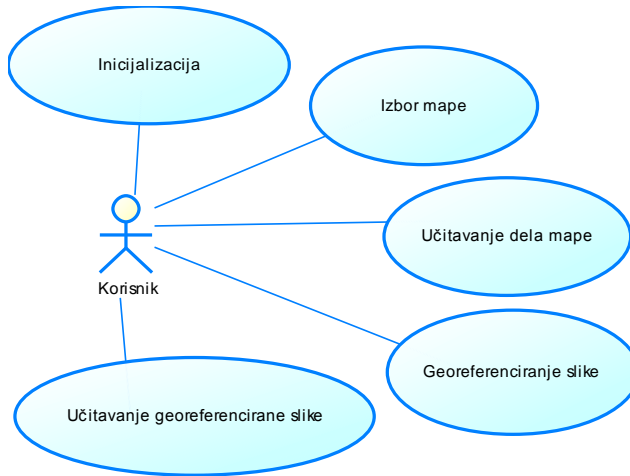
Slika 3.35: Klasa FGUtilPlugin

U klasi *FGUtilPlugin* implementirane su neke od osnovnih pomoćnih funkcija poput funkcije za određivanje stepena kolinearnosti tri tačke i ilustraciju rezultata, ilustracija funkcija pripadnosti za fazi tačku, fazi duž, fazi trougao i fazi krug i njihove alfa preseke. Pored ovih funkcija implementirane su i funkcije kojima su demonstrirane mogućnosti specifičnih algoritama računarske geometrije poput Voronoi dijagram skupa tačaka ili Delaunay triangulacije. Funkcije se pozivaju iz komandnog prozora AutoCAD editora komandama koje imaju ista imena kao i odgovarajuće funkcije.

Implementirane su i dve palete alata: *WMSController* i *FPostGisController*.

Preko palete alata *WMSController* implementirana je podrška za komunikaciju između *WMS servera (geoserver)* i *AutoCAD-a*.

Na slici 3.36 prikazan je dijagram slučajeva korišćenja palete alata *WMSController*.



Slika 3.36: Dijagram korišćenja palete *WMSController*

Slučaj korišćenja Inicijalizacija

Opis: Korisnik preko korisničkog interfejsa definiše osnovne parametre za povezivanje.

Okidač: Paleta se pokreće komandom *wms*.

Završni uslovi: Po završetku definisani su osnovni parametri za povezivanje:

- adresa WMS servera
- horizontalni i vertikalni broj piksela mape koja se preuzima sa servera.

Osnovni tok: Korisnik pokrene proširenu aplikaciju AutoCAD. Komandom *wms* pokreće se paleta alata. Prvo se unosi adresa WMS servera a potom se pokretanjem forme za određivanjem horizontalnog i vertikalnog broja piksela definiše preostali deo parametara.

Slučaj korišćenja **Izbor mape**

Opis: Preuzimanje liste mapa slanjem WMS zahteva *GetCapabilities*. U odgovoru koji na zahtev WMS server vrati nalaze se i podaci koji opisuju svaku pojedinačnu mapu.

Okidač: Paleta se pokreće komandom *wms*.

Završni uslovi: Po završetku u listi *layers* nalazi se lista raspoloživih mapa.

Osnovni tok: Korisnik pokrene proširenu aplikaciju AutoCAD. Komandom *wms* pokreće se paleta alata. Klikom na dugme *učitaj mape* pripremi se WMS zahtev sa komandom *GetCapabilities* i šalje WMS serveru. WMS server odgovara XML dokumentom u kojem su opisane raspoložive mape. Opis mapa je u skladu sa WMS standardom.

Slučaj korišćenja **Učitavanje dela mape**

Opis: Preuzimanje dela mape vrši se slanjem WMS zahteva *GetMap*. U odgovoru WMS server vraća sliku dela mape u izabranom formatu (jpeg, png, tiff, bmp) i dimenzijama (horizontalni i vertikalni broj piksela).

Okidač: Paleta se pokreće komandom *wms* a učitavanje dela mape aktiviranjem dugmeta *Get map*.

Završni uslovi: Po završetku slučaja korišćenja slika dela mape prikaže se na radnoj površini AutoCAD-a.

Osnovni tok: Korisnik pokrene proširenu aplikaciju AutoCAD. Komandom *wms* pokreće se paleta alata. Potom se odredi gornja leva i donja desna granična tačka, a zatim se aktiviranjem dugmeta *Get map* generiše *GetMap* zahtev u skladu sa WMS standardom. Po dobijanju odgovora od WMS servera slika se prikaže na radnoj površini AutoCAD-a.

Slučaj korišćenja **Georeferenciranje slike**

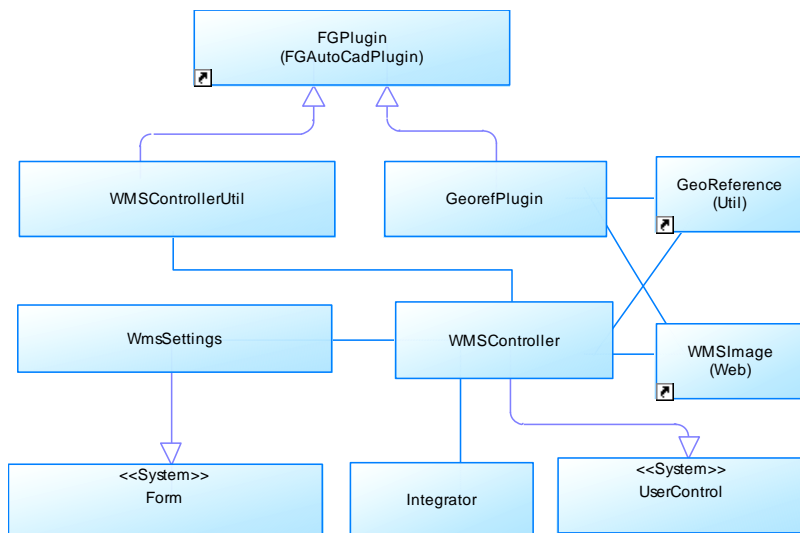
Opis: Georeferenciranje slike u AutoCAD-u.

- Okidač:* Slučaj korišćenja poziva se komandom *georef*.
- Završni uslovi:* Po završetku slučaja korišćenja u file sistemu kreirana je slika i odgovarajuća (tfw ili jgw) georeferentna tekstualna datoteka.
- Osnovni tok:* Pomeranjem i skaliranjem slike u AutoCAD radnoj površini odredi se tačan položaj i oblik slike a potom se pozivanjem komande *georef* uz sliku sačuvaju georeferentni podaci. Slika i georeferentni podaci o mapi preuzetoj preko WMS servera mogu se snimiti direktno bez pomeranja i skaliranja.

Slučaj korišćenja Učitavanje georeferencirane slike

- Opis:* Učitavanje georeferencirane slike u AutoCAD okruženje.
- Okidač:* Slučaj korišćenja poziva se komandom *addMap*.
- Završni uslovi:* Po završetku slučaja korišćenja na AutoCAD radnoj površini pojavi se na odgovarajućem mestu i obliku georeferencirana mapa.
- Osnovni tok:* Korisnik pozivom komande *addMap* pokrene formu za izbor datoteke u kojoj korisnik odredi sliku a potom se na osnovu odgovarajuće georeferentne datoteke odredi pozicija i veličina slike. Potom se objekat koji reprezentuje mapu prijavi AutoCAD bazi.

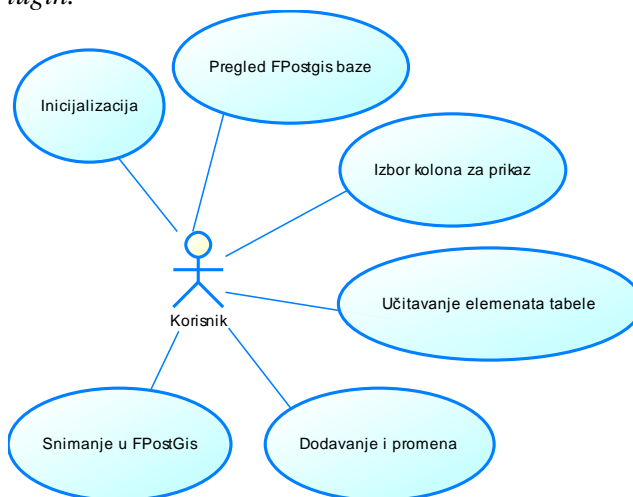
Na slici 3.37 prikazan je dijagram klasa kojim su imeplemtirani slučajevi korišćenja *Inicijalizacija*, *Izbor mape* i *Učitavanje dela mape*.



Slika 3.37: Dijagram klasa paketa WMSController

Klasa *WMSController* predstavlja implementaciju palete alata za upravljanje osnovnim poslovima u komunikaciji sa WMS serverom. U klasama *WMSControllerUtil* i *GeorefPlugin* implementirani su konkretni poslovi vezani za AutoCAD bazu. *Integrator* klasa je statična i koristi se za povezivanje alata *WMSControllerUtil* i ostalih paleta alata.

Na slici 3.38 prikazan je dijagram slučajeva korišćenja palete alata *FPostGisPlugin*.



Slika 3.38: Dijagram slučajeva korišćenja paketa *FPostGisController*

Slučaj korišćenja Inicijalizacija

Opis: Postavljanje osnovnih parametara.

Okidač: Paleta alata paketa *FPostGisController* poziva se komandom *fPostGis*. Potrebno je da se ovaj slučaj izvrši pre svih ostalih.

Završni uslovi: Definisani osnovni parametri za povezivanje sa FPostGis bazom.

Osnovni tok: Korisnik pozivom komande *fPostGis* pokrene paletu alata *FPostGisController*. Potom aktiviranjem dugmeta *Init* pokreće formu za unos osnovnih parametara:

- server
- port
- user
- password
- database

neophodnih za povezivanje sa FPostGis bazom.

Slučaj korišćenja Pregled FPostGis baze

Opis: Pregled svih javnih tabela u FPostGis bazi.

Okidač: Paleta alata paketa *FPostGisController* poziva se komandom *fPostGis*. Korisnik je završio sa *inicijalizacijom*.

Završni uslovi: U listi za prikaz svih javnih tabela prikazani su nazivi svih dostupnih tabela.

Osnovni tok: Korisnik pozivom komande *fPostGis* pokrene paletu alata *FPostGisController*. Nakon inicijalizacije aktiviranjem dugmeta *Public tables* kreira se SQL iskaz za pregled svih javnih tabela. Potom se po prijemu *DataTable* objekta popunjava lista svih javnih tabela.

Slučaj korišćenja Izbora kolona za prikaz

- Opis:* Pregled svih atributa izabrane tabele u FPostGis bazi.
- Okidač:* Paleta alata paketa *FPostGisController* poziva se komandom *fPostGis*. Korisnik je završio sa *Pregledom FPostGis baze*.
- Završni uslovi:* U listi za prikaz detalja o selektovanoj tabeli prikazane su sve kolone (naziv kolone i tip podataka).
- Osnovni tok:* Korisnik pozivom komande *fPostGis* pokrene paletu alata *FPostGisController*. Nakon pregleda svih javnih tabela izborom željene tabele formira se SQL iskaz za pregled detalja o tabeli. Potom se po prijemu *DataTable* objekta popunjava lista detalja o selektovanoj tabeli. Posebno su obeležene kolone kojima su predstavljeni neprecizni geoprostorni podaci *fuzzyPoint* i *fuzzyLine*.

Slučaj korišćenja Učitavanje elemenata tabele

- Opis:* Pregled svih elemenata izabrane tabele. Prikaz samo obeleženih kolona. *FuzzyPoint* i *fuzzzLine* attribute prikazuje u skladu sa definicijama datim u prethodnom poglavlju.
- Okidač:* Paleta alata paketa *FPostGisController* poziva se komandom *fPostGis*. Korisnik je završio sa *Izborem kolona za prikaz*.
- Završni uslovi:* Na radnoj površini AutoCad okruženja prikazani su izabrani geoprostorni objekti reprezentovani fazi tačkama ili fazi linijama.
- Osnovni tok:* Korisnik pozivom komande *fPostGis* pokrene paletu alata *FPostGisController*. Nakon izbora kolona za prikaz formira se SQL iskaz za preuzimanje elemenata selektovane tabele. Potom se po prijemu *DataTable* objekta kreiraju *BlockReference* objekti u skladu sa definicijama nepreciznih geoprostornih podataka i prijavljuju se AutoCad bazi elementa.

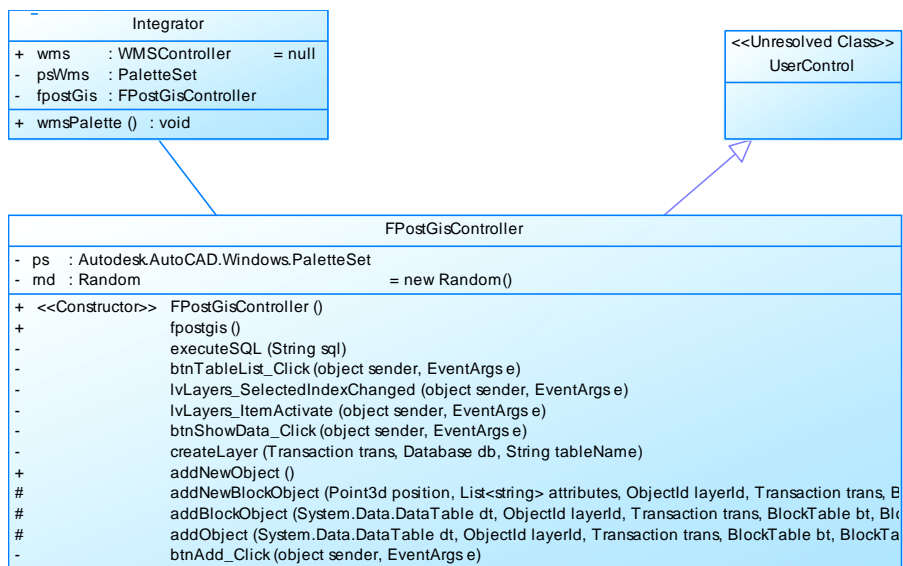
Slučaj korišćenja **Dodavanje i promena**

- Opis:* Dodavanje novog elementa i promena postojećih elemenata.
- Okidač:* Paleta alata paketa *FPostGisController* poziva se komandom *fPostGis*. Korisnik je završio sa *Učitavanjem elemenata tabele*.
- Završni uslovi:* U AutoCad bazi objekata nalaze se izmenjene verzije modela geoprostornih objekata.
- Osnovni tok:* Korisnik pozivom komande *fPostGis* pokrene paletu alata *FPostGisController*. Nakon učitavanja elemenata obeležene tabele korisnik može da doda novi element izborom pozicije centra fazi tačke i izborom odgovarajućeg poluprečnika za slučaj geoprostornih objekata reprezentovanih fazi tačkama.

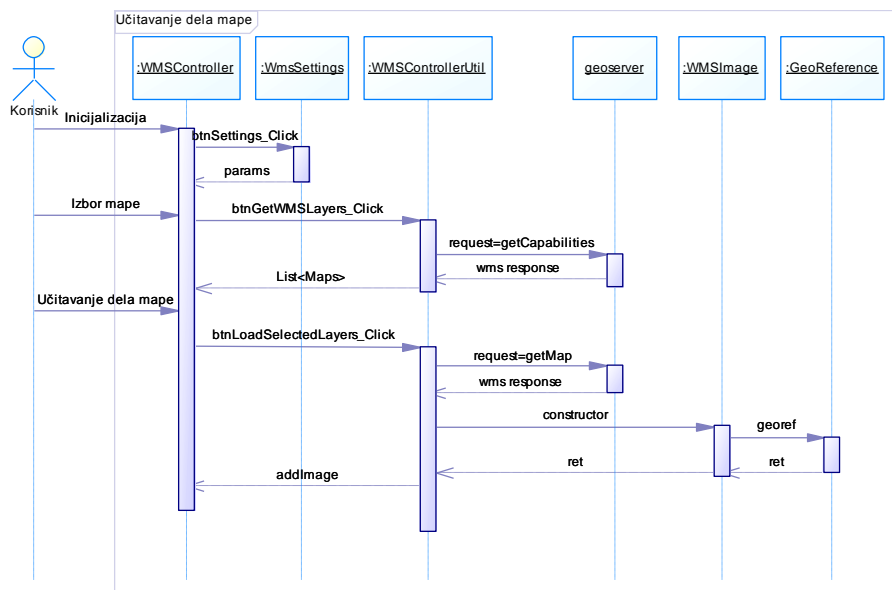
Slučaj korišćenja **Snimanje u FPostGis**

- Opis:* Snimanje izmenjenih elementa u FPostGis bazi.
- Okidač:* Paleta alata paketa *FPostGisController* poziva se komandom *fPostGis*. Korisnik je završio sa *Dodavanjem i promenom*.
- Završni uslovi:* U FPostGis bazi nalaze se izmenjene verzije modela geoprostornih objekata.
- Osnovni tok:* Korisnik pozivom komande *fPostGis* pokrene paletu alata *FPostGisController*. Nakon dodavanja i promene elemenata korisnik pozivom akcije za snimanje elemenata pokreće proces kreiranja SQL iskaza za snimanje elemenata u FPostGis bazu. Ovom prilikom koriste se funkcije paketa FPostGis za tekstualnu reprezentaciju nepreciznih geoprostornih podataka (WKT oblik)

Na slici 3.39 prikazan je dijagram osnovnih klasa paketa *FPostGisController*

Slika 3.39: Dijagram klasa paketa *FPostGisController*

Na slici 3.40 prikazan je dijagram sekvenci za učitavanje dela mape.

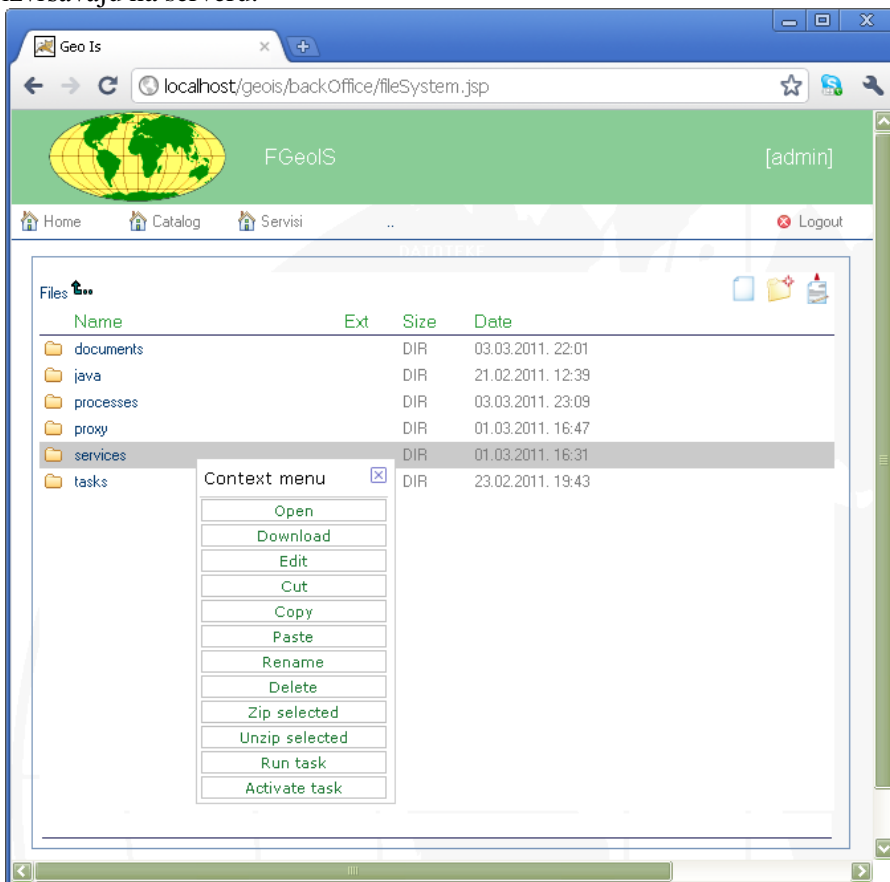


Slika 3.40: Dijagram sekvenci za učitavanje dela mape

Opis korišćenja implementiranih podsistema

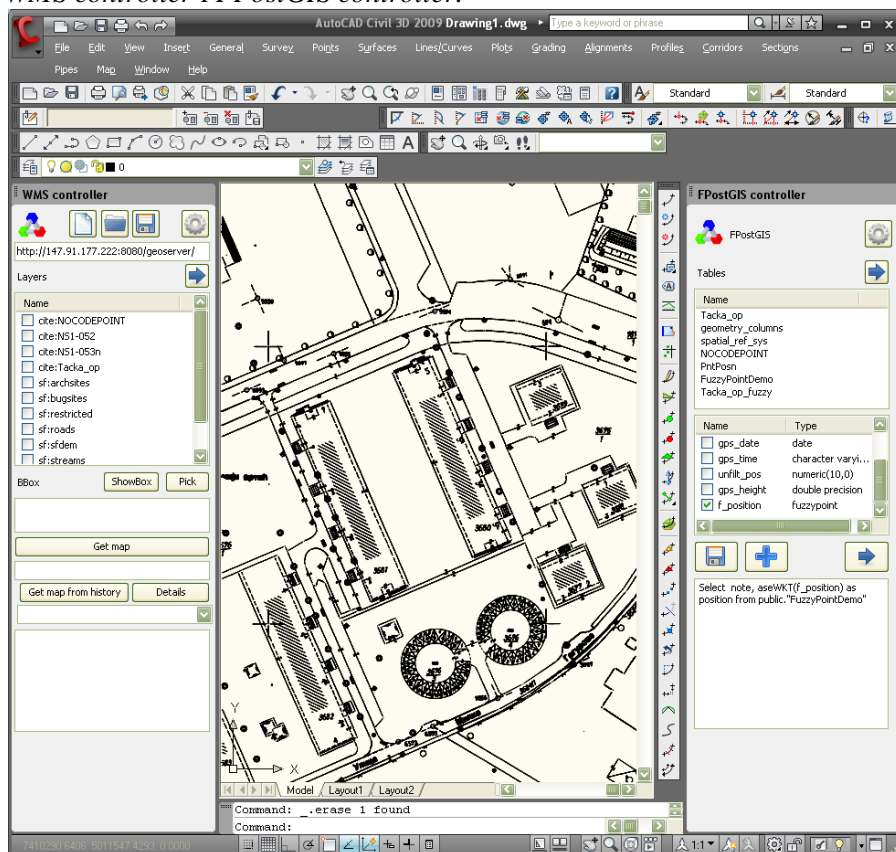
Na slici 3.41 prikazan je prikaz prozora za upravljanje *FGeoIS* podsistemom.

Na slici 3.41 su prikazani osnovni katalogi i kontekstni meni za uređivanje kataloga. U katalogu *services* nalaze se svi raspoloživi web servisi zajedno sa pratećom dokumentacijom a u *proxy* katalogu nalaze se klase za povezivanje između C# ili Java klijenata i odgovarajućeg web servisa.. Korisnik odnosno programer koji implmentira klijentsku aplikaciju koja koristi usluge web servisa treba da preuzme proxy klasu uključi je u projekta i koristi funkcije proxy klase je kao da su lokalna funkcija iako se izvršavaju na serveru.



Slika 3.41: Izgled stranice za upravljanje *FGeoIS* podsistemom

Na slici 3.42 prikazan je AutoCAD programski paket proširen sa paletama *WMS controller* i *FPostGIS controller*.



Slika 3.42: Izgled AutoCAD proširenja sa uključenim paletama alata

Poglavlje 4

Verifikacija modela

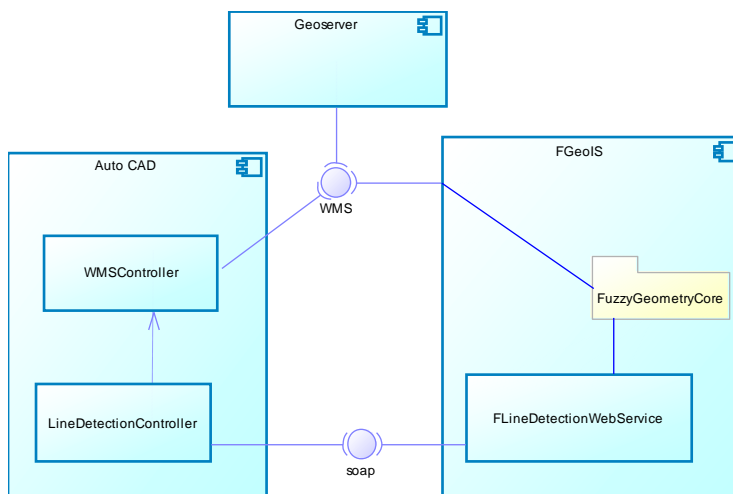
Verifikacija predstavljene arhitekture izvršena je na dva primera: detekcija fazi linija i detekcija topografskih simbola.

Osnovu za oba primera predstavlja problem izdvajanja informacija na osnovu rasterskih slika. U prvom primeru predložen je novi algoritam za izdvajanje fazi duži dok je u drugom primeru za potrebe detekcije topografskih znakova implementirana biblioteka klasa za modeliranje i simulaciju neuronska mreža. Pozicije prepoznatih znakova predstavljene su fazi tačkama. Mera nepreciznosti pozicije prepoznatog znaka određena je aproksimacijom funkcije pripadnosti koja se dobije kao izlaz iz neuronske mreže.

Detekcija fazi linija

Vektorizacija rasterskih slika problem je u kojem se na osnovu rasterske slike ili mape određuju položaji geometrijskih objekata u vektorskom obliku. U ovom primeru izvršena je implementacija procesa za detekciju fazi linija na osnovu rasterske slike. U procesu detekcije koriste se podsistemi i implementirani koncepti predstavljeni u prethodnom poglavlju.

Na slici 4.1 prikazan je dijagram komponenti podsistema za detekciju fazi linija.

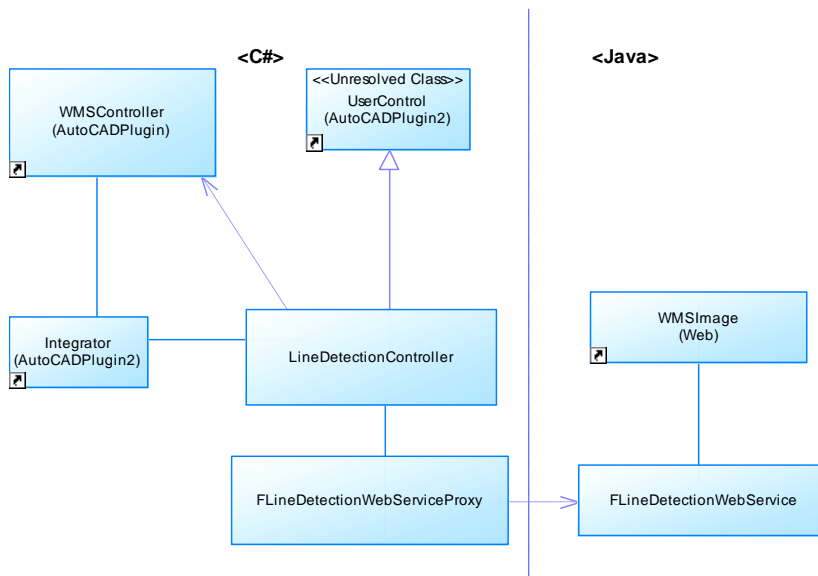


Slika 4.1: Dijagram komponenti podsistema za detekciju fazi linija

Korisnik u ovom primeru korišćenjem *WMSController* komponente (prošireni AutoCAD) definiše oblast u kojoj želi da se izvrši vektorizacija. Preko *LineDetectionController* komponente šalje *soap* zahtev *FPostGIS* podsistemu odnosno web servisu *FLineDetectionWebService*. Web servis po prijemu parametara korišćenjem klasa *FuzzyGeometryCore* paketa generiše *GetMap* zahtev geoserveru i preuzima kartu u rasterskom obliku. Java klasa *FLineDetectionWebService* koja se nalazi u *services* katalogu *FPostGIS* servera vrši detekciju i vraća listu fazi duži u WKT formatu.

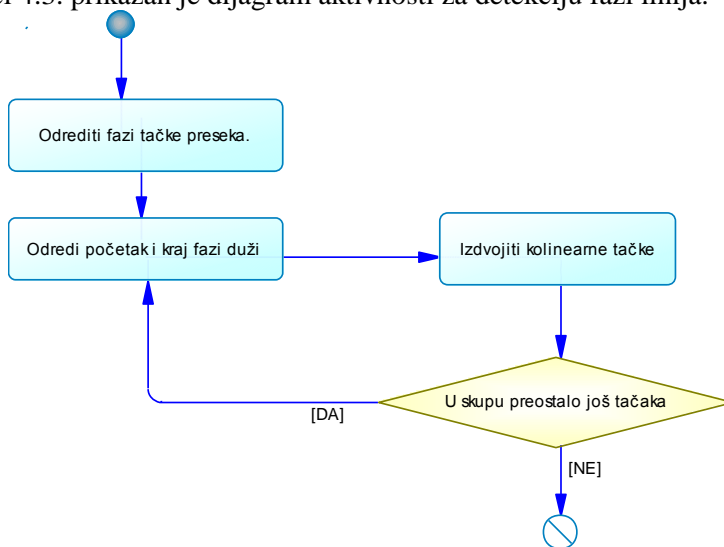
Na slici 4.2 prikazan je dijagram povezanosti klasa klijentskog i serverskog dela sistema.

Klijentski deo proširenja AutoCAD paketa implementiran je u skladu sa preporukama iz prethodnog poglavlja u C# programskom jeziku. Web servis implementiran je kao java klasa i prijavljen na serveru u katalogu *services*. Za potrebe povezivanja ova dva podsistema izgenerisana je proxy klasa (*FLineDetectionWebServiceProxy*) u C# programskom jeziku i smeštena u katalog *proxy* za potrebe integracije i korišćenja web servisa. Ovako generisana klasa potom je uključena u projekat za implementaciju AutoCAD proširenja. Na ovaj način izvršeno je jednostavno povezivanje dva podsistema realizovanih u različitim programskim okruženjima i platformama a koje su pri tom i fizički razdvojene (mogu da se izvršavaju na fizički razdvojenim računarima) .



Slika 4.2: Dijagram povezanosti klasa paketa za detekciju fazi linija

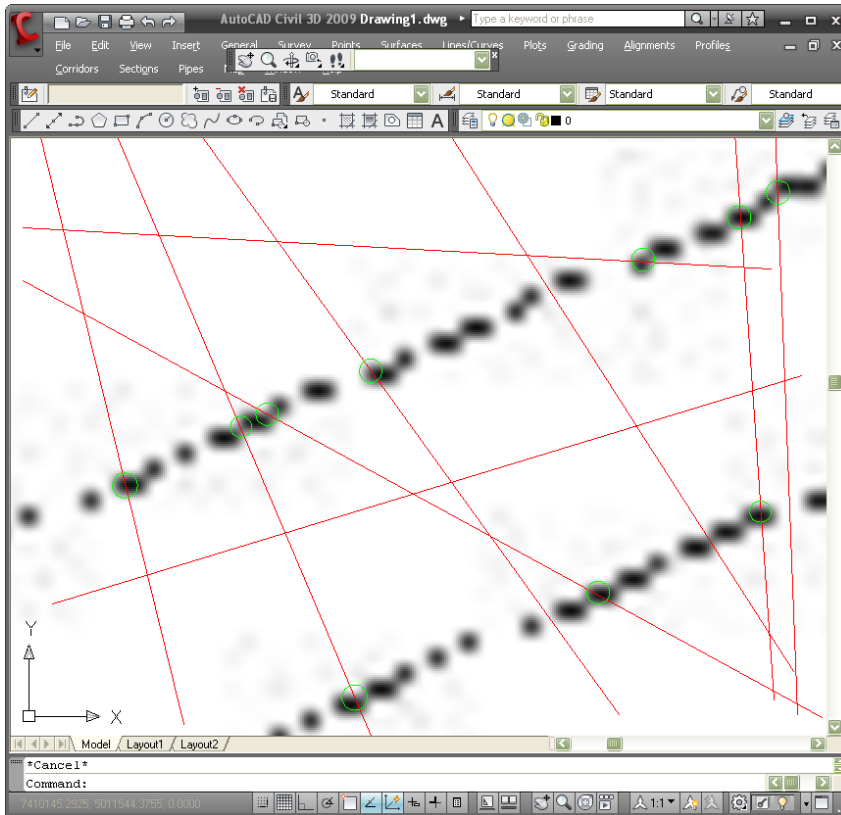
Na slici 4.3. prikazan je dijagram aktivnosti za detekciju fazi linija.



Slika 4.3: Dijagram aktivnosti za detekciju fazi linija

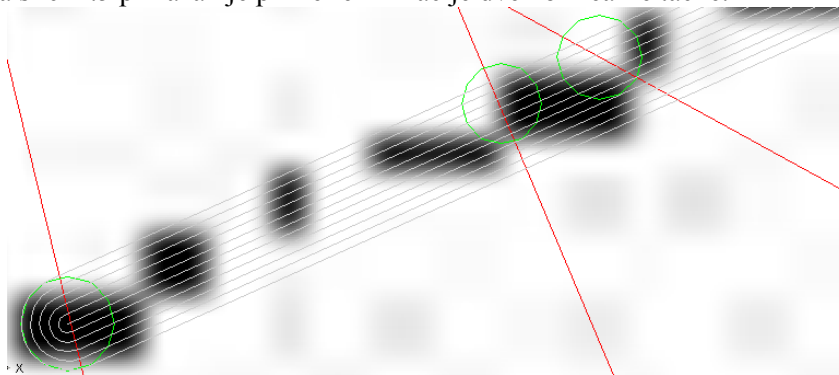
Primer

Na slici 4.4. prikazan je primer u kojem je generisano osam crvenih linija i na osnovu njih je locirano deset fazi tačaka (obeležene zelenom bojom). U koraku određivanja početka i kraja fazi duži odvoje se dve fazi tačke koje formiraju fazi duž koja u najboljoj meri aproksimira duž predstavljenu rasterskom slikom. Potom se za preostale tačke određuje u kojoj su meri kolinearne (definicija 2.41). Fazi tačke koje su kolinearne (mera kolinearnosti je različita od 0) sa tačkama koje definišu fazi duž se izbace iz skupa a nad preostalim tačkama se ponovi ceo postupak koji se ponavlja sve dok u plaznom skupu ima tačaka.



Slika 4.4: Primer određivanja polaznog skupa fazi tačaka

Na slici 4.5 prikazan je primer eliminacije dve kolinearne tačke.



Slika 4.5: Primer eliminacije dve kolinearne tačke

Na slici se vidi da dve fazi tačke koje su pridružene grupi fazi tačaka koje pripadaju (u nekoj meri) prepoznatoj fazi duži.

Detekcija topografskih simbola

Uvođenje savremene računarske tehnologije u proces izrade planova i karti neminovno uslovljava promenu klasičnog načina prikaza prostornih pojava. Karta je poseban oblik grafičkog dokumenta i u isto vreme sredstvo za skladištenje informacija i komunikaciju. Na kartama su objekti prikazani pomoću određenih znakova koji su unapred dogovoreni. Takav znak poseduje informaciju o određenom objektu, što znači da taj znak korisniku karte daje informaciju o kakvom se objektu radi.

Proces određivanja pozicije grupe objekata na osnovu jedne ili više topografskih karata (predstavljenih kao rasterske slike) veoma je kompleksan problem. U ovom procesu često je uključeno više operatera čiji zadatak je da na osnovu oblika topografskog znaka odrede njegovu poziciju znaka i njegove koordinate unesu u sistem za skladištenje prostornih podataka. Ovaj posao može delimično da se automatizuje u nekoliko segmenata:

- (i) kolaboracija više operatera,
- (ii) predprocesiranje rasterskih slika i transformacija u vektorski oblik,
- (iii) provera kvaliteta unesenih podataka,

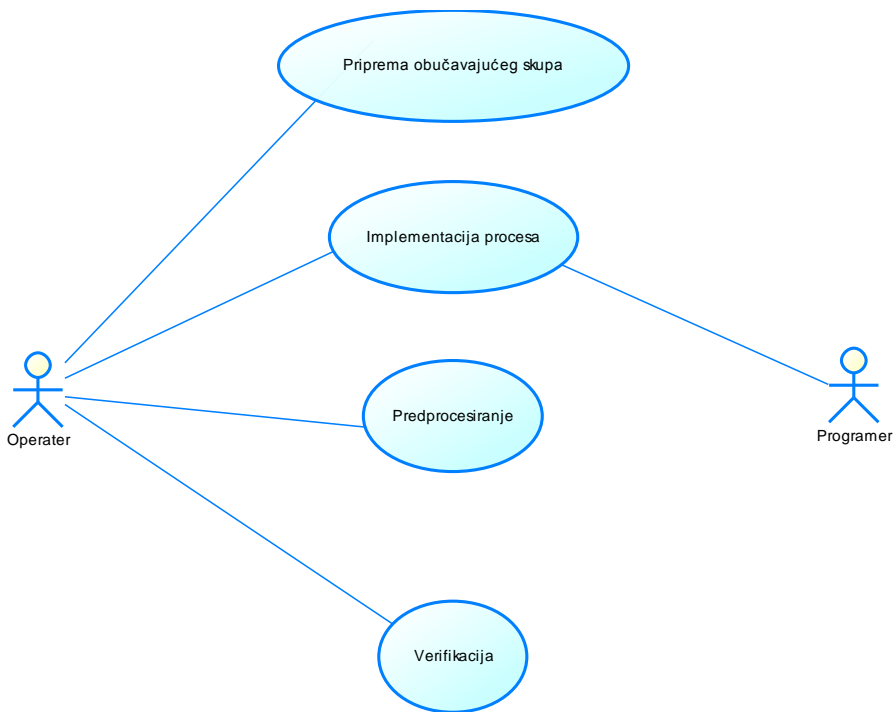
Automatizacija saradnje operatera prilikom unosa odnosi se na obezbeđenje okruženja u kojem više korisnika mogu da unose podatke istovremeno.

Predprocesiranje rasterskih slika i transformacija u vektorski oblik predstavlja posao u kojem se programski određuju mogući položaji predefinisanih topografskih znakova. Rasterski predstavljene topografske karte nastaju digitalizacijom karata gde su elementi ručno crtani (nisu svi znakovi identični). Čest je slučaj i da se prilikom skeniranja odnosno digitalizacije pokvari kvalitet podloge, odnosno oblik istog topografskog znaka na različitim mestima na karti postane delimično različit, ovaj problem nastaje zbog same prirode digitalizacije gde se kontinualni prostor reprezentuje diskretnom reprezentacijom.

Objekti u realanom svetu za čiju reprezentaciju koristimo znakove podležu određenim zakonitostima. Na primer, razmak između bandera je približno jednak a položaji bandera prate kretanje ulica. Ulična rasveta se takođe raspoređuje u odnosu na položaj ulica. Katastarske parcele se međusobno ne preklapaju i najčešće bar jednom delom dodiruju ulicu ili put. Ova pravila moguće je iskoristiti za merenje kvaliteta i utvrđivanje nekonzistentnosti unesenih podataka.

U ovom poglavlju prikazan je primer implementacije informacionog sistema u kojem su integrisani podsistemi opisani u poglavlju 3 i modeli podataka prikazani u poglavlju 2 sa ciljem predprocesiranja i određivanja položaja prepoznatih znakova na karti. U implementiranom primeru prikazana je mogućnost proširenja i dodavanja novih funkcija.

Učesnici u sistemu su *Operater* i *Programer*. *Operater* učestvuje u poslovima pripreme skupa znakova na osnovu kojih će se vršiti procesiranje rasterskih karata. U ovom poslu može da učestvuje i više operatera istovremeno. Nakon toga programer implementira proces u kojem koristi modele za reprezentaciju nepreciznih geoprostornih podataka zajedno sa tehnikama računarske inteligencije i obučavajućim skupom.



Slika 4.6: Dijagram slučajeva korišćenja

Ovako implementirani proces se zatim prebaci u *services* ili *task* katalog *FPostGIS* podsistema gde se pokrene predprocesiranje.

U toku predprocesiranja *operater* može da prati tok izvršavanja. Na kraju ili čak i u toku samog predprocesiranja *operater* vrši verifikaciju i eventualnu korekciju rezultata predprocesiranja. U svakom od ovih primera korišćenja učestvuje jedan ili više podsistema opisanih u poglavlju 3.

Slučaj korišćenja Priprema obučavajućeg skupa

- Opis:* Određuje se reprezentativan skup znakova koji će učestvovati u predprocesiranju.
- Okidač:* Ovaj slučaj korišćenja je prvi u nizu slučajeva i započinje pokretanjem aplikacije za formiranje obučavajućeg skupa.
- Završni uslovi:* Po završetku formirana je *xml* datoteka koja sadrži skup znakova koji će se koristiti u procesu obučavanja.
- Osnovni tok:* Korisnik pokrene proširenu aplikaciju AutoCAD delom u kojem je omogućeno formiranje obučavajućeg skupa i učitava georeferenciranu topografsku kartu. Potom, koristeći predefinisani geometrijski objekat za obeležavanje znakova obeleži pozicije znakova. Na kraju pozivanjem metode *sačuvaj obučavajući skup* formira se *xml* datoteka sa obučavajućim skupom.

Slučaj korišćenja Implementacija procesa

- Opis:* Implementacija procesa koji se koristi za predprocesiranje.
- Okidač:* *Operater* i *programer* po završetku pripreme pristupaju opisivanju procesa za prepoznavanje znakova.
- Završni uslovi:* Po završetku formirana je *jar* biblioteka u kojoj je data definicija procesa za predprocesiranje.
- Osnovni tok:* *Programer* pripremi osnovni opis procesa u obliku java programa, pripremi biblioteke klasa za podršku modeliranju neuronskih mreža i biblioteku za modeliranje nepreciznih geometrijskih objekata. Na kraju se cela biblioteka zapakuje u *jar* arhivu kao izvršni java program a kreira se i klasa predstavlja web servis za upravljanje izvršavanjem procesa.

Slučaj korišćenja **Predprocesiranje**

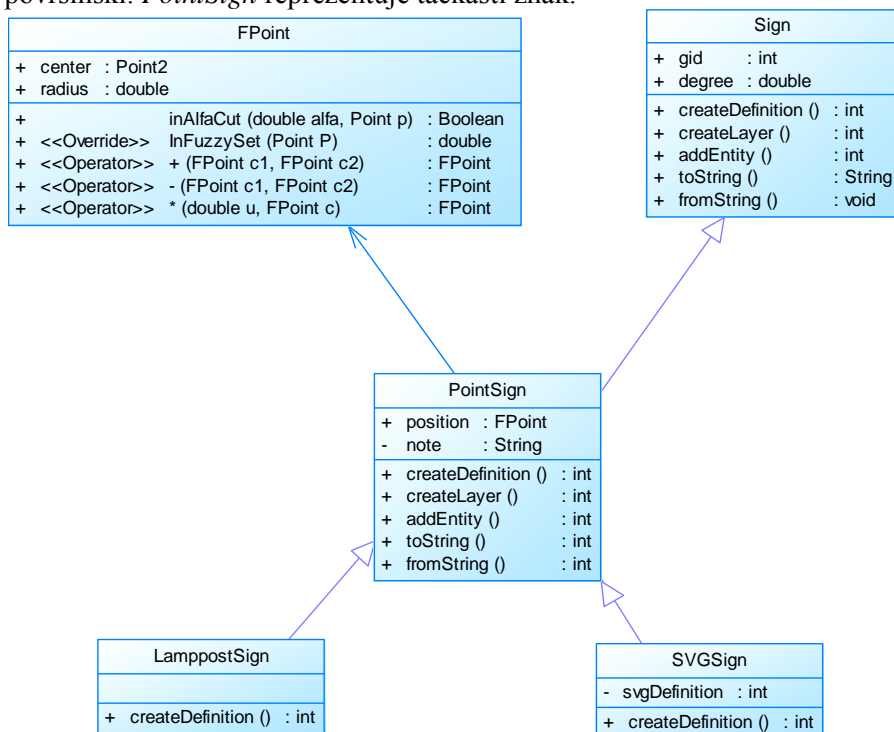
- Opis:* Izvršavanje procesa za prepoznavanje znakova na rasterskim topografskim kartama
- Okidač:* *Operater* pokreće ovaj slučaj korišćenja postavljanjem i pokretanjem procesa na aplikativnom serveru.
- Završni uslovi:* Po završetku preliminarni podaci se nalaze u PostGIS bazi.
- Osnovni tok:* *Operater* po pokretanju procesa korisnik dobija mogućnost da prati status izvršavanja kroz istoriju događaja koju emituje proces ili direktno preko preliminarnih podataka koji se generišu u bazi. Po završetku predprocesiranja u istoriji događaja na aplikativnom serveru zabeleži se tačno vreme završetka procesa kao i ukupno vreme i broj prepoznatih tačaka.

Slučaj korišćenja **Verifikacija**

- Opis:* *Operater* na kraju ciklusa obavezno mora da verifikuje prepoznate znakove.
- Okidač:* *Operater* pokreće ovaj slučaj kao posebnu aplikaciju implementiranu u proširenom AutoCAD okruženju.
- Završni uslovi:* Po završetku konačni podaci se nalaze u PostGIS bazi.
- Osnovni tok:* *Operater* pokreće specijalizovanu aplikaciju razvijenu kao proširenje AutoCAD okruženja i prateći predloge prihvata ili odbija preliminarne predloge prepoznatih znakova i pozicija.

Model podataka

Na slici 4.7 prikazan je dijagram klasa kojima su modelirani znakovi. Osnovna klasa *Sign* predstavlja apstraktnu klasu znaka sa proizvoljnom geometrijom. Razlikovaćemo tri vrste znakova tačkasti znakovi, linijski i površinski. *PointSign* reprezentuje tačkasti znak.



Slika 4.7: Dijagram klasa *sign* paketa

Implementacija ovih klasa izvršena je u C# i Java programskim jezicima. Razmena podataka između klasa različitih programskih jezika izvršena je pomoću dve metode *toString* i *fromString* koje vrše konverziju u odgovarajući *wkt* (*well known text*) format, čime je omogućena nesmetana razmena podataka između različitih platformi a u skladu je sa grupom OGC standarda. Predviđeno mesto razmene Java i C# objekata jeste prenos parametara prilikom poziva metoda web servisa.

U metodi *createDefinition* opisan je oblik objekta kojim se reprezentuje konkretan znak.

DTD *xml* datoteke kojom je opisan obučavajući skup prikazan je na sledećem listingu:

```
<!ELEMENT signs(sign*)>
<!ATTLIST sign url CDATA #REQUIRED>
<!ELEMENT sign>
<!ATTLIST      gid CDATA #REQUIRED
                x CDATA #REQUIRED
                y CDATA #REQUIRED
                width CDATA #REQUIRED
                height CDATA #REQUIRED>
```

Listing 4.1: DTD datoteke za opis obučavajućeg skupa.

Preliminarni podaci snimaju se u bazu podataka i čuvaju u tabeli sledećeg oblika:

```
CREATE TABLE "TopographicSign"
(
  gid serial NOT NULL,
  type int NOT NULL,
  note character varying(32),
  f_position fuzzyPoint
)
```

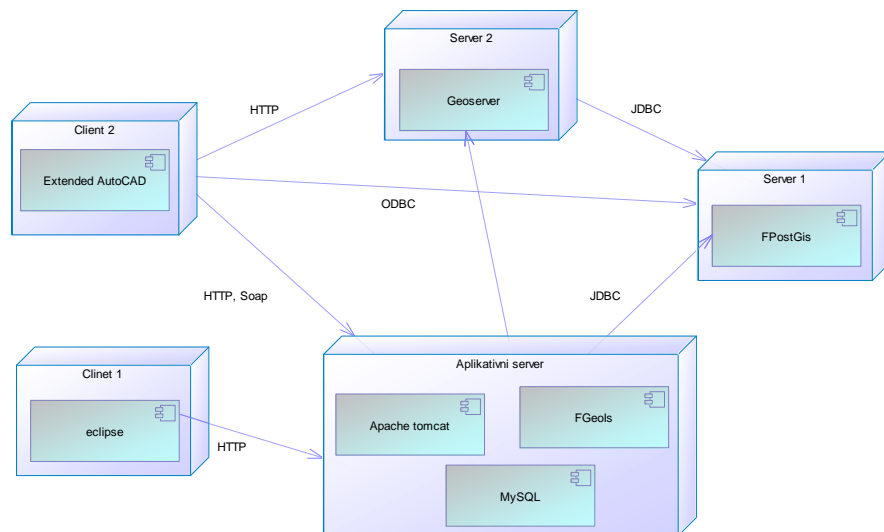
Listing 4.2: SQL izraz za kreiranje tabele topografskih znakova

Arhitektura implementiranog sistema

Na slici 4.8 prikazana je arhitektura implementiranog sistema. Predložena arhitektura je u skladu sa modelom prikazanim u prethodnom poglavlju.

Client 1 predstavlja računar koji se direktno koristi od strane *operatera* u slučajevima korišćenja priprema i verifikacija. *Client 2* predstavlja računar koji direktno koristi *programer*. Na njemu se razvija, testira i pakuje opis procesa u *jar* biblioteci, kreira klasa kojom je opisan web servis i šalje *FGeoIS serveru*. Client 1 odnosno proširena AutoCad aplikacija komunicira sa App serverom preko posebno razvijenog web servisa SOAP protokolom a prema *Geoserveru* HTTP protokolom.

Prošireni AutoCad ima mogućnost direktnog pristupa PostGIS bazi ili preko web servisa. Karte koje se preko URL adrese preuzimaju sa geoservera razmenjuju se korišćenjem HTTP protokola.



Slika 4.8: Dijagram razmeštaja komponenti implementiranog sistema

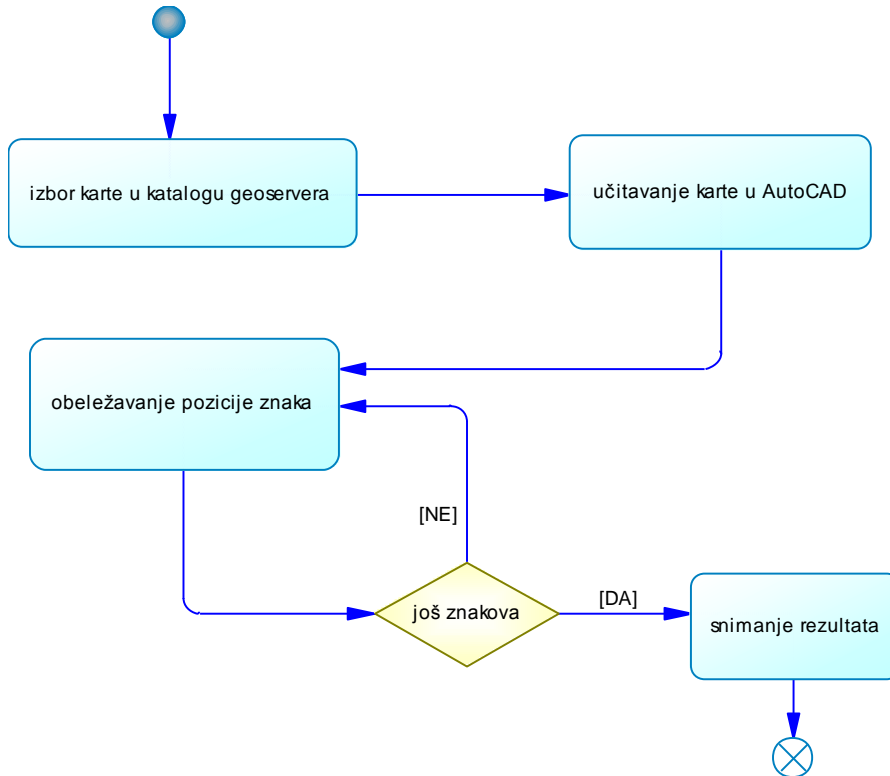
U slučaju korištenja *priprema* odnosno izbor obučavajućeg skupa učestvuje *operater* korišćenjem računara *Client2*.

Prvi korak je izbor karte koja je uskladištena na *geoserveru*. Izbor karte vrši se korišćenjem *WMSController* komponente opisane u poglavlju 3. Pored ove karte u AutoCAD je moguće učitati i ostale postojeće prostorne podatke koji mogu da pomognu prilikom pripreme obučavajućeg skupa.

Ponavljajući operaciju dodavanja pravougaonog objekta sa napomenom u kojoj se unosi vrsta znaka, obeležavaju se mesta na kojima se sigurno nalaze znakovi. Ovako predstavljeni elementi obučavajućeg skupa mogu da se privremeno sačuvaju u *dwg* datoteci jer posao ne mora da se završi bez prekidanja.

Po završetku *operater* pozivom komande *saveToXml* ili klikom na dugme "Save to xml" inicira kreiranje *xml* datoteke sa obučavajućim skupom u formatu opisanom u listingu 4.1.

Na slici 4.9 prikazan je dijagram aktivnosti u procesu kreiranja obučavajućeg skupa.

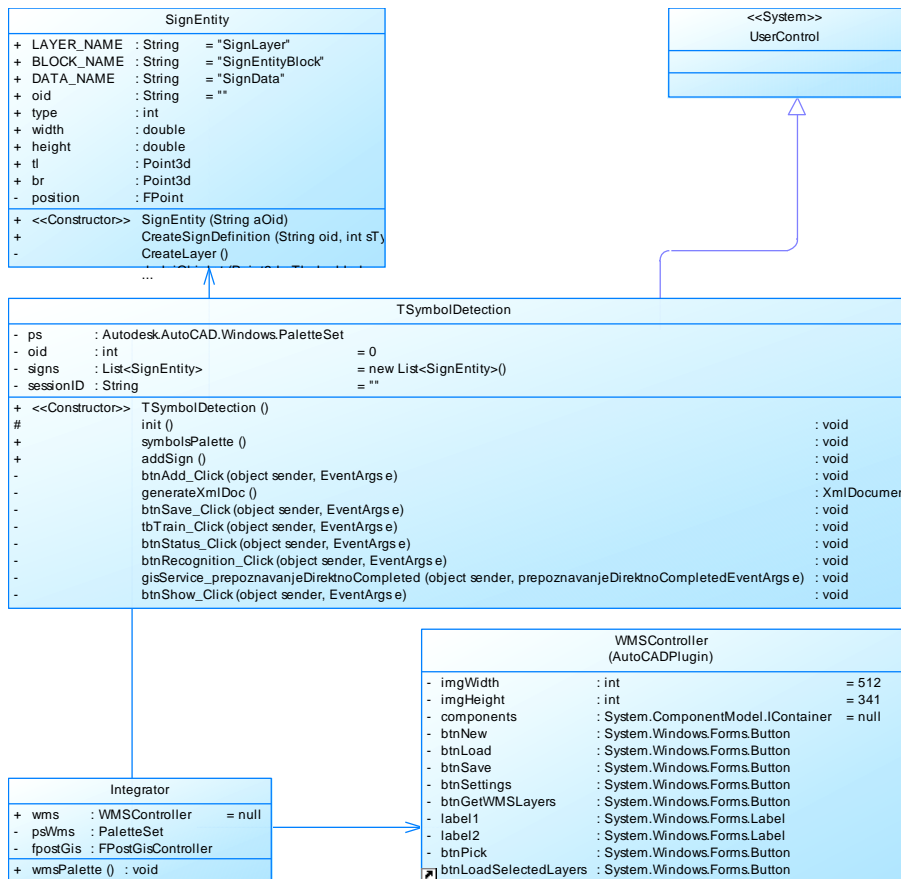


Slika 4.9: Dijagram aktivnosti slučaja korištenja *priprema*

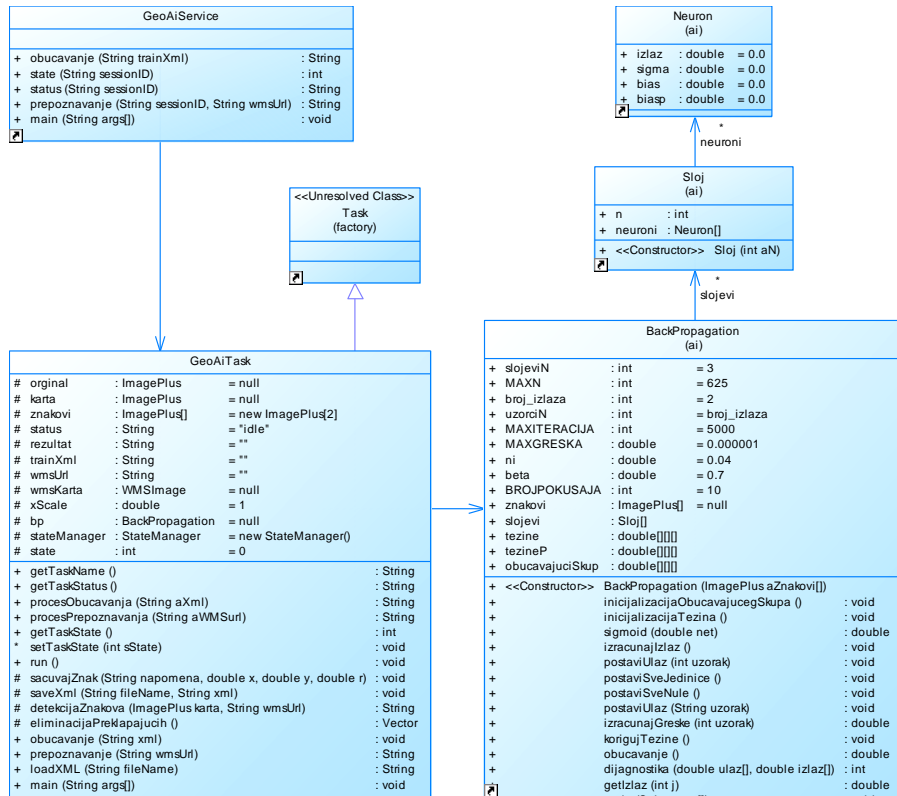
Na slici 4.9 prikazan je dijagram povezanosti klasa paketa *SymbolDetectionController*. Osnovna klasa je *TSymbolDetection* u kojoj je implementirana paleta alata za kreiranje obučavajućeg skupa, pokretanje procesa na serveru i preuzimanje rezultata. Klasom *SignEntity* opisan je AutoCad geometrijski objekat za reprezentaciju jednog znaka. Prilikom dodavanja objekti se smestaju na poseban AutoCad *layer* a karakterišu ih pored oblika (širina i visina) i pozicije (fuzzy tačka) i ostali atributi poput identifikatora (*oid*) i tip znaka (*type*). Ovako definisane attribute moguće je menjati preko ugrađenih formi za izmenu AutoCad objekata.

U slučaju da želimo da umesto pravougaonika određene boje znak predstavimo složenim izgledom definiciju znaka (njegov oblik) ugrađićemo u klasi koja nasleđuje *SignEntity*. Moguće proširenje klase jeste da se oblik znaka generiše na osnovu *SVG* (Scalable Vector Graphics) reprezentacije.

Klasa *TSymbolDetection* (Slika 4.10) povezana je sa *WMSController* klasom preko statičke klase *Integrator*.

Slika 4.10: Dijagram paketa *SymbolDetectionController*

Slučaj korišćenja *predprocesiranje* implementiran je kao java proces koji se izvršava na *FPostGIS* serveru. Na slici 4.11 prikazan je dijagram povezanosti klasa kojima je realizovan java proces za detekciju znakova.



Slika 4.11: Dijagram povezanosti klasa procesa za prepoznavanje znakova

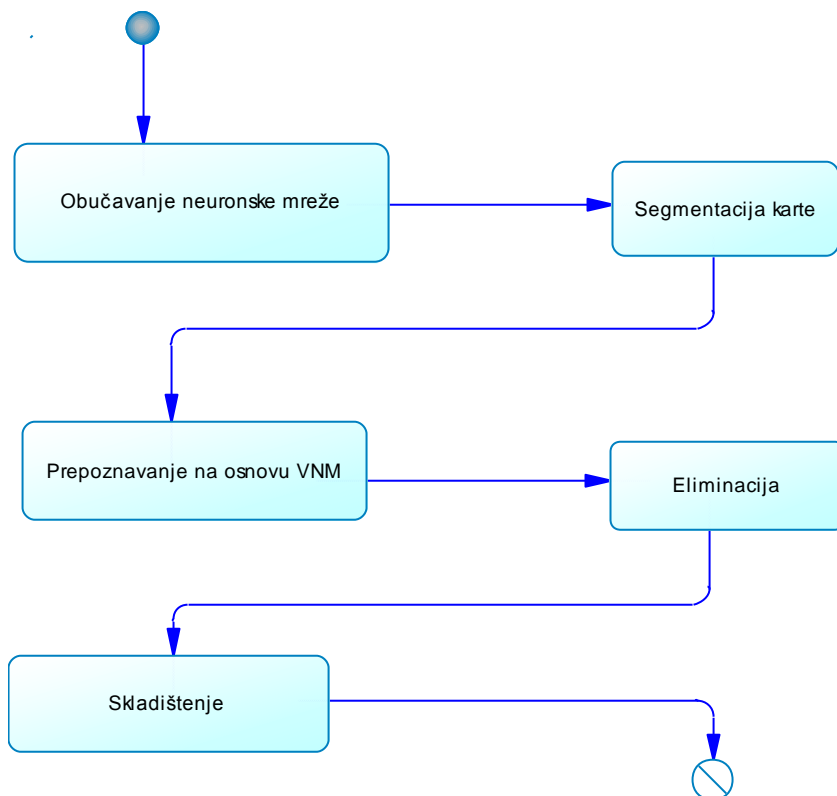
U klasa *GeoAiTask* implementiran je algoritam za prepoznavanje znakova. Klasa koja služi za upravljanje izvršavanjem procesa *GeoAiTask* je klasa *GeoAiService*. Ova klasa implementirana je tako da kad se prebaci u *service* katalog *FGeoIS* sistema radi kao web servis. Veštačka neuronska mreža kojom je realizovana detekcija znaka implementirana je preko klase *BackPropagation*. Ova klasa sadrži više slojelava modeliranih klasom *Sloj* a ona sarži niz *Neuron* klasa.

Na slici 4.12 prikazan je dijagram aktivnosti za detekciju znakova na karti. Ulaz je *xml* i URL karte na kojoj se prepoznaju znakovi, a rezultat je skup fazi tačaka kojim se reprezentuju objekti sačuvan u PostGIS sistemu za upravljanje prostornim podacima.

Prvi korak je preuzimanje *xml* datoteke kojom je opisan obučavajući skup iz slučaja korišćenja *priprema*. Sledeći korak je kreiranje neuronske mreže sa odgovarajućim brojem ulaznih neurona (broj ulaznih neurona odgovara broju piksela obuhvaćenih pravougaonikom obeleženog znaka). Nakon

toga, pripremi se obučavajući skup i izvrši obučavanje. U ovom primeru veličina pravougaonika je 25×25 piksela. Neuronska mreže je troslojna sa 625 ulaznih neurona, 20 neurona u skrivenom sloju i dva neurona u izlaznom sloju (broj je određen brojem različitih znakova u obučavajućem skupu)

Segmentacija se izvodi tako što se preskaču delovi slike u kojima se sigurno ne nalazi potencijalni znak. Ovo je pogodno u primeru topografskih karata jer ima dosta belih segmenata koji se ignorišu.



Slika 4.12: Dijagram aktivnosti procesa prepoznavanja znakova

Prepoznavanje ili analiza se izvodi pomoću prozora 25×25 piksela koji se sekvencijalno pomera duž x i y koordinata slike. Sadržaj prozora dovodi se kao ulaz u neuronsku mrežu. Signal se propusti kroz mrežu i ako se na izlazu pojavi neuron u pobuđenom stanju trenutna pozicija se memoriše kao pozicija na kojoj se nalazi znak pridružen pobuđenom neuronu.

U procesu prepoznavanja svakoj tački karte dodeli se stepen sličnosti sa prepoznatim znakom. Ovaj stepen sličnosti je realna broj u intervalu od nula do jedan a dobija se kao odgovarajući izlaz iz neuronske mreže. Kao finalni rezultat uzima aprksimacija ovako dobijenih rezultata fazi tačkom čiji centar odgovara lokalnom maksimumu tačaka grupisanih u okolini jednog znaka a poluprečnik ove grupe odgovara poluprečniku fazi tačke.

Skladištenje je moguće na dva načina direktno JDBC protokolom ili pozivom metode web servisa za skladištenje. Web servis za skladištenje implementiran je kao java klasa koja se pošalje na *FGeoIS* server koji njezine metode izloži kao metode web servisa. Primer klase za skladištenje dat je u listingu:

```
public class SimpleWebServis {

    public SimpleWebServis() {
    }
    public int sacuvajZnak(String napomena,
        double x, double y, double r){
        java.sql.Connection conn;
        try {
            Class.forName("org.postgresql.Driver");
            String url= "jdbc:postgresql://localhost:5432/postgis";
            conn = DriverManager.getConnection(url, "postgres",
                "root");

            String update = "INSERT INTO " +
                "public.\"FuzzyPointDemo\"(note,f_position)";
            update += " VALUES ( ";
            update += "? ,fls_fuzzyPoint(asewkt('POINT("+x+
                " "+y+"')', "+r+"))";
            update += ")";
            PreparedStatement pstmt= conn.prepareStatement(update);
            pstmt.setString(1, napomena);
            pstmt.executeUpdate();
            pstmt.close();
            conn.close();
            return 1;
        }
        catch ( Exception e ) {
            e.printStackTrace();
        }
        return 0;
    }
}
```

Listing 4.3: Primer web servisa za skladištenje

Deo koda kojim se poziv web servisa prilikom čuvanja podataka dat je na listingu:

```
String endpoint =
    "http://localhost/geois/files/java/SimpleWebServis.jws";
```

```

String i1 = new String("test jws");
Double i2 = new Double("12.3");
Double i3 = new Double("32.1");
Double i4 = new Double("0.1");

Service service = new Service();
Call call = (Call) service.createCall();

call.setTargetEndpointAddress( new java.net.URL(endpoint) );
call.setOperationName( "sacuvajZnak" );
call.addParameter( "op1", XMLType.XSD_STRING, ParameterMode.IN );
call.addParameter( "op2", XMLType.XSD_DOUBLE, ParameterMode.IN );
call.addParameter( "op3", XMLType.XSD_DOUBLE, ParameterMode.IN );
call.addParameter( "op4", XMLType.XSD_DOUBLE, ParameterMode.IN );
call.setReturnType( XMLType.XSD_INT );

Integer ret=(Integer)call.invoke(new Object [] { i1, i2 , i3, i4});

```

Listing 4.4: Primer poziva metode web servisa

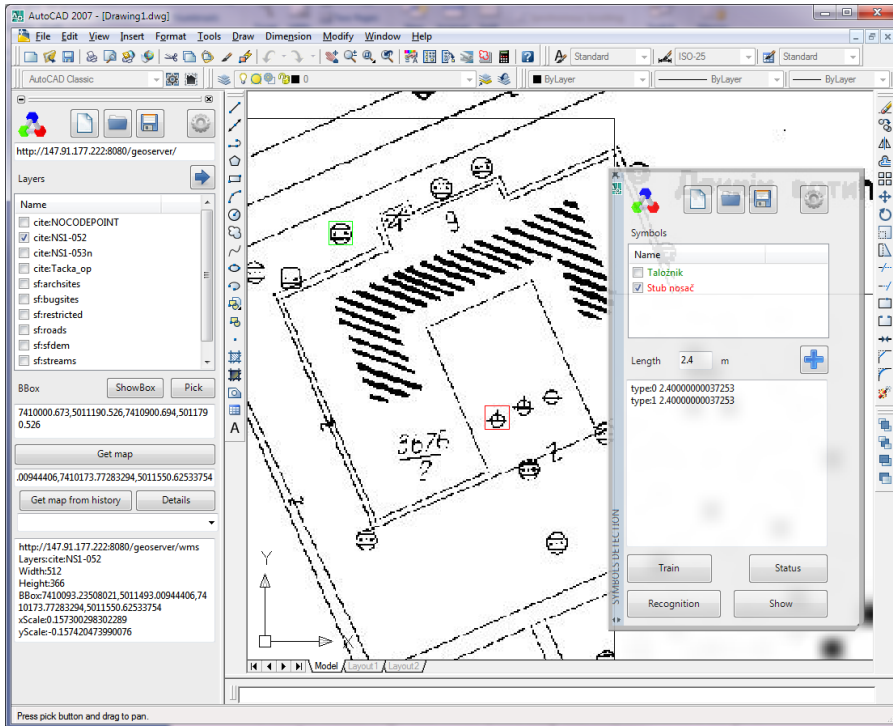
Rezultat

U ovom primeru korištene su karte smeštene u katalogu *Geoservera*. Pristup kartama je omogućen WMS standardom. U tabeli 4.1 prikazani su parametri za generisanje WMS zahteva u obliku URL adrese.

Naziv parametra	Vrednost
SERVICE	WMS
VERSION	1.1.1
REQUEST	GetMap
LAYERS	cite:NS1-052
HEIGHT	366
WIDTH	512
SRS	EPSG%3A4326
FORMAT	image%2Fjpeg
EXCEPTIONS	application%2Fvnd.ogc.se_inimage 7410093.23508021,5011493.00944406, 7410173.77283294,5011550.62533754
BBOX	7410173.77283294,5011550.62533754

Tabela 4.1 Pregled parametara za generisanje WMS zahteva

Na slici 4.13 prikazan je izgled ekranske forme AutoCad okruženja u kojem su otvorene palete alata *WMSController* (za kreiranje wms zahteva i prikaz mapa) i *TSymbolDetection* (za pripremu obučavajućeg skupa i komunikaciju sa web servisom za detekciju).



Slika 4.13: AutoCad sa otvorenim paletama alata za detekciju

Datoteka *signs.xml* generiše se klikom na dugme SaveToXml (slika diskete) palete *Symbols Detection*. Sadržaj datoteke dat je u nastavku:

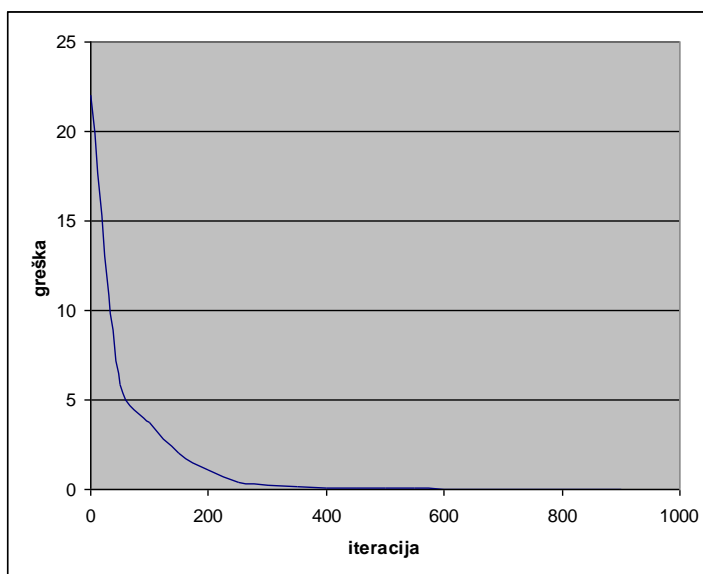
```
<signs
  url=http://147.91.177.222:8080/geoserver/
  layerName="cite:NS1-052">
  <sign gid="1"
    x="7410144.66621408"
    y="5011540.11969618"
    type="0" width="2.4" />
  <sign gid="1"
    x="7410160.61183161"
    y="5011521.39043132"
    type="1" width="2.4" />
</signs>
```

Neuronska mreža korišćena u ovom primeru ima 625 neurona u ulaznom sloju, 10 u skrivenom i 20 neurona u izlaznom sloju. Parametri za obučavanje back-propagation algoritmom dati su u tabeli 4.2

Naziv parametra	Vrednost
Maksimalan broj iteracija	5000
Maksimalna greška	0.0001
Brzina obučavanja	0.04
Uticaj prethodnih promena	0.7

Tabela 4.2: Parametri za obučavanje neuronske mreže

Na slici 4.14 prikazani su rezultati obučavanja neuronske mreže. Nakon 619 iteracija greška je iznosila 0.000099836543595238



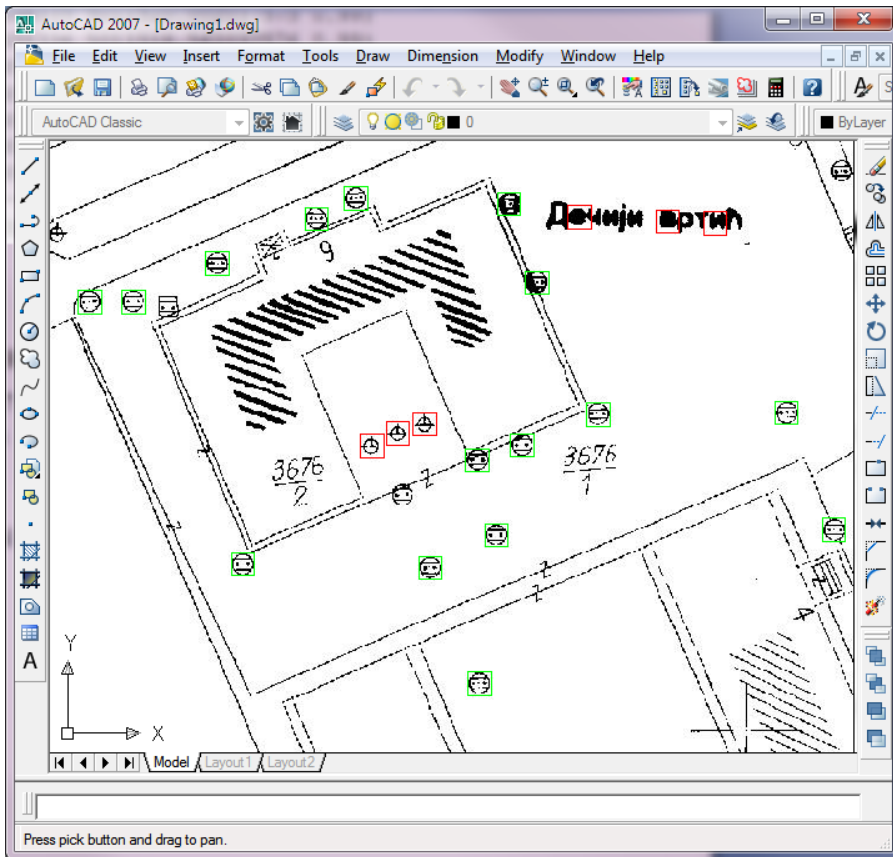
Slika 4.14: Rezultati obučavanja neuronske mreže

Nakon obučavanja pokrenut je proces detekcije. U tabeli 4.3 prikazani su rezultati detekcije.

Znak	Pozicija
0	FPOINT (7410160.241496335 5011545.634037579 0.99)
0	FPOINT (7410176.0814963365 5011545.05803758 0.97)
1	FPOINT (7410183.377496338 5011543.714037579 0.98)
0	FPOINT (7410156.113496333 5011543.522037580 0.97)
1	FPOINT (7410192.497496339 5011543.234037570 0.99)
1	FPOINT (7410197.393496340 5011543.042037579 0.92)
0	FPOINT (7410145.841496332 5011538.914037579 0.99)
0	FPOINT (7410178.961496337 5011536.898037578 0.99)
0	FPOINT (7410132.689496330 5011534.882037578 0.99)

0	FPOINT (7410137.201496331 5011534.882037578 0.99)
0	FPOINT (7410204.785496341 5011523.458037577 0.99)
0	FPOINT (7410185.297496338 5011523.266037576 0.98)
1	FPOINT (7410167.345496335 5011522.306037576 0.99)
1	FPOINT (7410164.561496335 5011521.346037576 0.99)
0	FPOINT (7410177.425496337 5011520.098037575 0.99)
1	FPOINT (7410161.873496335 5011520.002037575 0.99)
0	FPOINT (7410172.721496336 5011518.562037576 0.99)
0	FPOINT (7410209.681496342 5011511.362037575 0.99)
0	FPOINT (7410174.737496337 5011510.786037574 0.99)
0	FPOINT (7410148.529496333 5011507.810037574 0.98)
0	FPOINT (7410167.921496335 5011507.426037574 0.99)
0	FPOINT (7410173.009496337 5011495.426037572 0.96)

Tabela 4.3: Rezultat detekcije



Slika 4.14: Rezultat detekcije prikazan u AutoCad okruženju

U posmatranom regionu nalazi se ukupno 23 simbola, od toga 19 *taložnika* \ominus i 4 znaka koji predstavljaju *StubPosredniNosacElLampe* (kraće *stub*) \oplus . U procesu detekcije prepoznato je 16 *taložnika* i 3 simbola za *stub*.

Pogrešno su prepoznata tri simbola za *stub* i nisu prepoznata tri *taložnika* i jedan *stub*.

U procesu verifikacije korisnik korišćenjem AutoCad okruženja može da izvrši korekciju i rezultate sačuva u *FPostGis* bazi.

Poglavlje 5

Zaključak

Predmet istraživanja disertacije pripada oblasti primene računarske inteligencije u modeliranju geoprostornih simulacionih sistema.

Početak istraživanja na temu modelovanja neodređenih i nepreciznih geoprostornih podataka postavio je Carl Friedrich Gauss početkom 19. veka u knjizi *Teorija kretanja nebeskih tela* u kojoj je opisao način određivanja položaja nebeskih tela na osnovu približnog merenja rastojanja.

Posebno značajan doprinos modernim pristupima za modeliranje nepreciznosti i neodređenosti postavio je Lotfi A. Zadeh 1965. u radu *Fuzzy Sets* gde je po prvi put formalno definisao delimičnu pripadnost skupu a time i omogućio bogatije predstavljanje modela.

Intenzivnim razvojem računarske infrastrukture i globalne računarske mreže problem modeliranja i korišćenja geoprostornih podataka dobio je još jednu dimenziju a to je mogućnost primene u uslovima postojanja velike količine geoprostornih podataka, koji se prikupljaju sa različitih izvora, obrađuju i potom postaju dostupne velikom broju korisnika.

Model proširivog simulacionog sistema koji omogućava modeliranje nepreciznih geoprostornih podataka, njihovo prikupljanje, obradu, simulaciju procesa i kasnije distribuciju rezultata do krajnjih korisnika osnovna je tema ove disertacije. Osnovni aspekti proširivosti modela koji je predložen u ovoj disertaciji su: (i) mogućnost proširenja samog modela podataka (ii) mogućnost razvoja i primene algoritama za analizu i obradu nepreciznih geoprostornih podataka i (iii) mogućnost primene različitih tehnika računarske inteligencije za modeliranje procesa.

U prvom poglavlju analizirana su relevantna istraživanja kroz dostupne literaturne izvore i izvršena je sistematizacija ovih istraživanja u kategorije koje obuhvataju karakteristične pristupe modeliranju geoprostora u uslovima neodređenosti. Rezultati analize pokazuju da je problem modeliranja geoprostornih podataka u uslovima neodređenosti predmet intenzivnog istraživanja u tri kategorije: (i) reprezentacija *približnim ali preciznim* modelima (ii) probabilistički pristup i (iii) primenom teorije fazi skupova odnosno fazi pristupom. Kao osnova za razvoj modela u disertaciji izabran je pristup u kojem se koristi teorija fazi skupova i fazi logike zbog velikog broja formalnih metoda za njihovu analizu i obradu.

Drugo i treće poglavlje sadrže osnovni naučni doprinos disertacije. Ova poglavlja obuhvataju:

- (i) Formalnu definiciju modela nepreciznih geoprostornih podataka baziranih na teoriji fazi skupova, kao i matematičke dokaze osobina takvih modela.
- (ii) Definicije osnovnih prostornih operacija, funkcija i relacija između nepreciznih geoprostornih objekata.
- (iii) Formalnu specifikaciju arhitekture proširivog simulacionog sistema.

Prilikom modeliranja nepreciznih geoprostornih podataka i procesa vođeno je računa o činjenici da se u praktičnoj primeni obrada i zaključivanje po pravilu vrši na velikoj količini podataka. Zato je izabran model koji predstavlja kompromis između potrebe za detaljnim modelom i velike količine nepreciznih geoprostornih podataka.

Za specifikaciju arhitekture proširivog simulacionog sistema korišćen je objektni pristup i jezik UML. Softverska arhitektura bazirana je na Internet tehnologijama i važećim standardima.

Treće poglavlje prikazuje softversku arhitekturu i implementaciju prototipa simulacionog softverskog sistema koji se sastoji od više komponenti grupisanih u tri kategorije:

- (i) komponente koje čine prezentacioni deo sistema
- (ii) elementi za podršku modeliranju procesa, simulaciju i izvršavanje
- (iii) podsistem za skladištenje nepreciznih geoprostornih podataka

Doprinos ovog dela disertacije je softverska arhitektura koja integriše više komponenti baziranih na proverenim softverskim proizvodima realizovanim u različitim tehnologijama a pritom dozvoljava proširenje novim modelima, metodama i tehnikama za prikupljanje, analizu, obradu i skladištenje velike količine nepreciznih geoprostornih podataka. Predstavljena prototipska implementacija ispunjava preporuke Infrastrukture prostornih podataka o korišćenju standarda u razmeni geoprostornih podataka.

Četvrto poglavlje predstavlja praktičnu verifikaciju predloženih modela podataka, procesa i arhitekture sistema na jednom konkretnom primeru. U ovom primeru, pored već pomenute teorije fazi skupova, korišćene su i tehnike bazirane na veštačkim neuronskim mrežama i wavelet transformacije u procesu prikupljanja nepreciznih geoprostornih podataka.

Pravci daljih istraživanja koji proističu iz ove teze grupišu se u sledeće grupe.

Prva grupa obuhvata istraživanja vezana za razvoj kompleksnijih modela za reprezentaciju prostornih podataka i procesa baziranih na tipu 2 fazi skupova i drugih tehnika računarske inteligencije.

Druga grupa obuhvata istraživanja mogućnosti priemene predloženih modela u različitim oblastima u kojima postoji poteba analize prostorno i vremenski-prostorno nepreciznih podataka kao što su, recimo, medicina, poljoprivreda i slično.

Treća grupa obuhvata istraživanja vezana za standardizaciju u oblasti razmene nepreciznih geoprostornih podataka.

Literatura

- [1] R. H. Güting, D. Papadias, and F. Lochovsky, Eds., *Advances in Spatial Databases*, vol. 1651. Berlin, Heidelberg: Springer Berlin Heidelberg, 1999.
- [2] “The OpenGIS® Abstract Specification Topic 5: Features,” *The OpenGIS® Abstract Specification Topic 5: Features*. [Online]. Available: http://portal.opengeospatial.org/modules/admin/license_agreement.php?suppressHeaders=0&access_license_id=3&target=http://portal.opengeospatial.org/files/?artifact_id=29536. [Accessed: 01-Dec-2010].
- [3] S. Shekhar, *Encyclopedia of GIS*, 1st ed. Springer, 2007.
- [4] M. Ashley and P. Frederick, “UGML: an extension of GML to provide support for geographic objects with uncertain boundaries,” in *7th International Symposium on Spatial Accuracy Assessment in Natural Resources and Environmental Sciences.*, Lisboa, 2006.
- [5] “Uncertainty Markup Language (UnCertML),” *Uncertainty Markup Language (UnCertML)*, 8-2009. [Online]. Available: http://portal.opengeospatial.org/files/?artifact_id=33234. [Accessed: 12-Nov-2010].
- [6] R. Aris, *Mathematical Modelling Techniques*, Rei Una. Dover Publications, 1995.
- [7] N. Gershenfeld, *The Nature of Mathematical Modeling*. Cambridge University Press, 1998.
- [8] C. C. Lin and L. A. Segel, *Mathematics Applied to Deterministic Problems in the Natural Sciences*. SIAM: Society for Industrial and Applied Mathematics, 1988.
- [9] X.-S. Yang, *Mathematical Modelling for Earth Sciences*. Dunedin Academic Press Ltd., 2008.
- [10] P. Diamond and P. Kloeden, “Metric spaces of fuzzy sets,” *Fuzzy Sets and Systems*, vol. 35, no. 2, pp. 241-249, Apr. 1990.
- [11] B. B. Chaudhur and A. Rosenfeld, “On a metric distance between fuzzy sets,” *Pattern Recognition Letters*, vol. 17, no. 11, pp. 1157-1160, Sep. 1996.

- [12] I. Kramosil and J. Michálek, "Fuzzy metric and statistical metric spaces," *Kybernetika*, vol. 11, no. 5, p. 326–334, 1975.
- [13] S. Sharma, "On Fuzzy Metric Space," *Southeast Asian Bulletin of Mathematics*, vol. 26, no. 1, pp. 133-145, 2003.
- [14] Y. Yue and F.-G. Shi, "On fuzzy pseudo-metric spaces," *Fuzzy Sets and Systems*, vol. 161, no. 8, pp. 1105-1116, Apr. 2010.
- [15] O. Hadžić and E. Pap, *Fixed point theory in probabilistic metric spaces*. Dordrecht: Kluwer Academic Publishers, 2001.
- [16] O. Kaleva, "A comment on the completion of fuzzy metric spaces," *Fuzzy Sets and Systems*, vol. 159, no. 16, pp. 2190-2192, Aug. 2008.
- [17] O. Kaleva and S. Seikkala, "On fuzzy metric spaces," *Fuzzy Sets and Systems*, vol. 12, no. 3, pp. 215-229, Apr. 1984.
- [18] L. Behounek, U. Bodenhofer, and P. Cintula, "Relations in Fuzzy Class Theory:: Initial steps," *Fuzzy Sets and Systems*, vol. 159, no. 14, pp. 1729-1772, Jul. 2008.
- [19] M. Schneider, "Metric operations on fuzzy spatial objects in databases," in *Proceedings of the 8th ACM international symposium on Advances in geographic information systems*, 2000, p. 21–26.
- [20] V. Gregori, S. Morillas, and A. Sapena, "On a class of completable fuzzy metric spaces," *Fuzzy Sets and Systems*, vol. 161, no. 16, pp. 2193-2205, Aug. 2010.
- [21] C.-H. Cheng, "A new approach for ranking fuzzy numbers by distance method," *Fuzzy Sets and Systems*, vol. 95, no. 3, pp. 307-317, May. 1998.
- [22] S. Heilpern, "Comparison of fuzzy numbers in decision making," *Tatra Mt. Math. Publ.*, vol. 6, p. 47–53, 1995.
- [23] U. Höhle and N. Blanchard, "Partial ordering in L-underdeterminate sets," *Information Sciences*, vol. 35, no. 2, pp. 133-144, May. 1985.
- [24] S. Lee, K. H. Lee, and D. Lee, "Ranking the sequences of fuzzy values," *Information Sciences—Informatics and Computer Science: An International Journal*, vol. 160, p. 41–52, Mar. 2004.
- [25] E. P. Klement, R. Mesiar, and E. Pap, "A characterization of the ordering of continuous t-norms," *Fuzzy Sets and Systems*, vol. 86, no. 2, pp. 189-195, Mar. 1997.

-
- [26] M. Demirci, "A theory of vague lattices based on many-valued equivalence relations--I: general representation results," *Fuzzy Sets and Systems*, vol. 151, no. 3, pp. 437-472, May. 2005.
- [27] A. Markov, "Proširenje zakona o velikim brojevima na međusobno zavisne količine," presented at the Зборник радова Физичко-математичко друштво Казана Универзитета., 1906, vol. 2, pp. 135-156.
- [28] A. M. Turing, "On Computable Numbers, with an Application to the Entscheidungsproblem," *Proceedings of the London Mathematical Society*, vol. s2-42, no. 1, pp. 230-265, 1937.
- [29] J. V. Neumann and A. H. Taub, *The Collected Works of John von Neumann : 6-Volume Set*. Reader's Digest Young Families, 1963.
- [30] J. Von Neumann and A. W. Burks, "Theory of self-reproducing automata," *IEEE Transactions on Neural Networks*, vol. 5, no. 1, p. 3-14, 1994.
- [31] X. Lu, "An investigation on service-oriented architecture for constructing distributed web gis application," in *Services Computing, 2005 IEEE International Conference on*, 2005, vol. 1, p. 191-197.
- [32] G. Alonso and C. Hagen, "Geo-Opera: Workflow Concepts for Spatial Processes," *IN PROC. 5TH INTL. SYMPOSIUM ON SPATIAL DATABASES (SSD '97)*, p. 238--258, 1997.
- [33] L. Bernard, U. Einspanier, M. Lutz, and C. Portele, "Interoperability in gi service chains-the way forward," in *AGILE 2003: 6th AGILE conference on geographic information science*, 2003, p. 179.
- [34] D. H. Kim and M. S. Kim, "Web GIS service component based on open environment," in *Geoscience and Remote Sensing Symposium, 2002. IGARSS'02. 2002 IEEE International*, 2002, vol. 6, p. 3346-3348.
- [35] M. Neun and D. Burghardt, "Web services for an open generalisation research platform," in *8th ICA Workshop on Generalisation and Multiple Representation*, 2005.
- [36] Maarten Löffler, "Data Imprecision in Computational Geometry," Utrecht University, 2009.
- [37] L. Guibas, D. Salesin, and J. Stolfi, "Epsilon geometry: building robust algorithms from imprecise computations," in *Proceedings of*

the fifth annual symposium on Computational geometry, New York, NY, USA, 1989, p. 208–217.

- [38] F. Hassanzadeh and D. Rappaport, “Approximation Algorithms for Finding a Minimum Perimeter Polygon Intersecting a Set of Line Segments,” in *Proceedings of the 11th International Symposium on Algorithms and Data Structures*, Berlin, Heidelberg, 2009, p. 363–374.
- [39] M. Löffler and M. Kreveld, “Largest and Smallest Convex Hulls for Imprecise Points,” *Algorithmica*, vol. 56, no. 2, pp. 235-269, 2008.
- [40] M. Löffler and J. Snoeyink, “Delaunay triangulation of imprecise points in linear time after preprocessing,” *Computational Geometry*, vol. 43, no. 3, pp. 234-242, Apr. 2010.
- [41] M. Burl, M. Weber, and P. Perona, “A probabilistic approach to object recognition using local photometry and global geometry,” *Lecture Notes in Computer Science*, vol. 1407, pp. 628-641, 1998.
- [42] C. K. Cheung, W. Shi, and X. Zhou, “A Probability-based Uncertainty Model for Point-in-Polygon Analysis in GIS,” *GeoInformatica*, vol. 8, no. 1, pp. 71-98, 2004.
- [43] M. Schneider, “Design and implementation of finite resolution crisp and fuzzy spatial objects,” *Data & Knowledge Engineering*, vol. 44, p. 81–108, Jan. 2003.
- [44] M. Schneider, *Spatial Data Types for Database Systems: Finite Resolution Geometry for Geographic Information Systems*, 1st ed. Springer, 1997.
- [45] J. J. Buckley and E. Eslami, “Fuzzy plane geometry I: Points and lines,” *Fuzzy Sets and Systems*, vol. 86, no. 2, pp. 179-187, Mar. 1997.
- [46] J. Gasós and A. Rosetti, “Uncertainty representation for mobile robots: perception, modeling and navigation in unknown environments,” *Fuzzy Sets and Systems*, vol. 107, p. 1–24, Oct. 1999.
- [47] K. F. Gauss, *Theory of Motion of the Heavenly Bodies Moving About the Sun in Conic Sections: A Translation of Theoria Motus*, Illustrated edition. Dover Publications, 2004.
- [48] L. A. Zadeh, “Fuzzy sets,” *Information and Control*, vol. 8, no. 3, pp. 338-353, Jun. 1965.

-
- [49] D. Dubois and H. Prade, *Fuzzy Sets and Systems : Theory and Applications (Mathematics in Science and Engineering)*. New York: Academic Press, 1980.
- [50] A. Rosenfeld, "The fuzzy geometry of image subsets," *Pattern Recognition Letters*, vol. 2, no. 5, pp. 311-317, Sep. 1984.
- [51] A. Rosenfeld, "Distances between fuzzy sets," *Pattern Recognition Letters*, vol. 3, no. 4, pp. 229-233, Jul. 1985.
- [52] D. Dubois and M.-C. Jaulent, "A general approach to parameter evaluation in fuzzy digital pictures," *Pattern Recognition Letters*, vol. 6, no. 4, pp. 251-259, Sep. 1987.
- [53] R. K. Bose and D. Sahani, "Fuzzy mappings and fixed point theorems," *Fuzzy Sets and Systems*, vol. 21, no. 1, pp. 53-58, Jan. 1987.
- [54] S. Sharma, "Common fixed point theorems in fuzzy metric spaces," *Fuzzy Sets and Systems*, vol. 127, no. 3, pp. 345-352, May. 2002.
- [55] U. Bodenhofer, "A similarity-based generalization of fuzzy orderings preserving the classical axioms," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 8, p. 593-610, Oct. 2000.
- [56] H. Lee-Kwang and J. H. Lee, "A method for ranking fuzzy numbers and its application to decision-making," *Fuzzy Systems, IEEE Transactions on*, vol. 7, no. 6, p. 677-685, 2002.
- [57] H. B. Mitchell and P. A. Schaefer, "On ordering fuzzy numbers," *International Journal of Intelligent Systems*, vol. 15, no. 11, pp. 981-993, 2000.
- [58] E. P. Klement, R. Mesiar, and E. Pap, *Triangular norms*. Kluwer Academic Publishers, 2000.
- [59] Ulrich Bodenhofer, "Orderings of Fuzzy Sets Based on Fuzzy Orderings Part I: The Basic Approach," *Mathware & Soft Computing*, vol. 15, pp. 201-218, 2008.
- [60] Đ. Obradović, Z. Konjovic, E. Pap, and N. M. Ralević, "The maximal distance between imprecise point objects," *Fuzzy Sets and Systems*, vol. 170, no. 1, pp. 76-94, May. 2011.
- [61] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," *Information Processing Letters*, vol. 1, no. 4, pp. 132-133, Jun. 1972.

- [62] M. Segedinac, Đ. Obradović, and Z. Konjović, "Proširenje interpretera XProlog fazi unifikacijom," in *YuInfo 2009*, Kopaonik, 2009.
- [63] Đ. Obradović, Z. Konjović, and E. Pap, "Extending PostGIS by imprecise point objects," in *IEEE 8th International Symposium on Intelligent Systems and Informatics*, Subotica, Serbia, 2010, pp. 23-28.
- [64] Đ. Obradović and M. Segedinac, "Softverska arhitektura za transformaciju geoprostornih podataka," in *YuInfo 2009*, Kopaonik, 2009.
- [65] Đ. Obradović, D. Jovanović, and M. Govedarica, "Web bazirano okruženje za rad sa prostornim podacima i aplikacijama," in *YuINFO 2006*, Kopaonik, 2006.
- [66] D. Ivanović, Đ. Obradović, and Z. Konjović, "Softverska arhitektura Portala javnih servisa," presented at the Etran, 2007.
- [67] Đ. Obradović, D. Šolajić, and Z. Konjović, "Softverski sistem za administriranje procesa izvođenja nastave," presented at the YUInfo, Kopaonik, 2004.

Biografija

Đorđe Obradović je rođen u Kuli 06. maja 1973. godine. Diplomirao je na Fakultetu tehničkih nauka u Novom Sadu 1997. godine, na odseku Elektrotehnika i računarstvo, smer Računarstvo i upravljanje sistemima, usmerenje Računarstvo sa prosečnom ocenom 8,54. Diplomski rad sa temom „Dizajn i implementacija biblioteke klasa za modeliranje i obučavanje neuralnih mreža“ je odbranio sa ocenom 10 (deset).

Odmah po diplomiranju započeo je rad na Fakultetu tehničkih nauka kao stipendista Ministarstva za nauku i tehnologiju Republike Srbije. Radni odnos je zasnovao 1998. godine u Institutu za računarstvo i automatiku Fakulteta tehničkih nauka.

Poslediplomske studije završio je 2003. godine sa prosečnom ocenom 10, odbranom magistarske teze pod nazivom „Sofverski sistem za simulaciju antropomorfog hoda“.

Tokom rada na Fakultetu bio je angažovan kao asistent na sledećim predmetima: Osnovi računarstva, Primenjeno programiranje u tehnici, Osnovi veštačke inteligencije, Računarska inteligencija i Numerički algoritmi i numerički softver u inženjerstvu.

U toku studija i rada na fakultetu objavio je 25 radova u domaćim i stranim zbornicima sa konferencija i jedan rad u međunarodnom časopisu sa SCI liste. Sedam radova uključujući i rad publikovan u međunarodnom časopisu pripadaju oblasti disertacije.

Oženjen je, otac dvoje dece i živi u Novom Sadu. Od stranih jezika govori engleski jezik.

UNIVERZITET U NOVOM SADU
FAKULTET TEHNIČKIH NAUKA

Ključna dokumentacijska informacija

<i>Redni broj:</i> RBR	
<i>Identifikacioni broj:</i> IBR	
<i>Tip dokumentacije:</i> TD	Monografska dokumentacija
<i>Tip zapisa:</i> TZ	Tekstualni štampani materijal
<i>Vrsta rada (dipl., mag., dokt.):</i> VR	Doktorska disertacija
<i>Ime i prezime autora:</i> AU	mr Đorđe Obradović
<i>Mentor (titula, ime, prezime, zvanje):</i> MN	dr Zora Konjović, redovni profesor Fakultet tehničkih nauka, Novi Sad
<i>Naslov rada:</i> NR	Model računarskog simulacionog sistema za upravljanje geoprostorom u uslovima neodređenosti
<i>Jezik publikacije:</i> JP	srpski (latinica)
<i>Jezik izvoda:</i> JI	srpski (latinica) / engleski
<i>Zemlja publikovanja:</i> ZP	Srbija
<i>Uže geografsko područje:</i> UGP	Vojvodina

<i>Godina:</i> GO	2011
<i>Izdavač:</i> IZ	Autorski reprint
<i>Mesto i adresa:</i> MA	Fakultet tehničkih nauka, Trg Dositeja Obradovića 6, Novi Sad
<i>Fizički opis rada:</i> FO	(5/ 174 / 67 / 9 / 64 / 0 / 0) (broj poglavlja/strana/lit. citata/tabela/ slika/grafika /priloga)
<i>Naučna oblast:</i> NO	Računarske nauke i informatika
<i>Naučna disciplina:</i> ND	Geoinformatika
<i>Predmetna odrednica, ključne reči:</i> PO	Fazi skupovi, informacioni sistem, GIS, analiza slika
UDK	
<i>Čuva se:</i> ČU	Biblioteka Fakulteta tehničkih nauka, Trg Dositeja Obradovića 6, Novi Sad
<i>Važna napomena:</i> VN	Nema
<i>Izvod:</i> IZ	<p>Cilj - Cilj istraživanja je razvoj modela, implementacija prototipa i verifikacija računarskog simulacionog sistema koji omogućuje rad sa nepreciznim geoprostornim podacima u realnim uslovima, i proširivanje postojećim sistemima, razvoja novih algoritama za obradu, novih modela za reprezentaciju procesa i njihovu integraciju.</p> <p>Metodologija - Za modeliranje neodređenosti i nepreciznosti u podacima i procesima korišćen je fazi pristup. Za razvoj softverskog sistema korišćen je objektni pristup (UML 2.0.), model višeslojne distribuirane softverske arhitekture, kombinacija više objektno orijentisanih programskih jezika, i standardi iz oblasti prostornih podataka i procesa. Verifikacija je izvršena na primeru ekstrakcije prostornih osobina na</p>

	<p>osnovu rasterskih mapa, i implementirana su dva prostorna procesa u skladu sa definisanim modelima.</p> <p>Rezultati - Predloženi su modeli tačkastih (<i>fazi tačka</i>), pravolinijskih (<i>fazi duž</i>) i jednostavnih površinskih (<i>fazi trougao, fazi krug</i>) nepreciznih objekata pomoću fazi skupova. Date su definicije i osobine osnovnih prostornih operacija (prostorna merenja, prostorne funkcije i prostorne relacije). Dat je predlog za modeliranje geoprostornih procesa i algoritam za određivanje fazi konveksnog omotača koji se odlikuje smanjenom računskom složenošću u odnosu na postojeće algoritme.</p> <p>Ograničenja istraživanja/implikacije – Sistem ne obuhvata standardizovan format za razmenu nepreciznih prostornih podataka. To znači da se format razmene mora implementirati u softveru.</p> <p>Praktične implikacije - Predloženi modeli mogu se, pre svega, koristiti u geografskim informacionim sistemima, ali i u analizi slike, te drugim zadacima koji zahtevaju modelovanje realnog prostora (robotika i slično).</p> <p>Originalnost/vrednost – Originalni doprinosi su sledeći: novi modeli tačkastih (<i>fazi tačka</i>), pravolinijskih (<i>fazi duž</i>) i jednostavnih površinskih (<i>fazi trougao, fazi krug</i>) objekata, novi modeli osnovnih prostornih operacija (prostorna merenja, prostorne funkcije i prostorne relacije) i novi algoritam za određivanje fazi konveksnog omotača koji se odlikuje smanjenom računskom složenošću u odnosu na postojeće algoritme.</p>
<p><i>Datum prihvatanja teme od strane NN veća:</i> DP</p>	
<p><i>Datum odbrane:</i> DO</p>	

<i>Članovi komisije:</i>	
KO	
<i>Predsednik:</i>	dr Dušan Surla, profesor emeritus, Prirodno-matematički fakultet, Novi Sad
<i>Član:</i>	dr Dušan Petrovački, profesor emeritus, Fakultet tehničkih nauka, Novi Sad
<i>Član:</i>	dr Endre Pap, redovni profesor, Akademik VANU, Prirodno-matematički fakultet, Novi Sad
<i>Član:</i>	dr Miro Govedarica, vanredni profesor, Fakultet tehničkih nauka, Novi Sad
<i>Član:</i>	dr Leonid Stoimenov, vanredni profesor, Elektronski fakultet, Niš
<i>Član:</i>	dr Zora Konjović, redovni profesor, Fakultet tehničkih nauka, Novi Sad

UNIVERSITY OF NOVI SAD

FACULTY OF SCIENCE

Key words documentation

<i>Accession number:</i> ANO	
<i>Identification number:</i> INO	
<i>Document type:</i> DT	Monograph documentation
<i>Type of record:</i> TR	Textual printed material
<i>Content code:</i> CC	PhD thesis
<i>Author:</i> AU	Đorđe Obradović, MSc
<i>Mentor:</i> MN	Zora Konjović, PhD, full professor, Faculty of Sciences, Novi Sad
<i>Title:</i> TI	Model of the Computer Simulation System for Managing Geospace under Uncertainty Conditions
<i>Language of text:</i> LT	Serbian (Latin)
<i>Language of abstract:</i> LA	Serbian (Latin) / English
<i>Country of publication:</i> CP	Serbia
<i>Locality of publication:</i> LP	Vojvodina

<i>Publication year:</i> PY	2011
<i>Publisher:</i> PU	Author's reprint
<i>Publication place:</i> PP	Faculty of Technical Sciences, Trg Dositeja Obradovića 6, Novi Sad
<i>Physical description:</i> PD	(5/ 174 / 67 / 9 / 64 / 0 / 0) (chapters/pages/literature/tables/ pictures/graphs/appendix)
<i>Scientific field:</i> SF	Computer sciences and informatics
<i>Scientific discipline:</i> SD	Geoinformatics
<i>Subject, Key words:</i> SKW	Fuzzy sets, information systems, GIS, image analysis
UC	
<i>Holding data:</i> HD	
<i>Note:</i> N	None
<i>Abstract:</i> AB	<ul style="list-style-type: none"> • Purpose- The purpose is model development, software prototype implementation and verification of the computer simulation system which provides support for imprecise data processing under real conditions, as well as extensions to existing systems, development of novel algorithms for data processing, new models for processes' representation and their integration. • Design/methodology/approach - Fuzzy approach is used for modelling uncertainties and imprecision. For the software system's development object approach (UML 2.0), multi-tiered distributed software architecture,

	<p>combination of several object programming languages, and standards for geospatial data and processes are deployed. Verification is carried out by the example of spatial features extraction from raster maps which is applied to two spatial processes compliant to the proposed model.</p> <ul style="list-style-type: none"> • Findings- Models for point (<i>fuzzy point</i>), line (<i>fuzzy line</i>) and simple planar (<i>fuzzy triangle, fuzzy circle</i>) imprecise objects are proposed. The definitions and basic features of spatial operations (spatial measurements, spatial functions and spatial relations) are given. A proposal for modelling geospatial processes is given and the algorithm for fuzzy convex hull characterized by reduced computational complexity. • Research limitations/implications - The system does not comprise standardized format for imprecise data interchange. This means that interchange format should be implemented within the software. • Practical implications - The proposed models can be used, primarily for geographic information systems, but they can be also deployed in image analysis as well as tasks requiring modelling of the real space (robotics, etc.). <p>Originality/Value - The main original contributions are: novel models for point (<i>fuzzy point</i>), line (<i>fuzzy line</i>) and simple planar (<i>fuzzy triangle, fuzzy circle</i>) imprecise objects, novel models for imprecise spatial operations (spatial measurements, spatial functions and spatial relations), and the algorithm for fuzzy convex hull characterized by reduced computational complexity.</p>
<p><i>Accepted on Scientific Board on:</i></p> <p>AS</p>	
<p><i>Defended:</i></p> <p>DE</p>	

<i>Thesis Defend Board:</i>	
DB	
<i>President:</i>	Dušan Surla, PhD, full professor, Faculty of Sciences, Novi Sad
<i>Member:</i>	Dušan Petrovački, PhD, full professor, Faculty of Technical Sciences, Novi Sad
<i>Member:</i>	Endre Pap, PhD, PhD, full professor, Faculty of Sciences, Novi Sad
<i>Member:</i>	Miro Govedarica, PhD, full professor, Faculty of Technical Sciences, Novi Sad
<i>Member:</i>	Leonid Stoimenov, PhD, professor, Faculty of Electronic Engineering, Niš
<i>Member:</i>	Zora Konjović, PhD, full professor, Faculty of Technical Sciences, Novi Sad