

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6335

**STVARNOVREMENSKO LOCIRANJE
KORISNIKA JAVNOG GRADSKOG
PRIJEVOZA**

Marko Gašparac

Zagreb, lipanj 2019.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 6335

**STVARNOVREMENSKO LOCIRANJE
KORISNIKA JAVNOG GRADSKOG
PRIJEVOZA**

Marko Gašparac

Zagreb, lipanj 2019.

Sadržaj

1. Uvod
2. Suradno opažanje okoline
 - 2.1. Proces suradnog opažanja okoline
 - 2.2. Resursna ograničenja usluge
 - 2.3. Privatnost, sigurnost i očuvanje integriteta podataka
3. Razvoj programskog modula za automatizirano stvarnovremensko lociranje korisnika javnog gradskog prijevoza i suradno objavu podataka
 - 3.1. Osnovne ideje i funkcionalnosti
 - 3.2. Korišteni alati i tehnologije
4. Implementacijski detalji
 - 4.1. Struktura sustava i delegiranje odgovornosti komponentama
 - 4.2. Prikupljanje i inicijalna obrada velike količine podataka
 - 4.3. Učitavanje podataka i memorijska reprezentacija
 - 4.4. Očitavanje koordinata i lociranje putem GPS sustava
 - 4.5. Konačni automati i prijelazi između stanja
 - 4.6. Objava podataka na poslužitelj
5. Sigurnosni aspekti usluge i potrošnja resursa
6. Ispitivanje programskog modula i primjeri korištenja
7. Zaključak
8. Literatura

1. Uvod

Zagrebački električni tramvaj (kratica ZET), trgovačko društvo u izravnom vlasništvu Grada Zagreba, zaduženo je za organizirani prijevoz putnika u javnom gradskom prometu na području Grada Zagreba. Od 1891. godine, kada su 5. rujna prvi put planirano i organizirano gradskim ulicama krenula prva tramvajska kola, pa sve do 2005. godine i puštanja u promet prvog niskopodnog tramvaja, javni gradski prijevoz proživio je nekoliko etapa modernizacije i unaprijeđenja s ciljem pružanja prijevoznih usluga prema visokim europskim standardima.

Pravovremeno i lako dostupno informiranje korisnika javnog gradskog prijevoza o stanju u prometu jedan je od važnijih aspekata pružanja visokokvalitetne prijevozne usluge, te neizbježan koncept podizanja stupnja zadovoljstva korisnika prijevozne usluge na najvišu razinu. Užurbanim napretkom i razvojem informacijske i komunikacijske tehnologije, te ostalih grana primjenjenog računarstva, olakšava se i pospješuje interakcija s krajnjim korisnicima kao što i rastu njihovi zahtjevi i očekivanja od pružatelja usluga.

Na internetskom sjedištu ZET-a, pod rubrikom „O nama“, između ostalog objavljen je i sljedeći odlomak: „Praćenje aktualne prometne situacije, koordinirano i promptno postupanje u izvanrednim okolnostima te digitalnu razmjenu svih važnijih podataka u sklopu prometne mreže i voznog parka ZET-u omogućuje Sustav za nadzor i upravljanje prometom. Zahvaljujući automatiziranoj internoj distribuciji i obradi informacija iz tog se kompleksnog računalnog sustava stvaraju pretpostavke i za niz dodatnih usluga koje standard javnog gradskog prijevoza u Zagrebu podižu na najvišu razinu.“

Unatoč vrlo dobro organiziranom sustavu javnog gradskog prijevoza, te digitalnoj razmjeni svih važnijih podataka u sklopu prometne mreže, ZET još uvijek nije u potpunosti iskoristio mogućnosti informacijskih komunikacijskih tehnologija današnjice, te ih nije uspio u dovoljnoj mjeri približiti krajnjim korisnicima.

Podaci o stvarnim dolascima vozila na pojedina stajališta dostupni su isključivo u vidu digitalnih info panela postavljenih na frekventnijim stajalištima, okretištima ili kolodvorima. Informacije o stanju u prometu također je moguće dobiti pozivom na besplatni info telefon.

Početkom 2014. godine, motivirani prethodno opisanom problematikom, skupina studenata Fakulteta elektrotehnike i računarstva u Zagrebu, razvija mobilnu aplikaciju za Android operacijski sustav pod nazivom „Zetcheck“. Aplikacija je omogućila krajnjim korisnicima stvarnovremenske podatke o dolascima vozila javnog gradskog prijevoza bilo koje dnevne odnosno noćne linije, na bilo kojem stajalištu na području Grada Zagreba. Ubrzo nakon što je aplikacija postala dostupna široj javnosti, zabilježeno je opterećenje poslužitelja koji sustav koristi, te je tvrtka koja održava ZET-ov sustav onemogućila neovlašteni pristup. Iz ZET-a su poručili kako navedena aplikacija remeti rad sustava, te kako rade na vlastitom rješenju koje će uskoro postati dostupno. U listopadu iste godine ZET vrši integraciju svojih sustava sa Google-om, te omogućuje pregled voznog reda i planiranje ruta kretanja na Google kartama. Iako koristan iskorak, isti i dalje ne rješava u potpunosti problem informiranja korisnika gradskog prijevoza.

S obzirom da do današnjeg dana ZET nije ponudio niti jedno ozbiljnije rješenje, te da i dalje drži podatke o prometnoj mreži zatvorenima, javlja se potreba, ali i motivacija za razvijanjem vlastitog programskog rješenja temeljenog na suradnom opažanju okoline (engl. mobile crowdsensing).

2. Suradno opažanje okoline

Suradno opažanje okoline (eng. mobile crowdsensing) je tehnika kolektivnog prikupljanja podataka putem pametnih uređaja (npr. mobilnih telefona ili tableta) koji posjeduju senzore od strane velikog broja ljudi. Obradom tako prikupljenih podataka možemo provoditi analize, izmjeriti ili sa određenom pouzdanošću predvidjeti ishod nekog događaja ili procesa od općeg interesa.

Ovakav tip pametnih uređaja danas je sveprisutan. Tako na primjer, većina mobilnih telefona danas je u mogućnosti izmjeriti razinu osvjjetljenja putem kamere, odnosno šuma putem mikrofona. Pametni uređaji također mogu preuzeti vlastitu lokaciju sa GPS sustava, ili brzinu kretanja putem brzinomjera (engl. accelerometer). Kontinuiranim očitanjem podataka jednog ili više senzora, te njihovom kombinacijom iz višekorisničkih izvora u neposrednoj blizini, pouzdanost sustava raste, a potencijalna iskoristivost podataka više je nego očita. Zamislimo sustav koji je u mogućnosti proračunati stupanj zagađenosti zraka u nekoj urbanoj sredini na temelju očitavanja brzine kretanja te GPS lokacije vozača.

Pojam suradnog opažanja okoline prvi su puta definirali Raghu Ganti, Fan Ye i Hui Lei, inženjeri sa IBM T.J. Watson Research Center-a u New Yorku.

Suradno opažanje okoline grubo možemo podijeliti u tri tipa:

- **Suradno opažanje okoliša** – svrhu pronalaze u praćenju promjena u okolišu (npr. praćenje stupnja zagađenosti okoliša)
- **Suradno opažanje infrastrukture** – svrhu pronalaze u infrastrukturnim projektima (npr. praćenje stanja u prometu)
- **Društveno suradno opažanje** – svrhu pronalaze u praćenju društva (npr. društvene mreže)

Ukoliko razmatramo učešće krajnjih korisnika u procesu prikupljanju podataka, suradno opažanje možemo podijeliti i na sljedeći način:

- **Dobrovoljno suradno opažanje** – korisnik dobrovoljno pristaje sudjelovati u suradnom prikupljanju podataka.
- **Prisilno suradno opažanje** – suradno očitavanje, prikupljanje i djeljenje podataka odvija se bez pristanka korisnika, a ponekad i bez njegovog znanja.

Suradno opažanje okoline i sveprisutnost pametnih uređaja koji omogućuju takav tip prikupljanja podataka nastoje iskoristiti velike korporacije koje u tome vide priliku za prikupljanje potencijalno unovčivih podataka bez velikih, ako uopće ikakvih, kapitalnih ulaganja. Na temelju suradnog opažanja okoline, svoje usluge pružaju i korporacije poput Facebook-a, Google-a i Uber-a.

2.1. Proces suradnog opažanja okoline

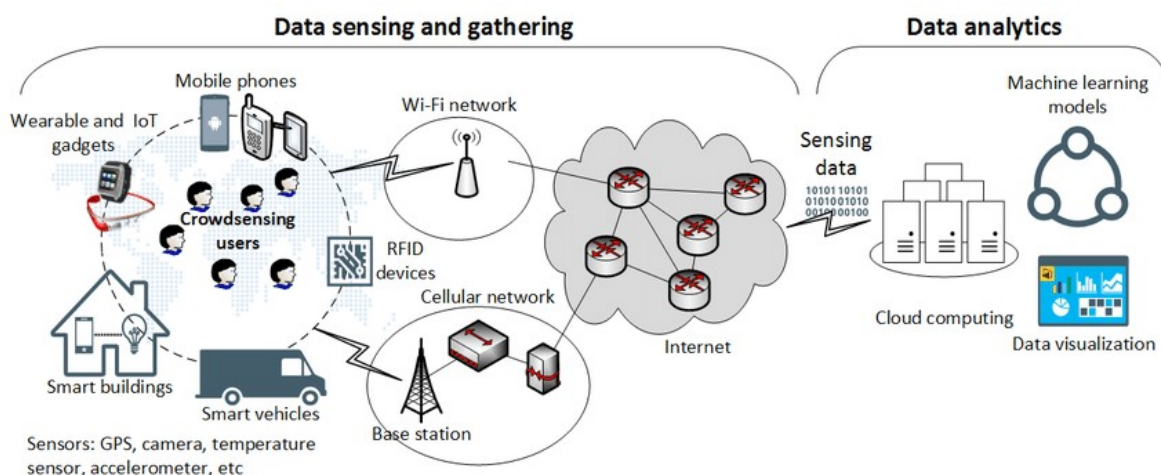
Proces suradnog opažanja okoline možemo podijeliti u tri glavna koraka:

- **Prikupljanje podataka**
- **Pohrana podataka**
- **Objavljivanje podataka**

Prikupljanje podataka oslanja se na koncept Interneta stvari. Izvor podataka može biti jedan, ili kombinacija više senzora poput kamere, mikrofona, GPS, brzinomjera i sl. Kod suradnog opažanja okoline, naglasak je na prikupljanju podataka iz više neovisnih izvora, čime postizemo bolju robusnost i pouzdanost sustava.

Razlikujemo nekoliko osnovnih modela prikupljanja podataka:

- **Ručno prikupljanje podataka** – korisnik ručno prikuplja podatke korištenjem aplikativne programske potpore (npr. snimanje videa kamerom)
- **Automatsko prikupljanje podataka kontrolirano od strane korisnika** – programska potpora može automatski generirati i sakupljati podatke, potaknuto odnosno omogućeno od strane korisnika
- **Automatsko prikupljanje podataka potaknuto kontekstom** – programska potpora može automatski generirati i sakupljati podatke, kada su zadovoljeni određeni uvjeti (npr. korisnik je u određenom trenutku ušao u određeno područje)



Slika 1. Model suradnog opažanja okoline

Prikupljene podatke privremeno je prije objavljivanja moguće pohraniti na strani klijenta te izvršiti njihovu obradu. Čest je proces deduplikacije podataka (engl. data deduplication). To je postupak uklanjanja redundatnih podataka radi smanjenja troškova buduće pohrane i prijenosa. Prije objavljivanja podataka, iste je također potrebno podvrgnuti procesu šifriranja i dodavanja šuma, kako bi se osigurali sigurnosni aspekti usluge.

2.2. Resursna ograničenja usluge

Budući da je usluga suradnog opažanja okoline temeljena na pokretnim uređajima, prilikom oblikovanja u obzir moramo uzeti i resursna ograničenja. Jedno od glavnih resursnih ograničenja danas prosječna je duljina trajanja baterije. Prilikom punjenja, pokretni uređaji postaju stacionarni i u većini su slučajeva neprikladni za usluge suradnog opažanja.

Korištenje usluga GPS lociranja izrazito je zahtjevno po bateriju. Zato je neophodno razmotriti i ostale manje zahtjevne modele, poput preuzimanja lokacije sa WiFi-a ili 3G odnosno 4G mreže, uzimajući pritom u obzir znatno slabiju preciznost i pouzdanost takvog rješenja. Ukoliko nam je preciznost pak bitna, mogli bismo razmotriti i model rjeđeg očitavanja podataka, ili pak obustavljanja onda kada za time nema potrebe. Uzmimo ponovno primjer izračuna razine zagađenja zraka vozača. Ukoliko se korisnik svakodnevno kreće sličnom rutom (npr. na putu od kuće do posla), dovoljno je zadanu rutu očitati jednom. Idući ćemo puta na temelju prethodnih kretanja moći zaključiti rutu kretanja, te bez potrebe za GPS-om efikasno odraditi jednako zahtjev posao uz manji utrošak energije.

Ukoliko pak razmjenjujemo višemedijski sadržaj, neophodno je voditi računa o raspoloživoj širini prijenosnog pojasa (engl. bandwidth), kao i ograničenjima u količini prijenosa podataka te ostalim zahtjevima poput iznosa kašnjenja te kolebanja kašnjenja. Radimo li na primjer s kamerom, te ustanovimo da su dvije slike uslikane na istoj lokaciji, u svrhu poboljšanja performansi sustava, nepotrebno je prenijeti i treću – ona ne nosi nikakvu dodatnu informaciju! Neki od ovih problema motivacija su za razvoj pete generacije mobilnih mreža.

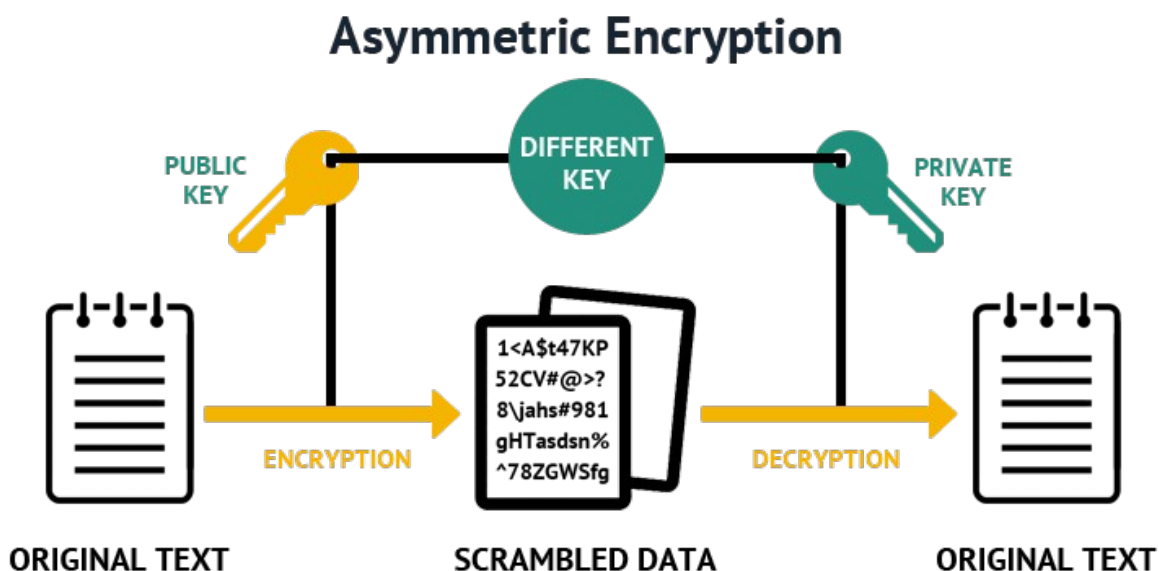
Ovim jednostavnim razmatranjem dolazimo do zaključka kako je prilikom oblikovanja sustava za suradno opažanje okoline od neophodne važnosti vođenje računa o oskudnosti dostupnih resursa. Pritom je potrebno pronaći odgovarajući kompromis između zahtjevanih visokih performansi sustava, te niske potrošnje, uzimajući pritom u obzir visok stupanj zadovoljstva korisnika.

2.3. Privatnost, sigurnost i očuvanje integriteta podataka

Podaci koje prikupljamo suradnim opažanjem okoline mogu sadržavati povjerljive informacije, poput korisnikove kućne lokacije, uobičajene rute kretanja ili slično. S obzirom da iste uobičajeno šaljemo putem mreže, neophodno je osigurati njihovu privatnost i očuvanje integriteta.

Nekoliko je osnovnih postupaka za ostvarenje istog:

- **Sakrivanje povjerljivih podataka** (engl. anonymization) – iz povjerljivih se podataka uklanja identifikacija korisnika. Iako povjerljivi podaci ostaju pohranjeni i prenose se mrežom, na temelju njih nije moguće zaključiti kome oni pripadaju.
- **Šifriranje podataka** (engl. secure multiparty computation) – povjerljivi se podaci šifriraju prije slanja komunikacijskim kanalom, te se pretvaraju u oblik kojeg je nemoguće raspoznati bez ključa za šifriranje.



Slika 2. Metoda asimetrične enkripcije podataka

- **Dodavanje šuma** (eng. data perturbation) – podacima se prije objavljivanja dodaje šum, koji ne utječe na njihovu točnost
- **Decentralizirana prikupljanje podataka** (engl. aggregation-free data collection) – prostorno-vremenski podaci se metodom prosljeđivanja poruka pohranjuju decentralizirano, u nekoliko različitih čvorova, osiguravajući pritom njihovu privatnost

Budući da je većina sustava suradnog opažanja okoline otvorena novim korisnicima, korisnici mogu namjerno ili nenamjerno generirati veću količinu pogrešne informacije te time narušiti pouzdanost sustava. S obzirom da je gotovo nemoguće procijeniti vjerodostojnost zaprimljenih podataka, većina sustava temeljena je na procjeni kvalitete informacije i njihovom filtriranju uspoređujući pritom podatke zaprimljene od korisnika u neposrednoj blizini, te odbacujući one koji značajnije odstupaju od prosjeka.

Jedan od glavnih nedostataka prethodno razmatranih postupaka jest njihova kompleksnost te dodatan negativan učinak na potrošnju energije.

3. Razvoj programskog modula za automatizirano stvarnovremensko lociranje korisnika javnog gradskog prijevoza i suradno prikupljanje podataka

Potaknut problematikom nedostupnosti podataka o stvarnovremenskoj lokaciji vozila javnog gradskog prijevoza u Gradu Zagrebu, ali i mogućnosti prikupljanja istih suradnim opažanjem okoline te sveprisutnošću mobilnih pametnih uređaja, razvijen je programski modul za automatizirano stvarnovremensko lociranje korisnika javnog gradskog prijevoza i suradno prikupljanje podataka.

3.1. Osnovne ideje i funkcionalnosti

Glavna ideja programskog modula iskoristiti je činjenicu da većina korisnika javnog gradskog prijevoza posjeduje pametni telefon te je povezana na Internet putem javne pokretne mreže treće odnosno četvrte generacije. U sve pametne telefone danas je također ugrađena funkcionalnost preuzimanja trenutne lokacije putem GPS sustava. Kombinirajući prethodne dvije tehnologije, te skup javno dostupnih podataka o infrastrukturi javnog gradskog prijevoza Grada Zagreba koji su objavljeni na internetskim stranicama ZET-a, razvijen je inteligentni sustav koji je u mogućnosti periodičkim očitavanjem podataka sa GPS-a pretpostaviti da korisnik u određenom trenutku koristi usluge javnog gradskog prijevoza, proračunati trenutno stajalište i liniju kojom se vozi, te u ovisnosti o željenoj razini pouzdanosti proračunatih podataka nakon određenog broja stajališta poslati željene podatke prema poslužitelju.

Na ovaj način, korisnik ne mora ručno odabirati kada je i gdje ušao, odnosno izašao iz vozila, te na kojoj liniji se nalazi. Ovakvim pristupom aplikacija postaje vrlo ugodna za korištenje, što indirektno utječe na korisničko iskustvo, te potiče veći broj novih korisnika. Aplikaciju je dovoljno preuzeti, instalirati te pokrenuti na pametnom uređaju. Podsjetimo se, veći broj korisnika usluga suradnog opažanja okoline ujedino znači i veći broj dostupnih podataka, kao i njihovoj većoj pouzdanosti.

Poslužitelj koji prikuplja podatke od korisnika javnog gradskog prijevoza, iste mora filtrirati, obraditi i pohraniti. Na temelju prikupljenih podataka, poslužitelj može npr. ponuditi interaktivnu kartu te iscrtati trenutnu lokaciju svih vozila, prikazati opterećenje na određenom djelu prometne mreže ili proračunati trenutak dolaska na određeno stajalište, bilo kada i bilo gdje. Korisnost ovakvih podataka višestruka je, kako za krajnje korisnike, tako i pružatelje prijevoznih usluga.

Programski modul osmišljen je i oblikovan kao nadogradnja za neku od postojećih aplikacija javnog gradskog prijevoza, te kao takav nema direktno implementirano korisničko sučelje. Isti se pokreće kao zaseban proces, odnosno pozadinski servis, te je aktivan za vrijeme rada operacijskog sustava. Kako bi funkcionirao ispravno, neophodno je prilikom instalacije dodjeliti privilegiju pristupa

lokaciji putem GPS-a, te za vrijeme korištenja na pametnom telefonu omogućiti pristup internetu, te uslugu Lokacije.

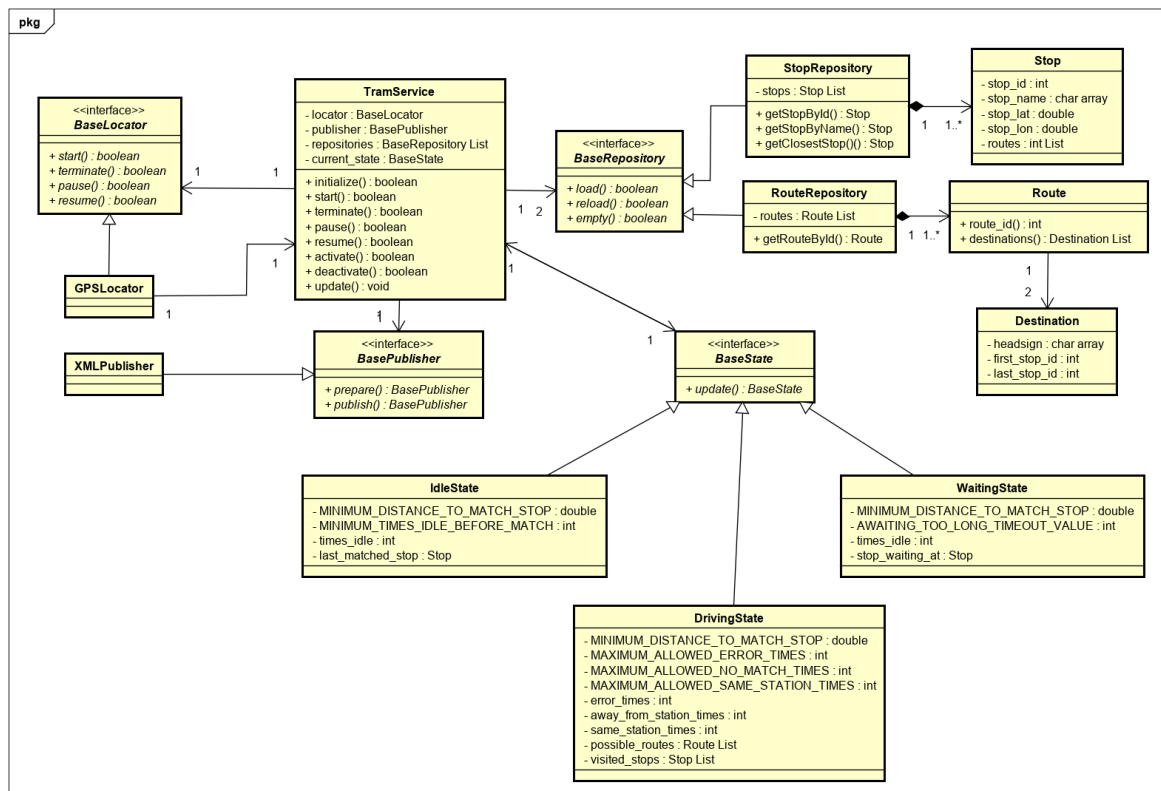
3.2. Korišteni alati i tehnologije

Sustav je implementiran za operacijski sustav Android, koristeći programski jezik Java, te razvojno okruženje Android Software Development Kit.

4. Implementacijski detalji

4.1. Struktura sustava i delegiranje odgovornosti komponentama

Strukturni UML dijagram razreda programskog modula za stvarnovremensko lociranje korisnika javnog gradskog prijevoza i suradnu objavu podataka prikazan je na Slici 3. Zbog zadržavanja jednostavnosti, na dijagramu su prikazani samo osnovni razredi bitni za razumjevanje principa rada programskog modula.



Slika 3. UML dijagram razreda programskog modula

Glavna komponenta unutar koja se pokreće program i koja usklađuje rad svih ostalih dijelova programa, klasa je TramService. Klasa TramService pohranjuje, te preko apstraktnih sučelja pristupa sljedećim komponentama:

- **BaseLocator** – apstraktno sučelje preko kojeg pristupamo razredu GPSLocator. GPSLocator implementira i zadužen je za pribavljanje podataka sa GPS-a u komunikaciji sa operacijskim sustavom putem ugrađenog razreda LocationManager Android Software Development Kit-a. GPSLocator ujedino služi i kao „generator takta“, jer istekom 30 sekundi, ukoliko se korisnik nalazi u Idle stanju, odnosno 10 sekundi, ukoliko se korisnik nalazi u Waiting ili Driving stanju, očitava podatke za GPS-a, te ih propagira prema osnovnom razredu TramService koji potom delegira odgovornost obrade očitanih podataka ostalim komponentama.

- **BasePublisher** – apstraktno sučelje preko kojeg pristupamo razredu XMLPublisher. XMLPublisher zadužen je za pripremu podataka u obliku XML dokumenta, te njihovu objavu na poslužitelj korištenjem protokola HTTP. Osnovna klasa TramService proziva XMLPublisher u slučaju kada obradom očitanih podataka sa GPS-a ustanovi da je korisnik javnog prijevoza stigao na novo stajalište. Neki od podataka koji se šalju jesu trenutno stajalište, kao i moguće linije vožnje.
- **BaseRepository** – apstraktno sučelje preko kojeg pristupamo repozitorijima podataka. Dva su glavna repozitorija podataka: StopRepository i RouteRepository. StopRepository sadrži instance razreda Stop, koje su memorijska reprezentacija stajališta u Gradu Zagrebu. RouteRepository sadrži instance razreda Route, koje su memorijska reprezentacija linija javnog gradskog prijevoza Grada Zagreba. Objekti tipa Route također pohranjuju kolekciju objekata tipa Destination, koji reprezentiraju sve smjerove te linije, odnosno početno stajalište, krajnje stajalište, te natpis na vozilu tog smjera.
- **BaseState** – razred TramService također pohranjuje trenutno od tri moguća stanja: IdleState, WaitingState te DrivingState. IdleState početno je stanje programskog modula. Trenutnom stanju prosljeđuju se podaci o očitanoj stanju sa GPS-a. Trenutno stanje potom na temelju ulaznih podataka, te prethodnih stanja računa ostaje li u trenutnom stanju ili prelazi u jedno od preostala dva stanja.

Razred TramService također implementira sljedeće javne metode:

- **initialize()** – vrši inicijalizaciju programskog modula. Inicijalizacija je obavezna prije pokretanja modula. Inicijalizacija podrazumjeva učitavanje podataka iz datoteka u odgovarajuće repozitorije.


```

@Override
public boolean initialize() {
    if(this.status != TramServiceStatus.TRAM_LOCATOR_CREATED)
        return false;
    if(!stopRepository.load() || !routeRepository.load())
        return false;
    this.status = TramServiceStatus.TRAM_LOCATOR_INITIALIZED;
    return true;
}

```

Slika 4. Metoda za inicijalizaciju programskog modula

- **start()** – započinje rad programskog modula. Pokreće osluškivanje GPS-a svakih 30 sekundi, te postavlja modul u Idle stanje.

```

@Override
public boolean start() throws SecurityException {
    if(this.status != TramServiceStatus.TRAM_LOCATOR_INITIALIZED)
        return false;

    if(!baseLocator.start())
        return false;

    this.status = TramServiceStatus.TRAM_LOCATOR_IDLE;
    this.state = new IdleState( locator, this);
    return true;
}

```

Slika 5. Metoda za pokretanje programskog modula

- **terminate()** – završava rad programskog modula. Završava osluškivanje GPS-a.

```

@Override
public boolean terminate() {
    if(this.status == TramServiceStatus.TRAM_LOCATOR_TERMINATED)
        return false;
    if(!baseLocator.terminate())
        return false;
    this.status = TramServiceStatus.TRAM_LOCATOR_TERMINATED;
    return true;
}

```

Slika 6. Metoda za završetak rada programskog modula

- **pause()** i **resume()** – služe za privremeno zaustavljanje i nastavak rada programskog modula.

- **activate()** i **deactivate()** – služe za povećanje frekvencije očitavanja podataka GPS-a na svakih 10 sekundi u slučaju prelaska u aktivno stanje (WaitingState ili DrivingState) odnosno smanjenje frekvencije na svakih 30 sekundi u slučaju prelaska u inaktivno stanje (IdleState)

```
@Override
public boolean activate() {
    if(this.status != TramServiceStatus.TRAM_LOCATOR_IDLE)
        return false;

    baseLocator.setFrequency(10000);
    this.status = TramServiceStatus.TRAM_LOCATOR_ACTIVE;
    return true;
}

@Override
public boolean deactivate() {
    if(this.status != TramServiceStatus.TRAM_LOCATOR_ACTIVE)
        return false;
    baseLocator.setFrequency(30000);
    this.status = TramServiceStatus.TRAM_LOCATOR_IDLE;
    return true;
}
```

Slika 7. Metode za regulaciju frekvencije očitavanja podataka sa GPS-a

Međusobnom suradnjom opisanih komponenti preko glavnog razreda TramService pospješuje se kompaktnost izvedbe. Programski modul oblikovan je sukladno oblikovnim obrascima te općim načelima oblikovanja prilagodljivih i modularnih programskih podsustava.

Primjenom načela inverzije ovisnosti te jedinstvene odgovornosti omogućena je jednostavna nadogradnja i dodavanje novih funkcionalnosti bez izmjene postojećeg koda. Tako je npr. lako dodati nov način dohvaćanja lokacije putem mobilne mreže ili WiFi-a, umjesto GPS-a, odnosno promijeniti format objavljivanja podataka sa XML-a na nešto moderniji JSON.

