Inteligentni agenti u rješavanju problema zagušenosti u prometu

Ziterbart, Bruno

Undergraduate thesis / Završni rad

2015

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Split, University of Split, Faculty of science / Sveučilište u Splitu, Prirodoslovno-matematički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:166:618275

Rights / Prava: Attribution-NonCommercial 4.0 International/Imenovanje-Nekomercijalno 4.0 međunarodna

Download date / Datum preuzimanja: **2024-03-29**

Repository / Repozitorij:

Repository of Faculty of Science

zir.nsk.hr

UNIVERSITY OF SPLIT

dabar

DIGITALNI AKADEMSKI ARHIVI I REPOZITORIJI

SVEUČILIŠTE U SPLITU
PRIRODOSLOVNO MATEMATIČKI FAKULTET

ZAVRŠNI RAD

INTELIGENTNI AGENTI U RJEŠAVANJU PROBLEMA ZAGUŠENOSTI U PROMETU

Bruno Ziterbart

Split, srpanj 2015.

Sadržaj

Uvod 5

1. Inteligentni agenti 6

1.1. Arhitekture inteligentnih agenata 7

1.1.1. Jednostavan reaktivan agent (*eng. Simple Reflex Agent*) 8

1.1.2. Cilju orijentiran agent (*eng. Goal Based Agent*) 9

1.1.3. Koristi orijentiran agent (*eng. Utility Based Agent*) 10

2. Višeagentski sustav (*eng. Multiagent System*) 11

2.1. Jezici za komunikaciju između agenata (*eng. Agent Communication Languages)* 13

2.1.1. KQML (*Knowledge Query and Manipulation Language*) 13

2.1.2. FIPA ACL (*eng. FIPA agent communication language)* 17

2.1.3. Telescript 18

3. Višeagentski sustavi u prometu 19

3.1. MARLIN-ATSC 19

3.2. Singapore ITS 20

3.3. Usporedba MARLIN-a i Singaproe ITS-a 20

4. Realizacija projekta 21

4.2. Definiranje zahtijeva 21

4.3. Odabir tehnologija i alata 22

4.4. Implementacija 22

4.4.1. Inteligentni Agenti 22

4.4.2. Okolina 27

4.4.3. Komunikacija inteligentnih agenata 27

4.4.4. Uloga korisnika 30
4.4.5. Rezultati simulacije 31
Zaključak 34
Slike 35
Tablice 36
Kod 36
Literatura 37
Sažetak 37
Summary 38
Skraćenice 38

Uvod

Svakodnevno se susrećemo s različitim problemima u prometu. Kako je sudjelovanje u prometu sastavni dio života ljudi, trebalo bi ga nastojat učinit što je moguće ugodnijim. Živimo u 21. stoljeću, broj automobila i ostalih vozila na prometnicama raste svakim novim danom, a skupa s njima rastu i problemi. Prosječan stanovnik europskog grada izgubi više od godinu dana života zbog prometnih zagušenja ili neposjedovanja točnih informacija o odvijanju prometa. Mnogi stručnjaci tvrde kako bi u rješavanje novonastalih problema u odvijanju prometa trebalo uključit neki oblik inteligencije. Opći ciljevi su povećanje sigurnosti i propusnosti u prometu te što je moguće veća ekološka prihvatljivost. Navedene ciljeve je moguće ostvariti prikupljanjem informacija u realnom vremenu, obradom prikupljenih podataka i komunikacijom ne samo sudionika nego i ostalih elemenata prometa putem mreže. Kako sam student koji svakodnevno ima predavanja na različitim lokacijama, moram koristiti gradski prijevoz koji zbog zagušenja u prometu često nije pouzdan. Uzevši to u obzir, ideja za povećanjem protočnosti prometa mi se svidjela i odlučio sam tome posvetiti svoj završni rad. Nakon odabira teme, preostalo je još samo odabrati način realizacije projekta. Tijekom školovanja na ovome fakultetu, napravio sam nekoliko projekata koji su se temeljili na objektno orijentiranom programiraju. S tim predznanjem nije mi teško započeti izradu višeagentskog sustava. Odlučio sam napraviti višeagentski sustav zato što volim nove izazove jer na taj način dodatno učim nove stvari, a i sama izrada projekta mi bude zanimljivija. Štoviše, inteligentni agenti su područje usko vezano za područje umjetne inteligencije koje mi se posebno sviđa. U nastavku rada nalaze se teoretske osnove područja koja su vezana za ovaj završan rad i način realizacije projekta.

1. Inteligentni agenti

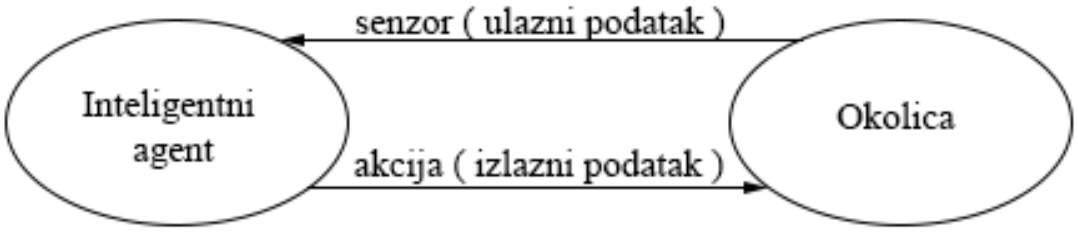
Postoje mnoge definicije inteligentnih agenata (*eng. intelligent agents*), a posebno se ističe ona koja ih definira kao programsku podršku koja ima sposobnost da samostalno i bez intervencije korisnika izvršava postavljeni zadatak, u tom slučaju, krajnji korisnik se izvještava o završetku zadatka ili samoj pojavi očekivanog događaja [1]. Kako bi što uspješnije riješio dani zadatak, agent je u interakciji s okolinom u kojoj se mogu nalaziti i drugi inteligentni agenti. Prema tome, agent radi u nekom okruženju koje može bit stvarno ili virtualno. Okruženje opaža putem različitih senzora kao što su npr. senzor za boju ili senzor za temperaturu. Osim što opaža okruženje u kojem se nalazi, on na njega i djeluje. Agent može imat jedan ili više zadataka za izvršiti, u tom slučaju on sam odlučuje poredak ili zadatak koji će izvršiti. Nerijetko agent ima i sposobnost učenja. Kod inteligentnih agenata, inteligencija je sposobnost kako agenti prihvaćaju zadane ciljeve, te kako ih izvršavaju. Inteligencija odražava razinu kvalitete razmišljanja i naučenog ponašanja. Inteligentni agenti su prije svega programske komponente i služe kao posrednici između svojih korisnika i računala. Prisutni su u našem svakodnevnom životu. Vezano za temu koju obrađuje ovaj završni rad, danas imamo inteligentne agente koji nam daju izvješće o stanju na cesti, pomažu paralelno parkirati automobil, postoje i sustavi za izbjegavanje sudara, kontrolu krivoga smjera, adaptivno upravljanje vozilom u pokretu itd. Uz inteligentne agente se često veže i pojam PACO (*eng. PAGE)* , što dolazi od , percepti (*eng. percepts*) , akcije (*eng. actions*), cilj (*eng. goal)* i okolica (*eng. environment*). Kako bi se što lakše objasnio pojam, sljedeća tablica prikazuje nekoliko primjera (Tablica 1).

Inteligentni agent	*Agent za upravljanje vozilom*	*Pametna kuća*
Percepti	*Kamera, GPS, brzinomjer ...*	*Termometar, sat, senzor pokreta, zvučni senzor ...*
Akcije	*Kreni, stani, ubrzaj, uspori ...*	*Ugrij/ohladi prostoriju, upali/ugasi svjetla ...*
Cilj	*Sigurna, brza i ugodna vožnja po propisima*	*Smanjiti troškove, bolje iskoristiti energiju ...*
Okolica	*Cesta, parkiralište ...*	*Prostorije s ljudima i bez njih, različito doba dana ....*

Tablica 1 Inteligentni agenti - PACO (eng. PAGE)

1.1. Arhitekture inteligentnih agenata

U današnje vrijeme, inteligentni agenti imaju široku primjenu, koriste se u komercijalnim aplikacijama, elektroničkom poslovanju, računalnim igrama, različitim poslovnim procesima i mnogim drugim mjestima. U spomenutim primjerima njihova uloga se bitno razlikuje. Tako razlikujemo i arhitekturu inteligentnih agenata. U suštini, arhitektura inteligentnih agenata je skup koji sadrži strukture podataka, operacije nad tim strukturama podataka i tok podatak između njih [2]. Svim inteligentnim agentima je zajedničko da opažaju okolinu preko senzora, te pomoću aktuatora u nju šalju određenu akciju (Slika 1).

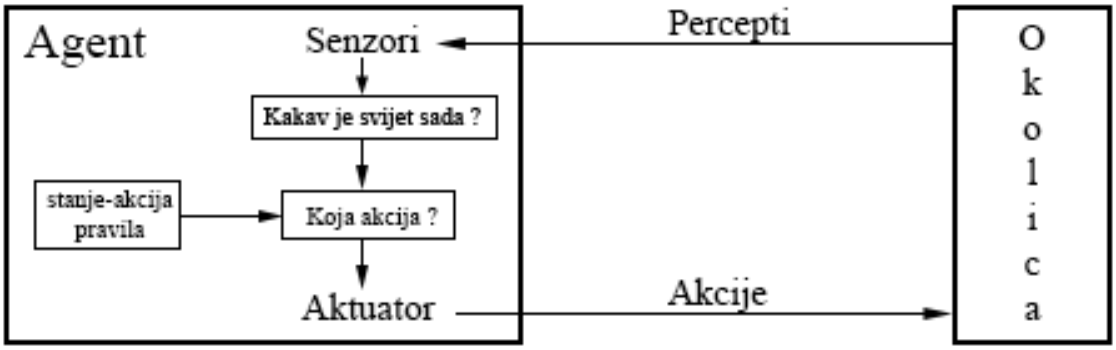


Slika 1 Odnos inteligentnog agenta i okolice

Da bi se agent smatrao inteligentnim, treba bit autonoman. U tom slučaju je sposoban djelovati bez neposredne intervencije i ima kontrolu nad svojim akcijama. Također treba bit i adaptivan. Adaptivnost označava mogućnost prilagodljivosti i učenja. Kako je već ranije spomenuto, agent se nalazi u nekoj okolini, moguće je da se u toj okolini nalaze i drugi agenti. Agent treba bit društven, odnosno sposoban za komunikaciju s drugim agentima.

1.1.1. Jednostavan reaktivan agent (*eng. Simple Reflex Agent*)

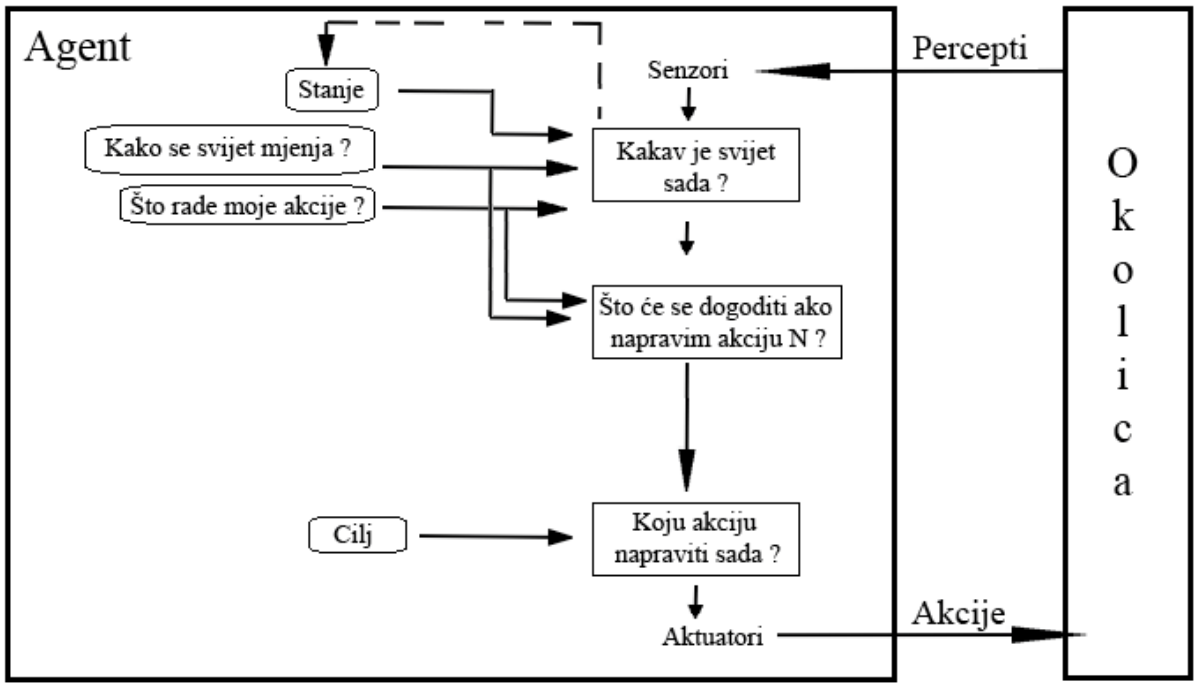
Kao što i sam naziv kaže, riječ je o jednostavnom agentu čije akcije ovise o perceptima, povijest mu ništa ne znači nego se oslanja na ono što sam trenutno doživi. Implementiacija se provodi po tzv. stanje-akcija pravilima (*eng. condition-action rules)*. Kao primjer će nam poslužiti robot za razvrstavanje pošte. Njegova okolina je poštanski ured, a zadaća mu je svako pismo staviti u pretinac ovisno o tome u koju državu se šalje, npr. ako pismo ide u Split, onda stavi pismo u pretinac „Hrvatska“ ili npr. ako pismo ide u „Berlin“, onda stavi pismo u pretinac „Njemačka“. (Slika 2) detaljnije prikazuje arhitekturu jednostavnog reaktivnog agenta.



Slika 2 Jednostavan reaktivan agent

1.1.2. Cilju orijentiran agent (*eng. Goal Based Agent*)

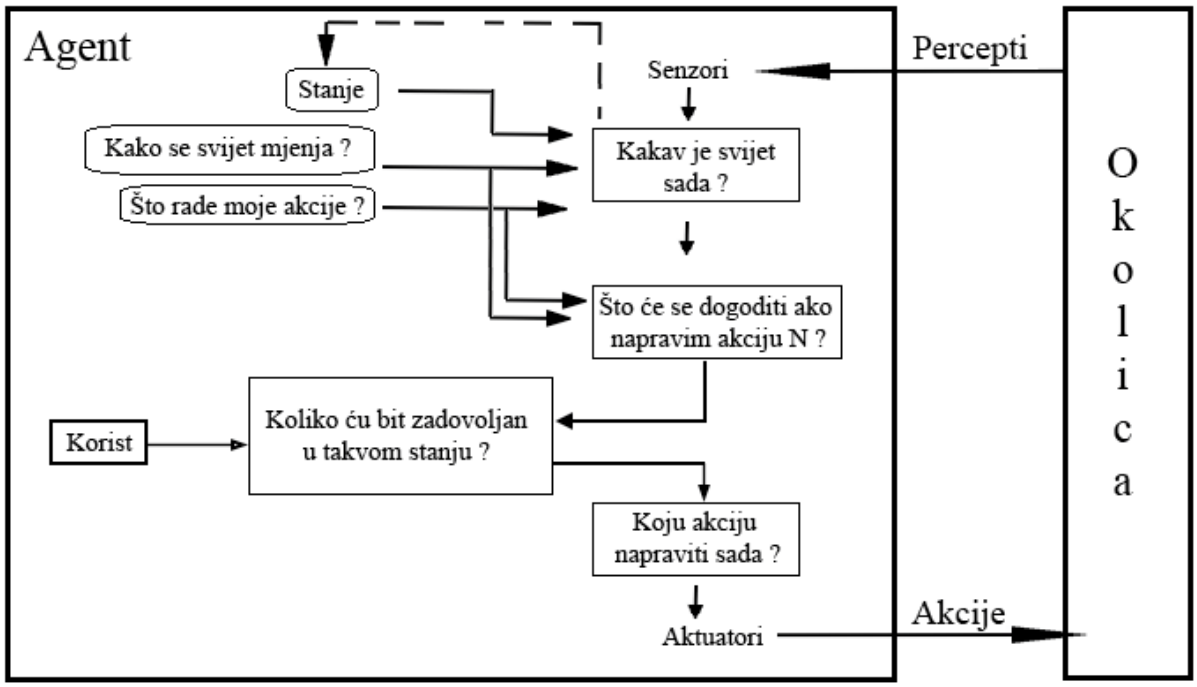
Cilju orijentirani agent je nešto složeniji od prethodno spomenutog. Ovakav tip agenata ima namjeru, nije samo reaktivan. Nastoji ispuniti svoj cilj, informacija vezana za cilj agenta opisuje situaciju ili stanje koje je poželjno. Tako npr. ukoliko agent ima više mogućnosti, odabire onu koju mu pomaže u ispunjavanju cilja. Dakle, ima sposobnost rasuđivanja. Pomoću pretraga i planiranja pravi redoslijed akcija koje ga dovode do ispunjavanja cilja. Same pretrage i planiranja su dio umjetne inteligencije, što područje vezano za inteligentne agente čini dodatno zanimljivim. Bitna odlika cilju orijentiranog agenta je to što u obzir uzima i budućnost, postavlja si piranje: „Što će se dogoditi ako ja napravim akciju N ? „. Primjer je robot koji se nalazi u labirintu, njegov cilj je izaći iz labirinta. Može postojati jedan ili više izlaza iz labirinta. Robot ima mogućnost izbora smjera u kojem se kreće. Na njemu je da odluči gdje će se kretati i na koji način će ispuniti svoj cilj. Arhitektura cilju orijentiranog agenta je prikazana na (Slika 3).



Slika 3 Cilju orijentirani agent

1.1.3. Koristi orijentiran agent (*eng. Utility Based Agent*)

Za razliku od cilju orijentiranog agenta koji razlikuje samo ciljno i ne-ciljno stanje, koristi orijentiran agent ima mjeru prema kojoj vidi koliko je poželjno određeno stanje. Dakle, agent u svakom trenutku zna koliko je „zadovoljan“ trenutnim stanjem. Općenito, agent može vrednovati stanja, uspoređivati ih i odlučiti s kojim stanjem je zadovoljniji, odnosno koje stanje mu je poželjnije. Ovakav tip agenta treba mijenjati i pratiti okolicu. Npr. agent čisti određenu prostoriju, agent čisti i ispituje okolicu, zna koliko je očistio i može vrednovati svoj učinak. Moguće je da nije „zadovoljan“ čistoćom prostorije pa nastavlja radit, nakon nekog vremena dodatnog čišćenja agent postaje zadovoljniji, u trenutku kada postane „potpuno zadovoljan“ svojim učinkom, prekida čišćenje. Arhitektura koristi orijentiranog agenta je prikazana ispod na (Slika 4).



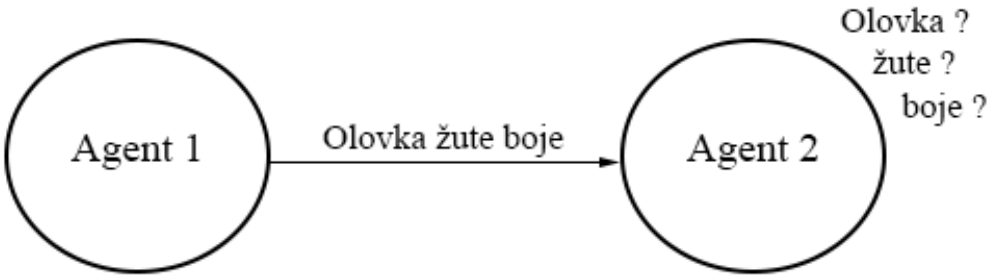
Slika 4 Koristi orijentiran agent

Osim spomenutih arhitektura inteligentnih agenata, postoje i mnogo složenije arhitekture kao što su hibridni agenti, agenti s deduktivnim rasuđivanjem itd. Kako tema ovog završnog rada nije usko vezana za arhitekture inteligentnih agenata, nećemo detaljnije ulaziti u razradu istih.

2. Višeagentski sustav (*eng. Multiagent System*)

Višeagentski sustav je sustav sastavljen od barem dva agenta koja međusobno komuniciraju kako bi izvršili složene zadatke [3]. Takvi sustavi su se počeli proučavati još 1980., dok petnaestak godina kasnije postaju opće priznati kao znanstveno područje. Od tada broj ljudi zainteresiranih za višeagentske sustave enormno raste. Unutar višeagentskih sustava postoje mnoga područja istraživanja kao što su semantika, komunikacija, prikazivanje znanja, planiranje, učenje itd. Veliki broj područja koja se nalaze unutar višeagentskih sustava su se razvila iz umjetne inteligencije. Višeagentski sustavi su s vremenom postali metafora za razumijevanje i izgradnju sustava koje mi danas ugrubo zovemo umjetni društveni sustavi (*eng. artificial social systems)*. Takvi sustavi imaju skup ograničenja vezanih za agentovo ponašanje u višeagentskom okruženju [4]. Uloga

ograničenja je da omoguće agentu postojanje u dijeljenoj okolini i postizanje ciljeva u prisutnosti drugih agenata. Da bi to bilo moguće, bitno je da agenti imaju dobar način komunikacije. Npr. neka dva agenta međusobno komuniciraju zato što jedan agent želi kupiti neki predmet od drugoga. Bitno je da za opisivanje predmeta koriste istu terminologiju. Prvi agent mora bit u mogućnosti reći koja su točna svojstva predmeta, dok drugi ta svojstva mora razumjeti i znati što npr. znači „boja“, a što “žuta“ ili „plava“ znači. Komunikacija dva agenta prikazana je ispod na (Slika 5)

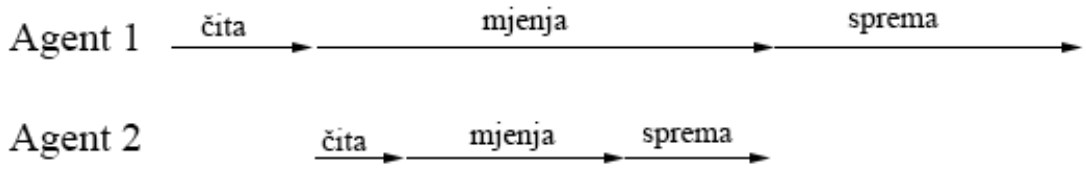


Slika 5 Komunikacija dva agenta

Komunikacija je oduvijek bila bitna tema u području računarstva, do sada su razvijeni mnogi protokoli kako bi komunikacija unutar sustava ili komunikacija između nekoliko sustava bila moguća. Tipičan problem je sinkronizacija više procesa, što se uvelike istraživalo 1970.-ih i 1980.-ih. U suštini, dva procesa trebaju bit sinkronizirana ukoliko postoji mogućnost da interferiraju na njima štetan način. Klasičan primjer takvog problema interferencije dva procesa je neispravno ažuriranje vrijednosti neke varijable. Zbog jednostavnijeg razumijevanja, problem će se detaljnije razložiti. Recimo da imamo jedan višeagentski sustav. Dva agenta *a1* i *a2* mogu promijeniti vrijednost varijable *v*. Agent *a1* započinje proces mijenjanja vrijednosti varijable *v*, prvo pročita trenutnu vrijednost varijable *v,* zatim mijenja vrijednost varijable i na kraju novu vrijednost spremi u varijablu *v.* Između čitanja i spremanja nove vrijednosti koje je izvršavao agent *a1,* agent *a2* je promijenio vrijednost varijable *v* spremajući neku vrijednost u nju. Kada agent *a1* spremi

vrijednost u varijablu *v,* izmjena vrijednosti koju je izvršavao agent *a2* je izgubljena. Sličan problem se često pojavljuje u programima koji komuniciraju pomoću dijeljenih struktura podataka.

(Slika 6) prikazuje interferenciju dva procesa.



Slika 6 Interferencija dva procesa

2.1. Jezici za komunikaciju između agenata (*eng. Agent Communication Languages)*

Kao što ljudi imaju jezike pomoću kojih komuniciraju, isto tako i agenti imaju svoje zajedničke jezike kako bi mogli razmjenjivati informacije. U ranim 1990.-ima DARPA (*eng. Defence Advanced Research Projects Agency*) je osnovala KSE (*eng. Knowledge Sharing Effort*) s ciljem razvijanja protokola za razmjenu znanja između autonomnih sustava [2]. Danas imamo veliki broj jezika namijenjenih za komunikaciju između agenata, a u nastavku će se obraditi nekoliko najpoznatijih.

2.1.1. KQML (*Knowledge Query and Manipulation Language*)

KQML je porukama orijentiran (*eng. message-based*) jezik za komunikaciju između agenata [2]. Definira zajednički format poruka. Zbog jednostavnijeg razumijevanja, koristit će se pojmovi vezani za objektno orijentirano programiranje. Kod KQML-a poruka je isto što i objekt kod objektno orijentiranog programiranja, svaka poruka je instanca neke klase (*performative*), pogledati (Tablica 3), svaka poruka ima određeni broj parametara (Tablica 2). Parametri su atribut-vrijednost parovi.

Parametar	Značenje
:content	Sadržaj poruke
:reply-with	Očekuje li pošiljatelj odgovor, i ako očekuje, identifikator za odgovor
:in-reply-to	Referenca na : reply-with parametar
:sender	Pošiljatelj poruke
:receiver	Primatelj poruke

Tablica 2 Parametri KQML poruke

Performative	Značenje
forward	Agent 1 želi da Agent 2 proslijedi poruku drugom agentu
reply	Agent očekuje odgovor
sorry	Agent ne može dati odgovor s više informacija
recommend-all	Agent 1 želi imena svih agenata koji mogu odgovoriti Agentu 2
recommend-one	Agent 1 želi ime agenta koji može odgovoriti Agentu 2

Tablica 3 Performative KQML poruke

Sada kada imamo osnovno znanje vezano za KQML poruke, navest ćemo primjer jedne takve poruke. (Kod 1) prikazuje primjer jedne jednostavne KQML poruke.

```
( ask-one
      :content ( DATE Today ?date )
      :reciever agent4
      :language LPROLOG
      :ontology NYSE-Ticks
)
```

Kod 1 Primjer KQML poruke

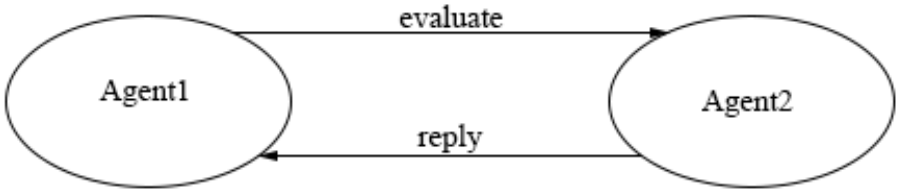
Bez posebnog poznavanja KQML jezika, samim čitanjem poruke nije teško zaključiti da pošiljatelj poruke pita agenta pod nazivom `agent4` koji je danas datum. Performativa ove poruke je `ask-one`, pomoću nje je naznačeno da agent postavlja pitanje na koje očekuje točno jedan odgovor. Poruka sadrži i atribute `:content` i `:reciever`. Atribut `:content` specificira sadržaj poruke, u ovom primjeru pitanje koji je datum danas. Atribut `:reciever` označava primatelja poruke, u ovom primjeru to je `agent4`. U poruci se još nalaze i atributi `:language` i `:ontology`. Atribut `:laguage` označava jezik kojim je iskazan sadržaj, pretpostavlja se da primatelj poruke razumije taj jezik, koji je u ovom primjeru LPROLOG. Na kraju, atribut, `:ontology` definira terminologiju korištenu u poruci, ali time se nećemo posebno baviti. U nastavku se nalazi primjer dijaloga između dva agenta (Kod 2).

```
(evaluate
      :sender Agent1
      :reciever Agent2
      :language KIF
      :ontology time
      :reply-with Q
      :content ( DATE Today ?date )
)
(reply
      :sender Agent2
      :reciever Agent1
      :language KIF
```

```
        :ontology time
        :in-reply-to Q
        :content ( Today: 03.07.2015.)
)
```

Kod 2 Primjer KQML dijaloga

U navedenom dijalogu, `Agent1` šalje upit agentu `Agent2`, nakon čega dobiva odgovor. Upit je vezan za vrijednost datuma. `Agent1` je upitu dao ime `Q`, tako da se kasnije `Agent2` može referencirati na njega kada bude odgovarao. Atribut `:ontology` ima vrijednost `time`, očito ontologija definira terminologiju vezanu za vrijeme. Odgovor koji šalje `Agent2` kaže da je danas 03.07.2015. (Slika 7) predstavlja navedeni primjer KQML dijaloga.



Slika 7 Primjer KQML dijaloga

KQML jezik je značajan za komunikaciju u višeagentskim sustavima, veliki broj višeagentskih sustava je napravljen na način da se komunikacija odvija upravo putem njega. [2] Unatoč takvom uspjehu, uslijedile su mnogobrojne kritike na KQML jezik kao što su sljedeće:

- Osnovni skup performativa KQML jezika nije strogo ograničen, stoga su razvijene različite implementacije KQML-a, pa takvi sustavi nisu mogli surađivati.
- Mehanizmi za prijenos poruka od Agenta1 do Agenta2 nikada nisu točno definirani, zbog čega je suradnja agenata koji pričaju tim jezikom dodatno otežana.
- Semantika jezika nije strogo definirana.

2.1.2. FIPA ACL (*eng. FIPA agent communication language)*

Još 1995., Foundation for Intelligent Physical Agents (FIPA) je započela razvijati standarde za agentske sustave. Naglasak je bio na razvoju jezika za komunikaciju agenata [2]. FIPA je službeno prihvaćena od strane IEEE-a 2005. godine [5]. Jezik je dosta sličan ranije spomenutom KQML jeziku. Definira 20 performativa, kao što je npr. `inform`, koje definiraju kako interpretirati poruku. Neke od njih nalaze se u (Tablica 4) .

Performative	**Značenje**
`agree`	Pošiljatelj poruke javlja da će se pobrinuti za izvršenje tražene akcije
`cancel`	Naglašava da više nema potrebe za izvršavanjem akcije iz prethodne `request` poruke.
`inform`	Pošiljatelj ove poruke šalje izjavu u koju vjeruje da će primatelj poruke povjerovati.
`failure`	Izvršavanje određene akcije nije uspjelo
`propose`	Pošiljatelj poruke šalje prijedlog primatelju.

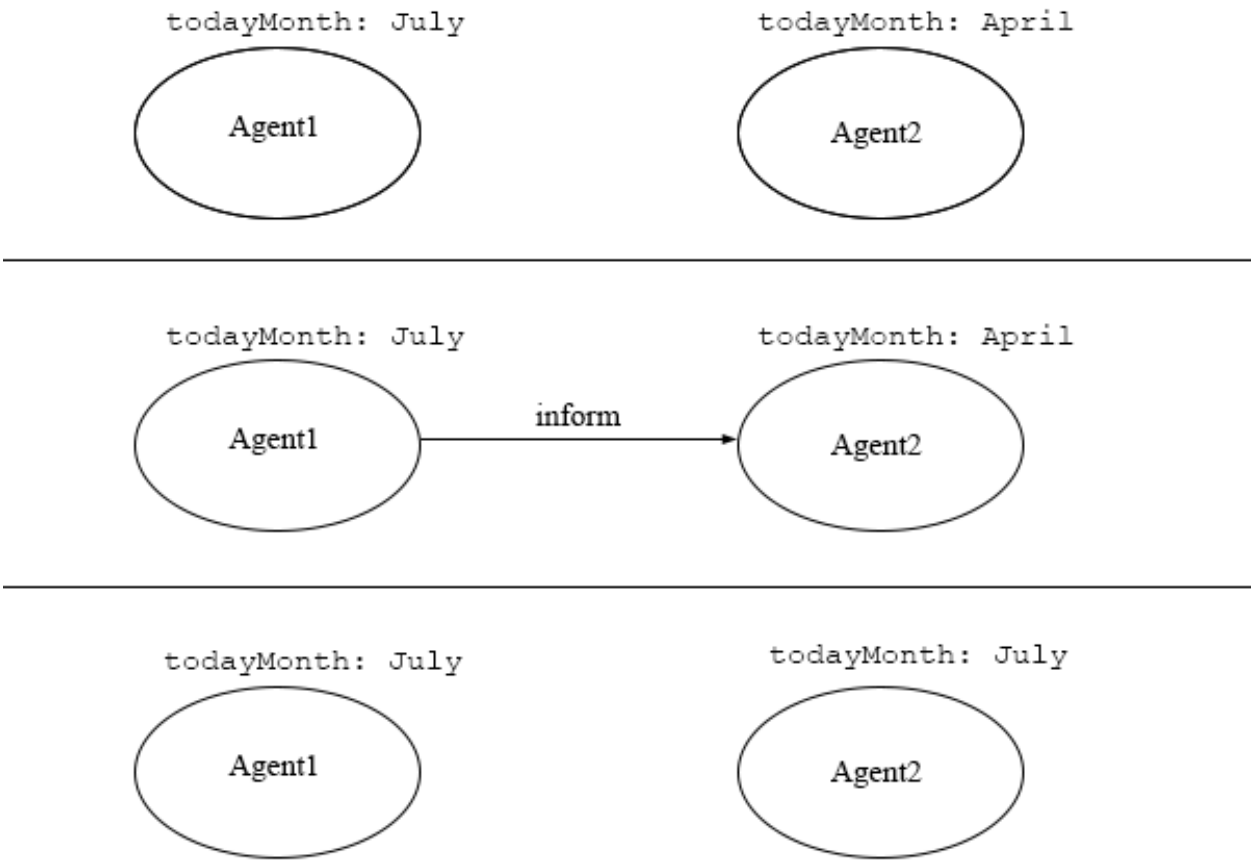
Tablica 4 FIPA performative

U nastavku se nalazi primjer FIPA poruke (Kod 3).

```
(inform
      :sender Agent1
      :receiver Agent2
      :content ( month todayMonth July )
      :language K
      :ontology time
)
```

Kod 3 Primjer FIPA poruke

(Slika 8) prikazuje komunikaciju između dva agenta putem prethodno spomenute `inform` poruke.



Slika 8 FIPA `inform` poruka

Sada je već očito da je FIPA jezik dosta sličan KQML-u , glavna razlika između ta dva jezika je skup permativa koje pružaju. Kako je već ranije spomenuto, veliki broj kritika je upućen prema KQML-u zbog nedostatka adekvatne semantike, zbog toga nije začuđujuće da su tvorci FIPA jezika dali veliku važnost stvaranju opsežne formalne semantike u njihovom jeziku.

2.1.3. Telescript

Telescript je prvi komercijalni jezik za komunikaciju između agenata koji je razvila tvrtka „General Magic“ [6]. Telescript se zapravo smatra tehnologijom a ne samo jezikom, više nije u procesu razvijanja. Kod Telescripta se sve svodi na dva osnovna koncepta: agente i mjesta (*eng.*

places). Mjesto nudi usluge (*eng. services*) agentu. Svaki agent se nalazi na određenom mjestu i može se micati s jednog mjesta na drugo. Štoviše, agenti koji se nalaze na istom mjestu mogu imati sastanke (*eng. meetings*). Za vrijeme sastanka agenti mogu komunicirati na način da pozivaju metode jedni od drugih. [6] Telescript tehnologija se sastoji od tri glavne komponente. Prvu komponentu čini objektno orijentirani programski jezik za razvijanje agentske programske podrške. Druga komponenta je stroj (*eng. engine*) za izvršavanje tih programa, a treća komponenta je skup protokola koji omogućavaju komunikaciju između agenata.

3. Višeagentski sustavi u prometu

U prošlom poglavlju su detaljno opisani višeagentski sustavi. Sljedeći korak je prikazati njihovu primjenu u području vezanom za ovaj završni rad. U nastavku će se navesti neki primjeri višeagentskih sustava u rješavanju problema nastalih u prometu.

3.1. MARLIN-ATSC

MARLIN-ATSC je skraćenica za „Multi-Agent Reinforcement Learning for Integrated Network of Adaptive Traffic Signal Controllers“ [7]. Projekt MARLIN je nastao u Torontu, Kanada 2012. godine. Voditelji projekta su Samah el-Tantawy i Baher Abdulhai. Riječ je o višeagentskom sustavu koji pomoću principa umjetne inteligencije i teorije igara nastoji smanjiti vrijeme čekanja na raskrižjima. Sustav radi u virtualnom okruženju i uspio je smanjiti vrijeme čekanja za 40%, što bi značilo da se uštedi oko 12 i pol minuta na svakih 30 križanja s semaforima kroz koja prođemo. Svaki semafor predstavlja inteligentnog agenta, opremljen je kamerama kako bi imao dobre ulazne podatke u realnom vremenu. Semafori su međusobno povezani i mogu komunicirati. Agenti se međusobno dogovaraju koju akciju će izvršiti kako ne bi akcija jednog agenta imala negativne efekte na druge agente. Agenti su sposobni učiti, a znanje svakog agenta prosljeđuje se drugima putem mreže. Za sada je sustav korišten samo u testnom okruženju, dao je dobre rezultate i postoji velika mogućnost da će kroz određeno vrijeme bit postavljen na ulice. Dobra strana projekta je što se detaljno testira i kontinuirano nadograđuje. Osobno mi se sviđa što agenti međusobno komuniciraju i nastoje ostvariti globalni cilj, što bolju protočnost u prometu, a nisu samo orijentirani na lokalni cilj koji je bolja protočnost prometa u određenom raskrižju. S druge strane, moje mišljenje je da same kamere nisu dovoljno dobar izbor senzora za stvarno okruženje jer lako

može doći do narušavanja kvalitete slike, npr. obilne padaline ili slično. To bi moglo rezultirati stjecanjem neispravnih informacija. Mislim da bi bilo dobro postaviti još nekoliko vrsta senzora kao da se npr. na cestu postave senzori koji će prepoznati prelazak automobila preko njih ili nešto slično.

3.2. Singapore ITS

Singapore ITS (*Intelligent Transport System*) je višeagentski sustav koji se koristi za prikupljanje informacije vezanih za promet. Za razliku od MARLIN-a, ovaj sustav radi u stvarnom okruženju. Informacije se prikupljaju od strane 5000 taksija [8], ugrubo rečeno , svaki taksi je jedan inteligentni agent. Taksiji šalju svoju lokaciju i brzinu kojom se kreću u kontrolnu stanicu. Kontrolna stanica zahvaljujući informacijama koje dobije od taksija može stvoriti sliku trenutnog stanja u kojem se nalazi promet, vidi kako se promet odvija i gdje dolazi do zagušenja. Unutar sustava postoji i drugi tip inteligentnog agenta, autobus. Autobus također zna svoju lokaciju i brzinu, autobusne stanice diljem Singapura imaju postavljene ekrane na kojima piše za koliko minuta otprilike dolazi koji autobus. Sustav se kontinuirano nadograđuje, tako do sada već ima implementiranu i komunikaciju s parkiralištima. Tako vozači dobivaju informacije o njima najbližim parkiralištima i broju slobodnih mjesta koja se tamo nalaze. Moje mišljenje je da bi kontrolna stanica trebala bit povezana s većim brojem vozila kako bi dobivala veći broj informacija što bi rezultiralo boljom slikom stanja u prometu. Štoviše, smatram da bi se u sustav trebala uključit i raskrižja jer se na taj način otvara novi spektar mogućnosti kontrole prometa kao što su npr. preusmjeravanja prometa u ulice veće protočnosti.

3.3. Usporedba MARLIN-a i Singaproe ITS-a

Za potrebe ovog završnog rada pogledao sam veliki broj radova i projekata vezanih za temu koju obrađujem. Ranije spomenuta dva višeagentska sustava su mi dala određene ideje za izradu vlastitog projekta te ih toga smatram bitnima. Prva velika razlika između sustava je ta što MARLIN radi u virtualnom okruženju dok je Singapore ITS smješten u stvarno okruženje, tj. singapurski promet. Kod MARLIN-a mi se sviđa to što postoji komunikacija između križanja i vozila te križanja i križanja. Dok kod Singapore ITS-a komunikaciju između kontrolne stanice i vozila smatram velikom prednosti. Na temelju toga, odlučio sam napraviti višeagentski sustav u

virtualnom okruženju koji će imat komunikaciju između križanja i vozila, križanja i križanja te vozila i kontrolne stanice.

4. Realizacija projekta

Današnja vozila su opremljena različitim senzorima i računalima jake procesorske moći. Određeni broj automobila su polu autonomna vozila ili vozila koja u potpunosti upravljaju samima sobom, može se reći da takva vozila imaju određenu inteligenciju. Nažalost, ceste po kojima se takva vozila kreću uglavnom ne posjeduju inteligenciju.

4.2. Definiranje zahtijeva

Cilj je napraviti višeagentski sustav koji predstavlja simulaciju prometa. U sustavu se nalazi nekoliko tipova inteligentnih agenata: vozila, raskrižja, kontrolna stanica i prometna signalizacija. Program treba imati dobro korisničko sučelje kako bi korisnik jasno vidio što se događa u simulaciji. Korisnik također treba imati mogućnost unosa određenih parametara vezanih za simulaciju kako bi se mogao simulirati promet u različitim stanjima. Za vrijeme trajanja simulacije, korisnik je u mogućnosti komunicirati s raskrižjima, može ih onemogućiti (simulirati radove na cesti) i ponovno ih vratiti u funkcionalno stanje. Ukoliko je neko raskrižje onemogućeno, ostala raskrižja trebaju javiti vozilima da izbjegavaju putanju do cilja koja prolazi tim raskrižjem. Raskrižja moraju bit svjesna stanja u prometu. Kada se vozilo nalazi na raskrižju, može poslati zahtjev raskrižju za izračun najkraćeg puta do njegovog cilja, izračun se svaki puta iznova vrši kada vozilo dođe do novog raskrižja. Štoviše, korisnik mora imati uvid u instrukcije koje je raskrižje uputilo vozilima. Korisnik mora imati mogućnost pauziranja simulacije kako bi se vidjelo trenutno stanje u prometu. Potrebno je omogućiti komunikaciju između vozila i raskrižja, raskrižja i raskrižja, raskrižja i kontrolne stanice te raskrižja i signalizacije Program mora imati mogućnost spremanja rezultata simulacije u obliku datoteke i u obliku slike dijagrama.. Naglasak cijelog projekta je na implementaciji.

4.3. Odabir tehnologija i alata

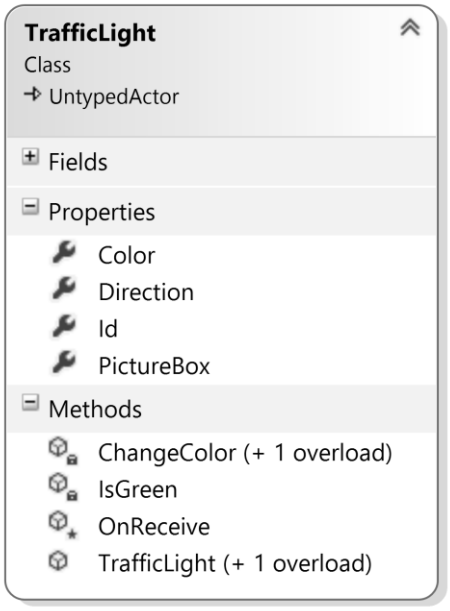
Program je napisan u objektno orijentiranom jeziku C#. Jezik je odabran zbog toga što omogućava da se agenti realiziraju kao prošireni objekti. Štoviše, podržava veliki broj biblioteka i alata koji olakšavaju izgradnju simulacije. Uz sve navedene prednosti, kada sam uzeo u obzir da već nekoliko godina aktivno koristim C#, izbor programskog jezika nije bio težak. Implementacija komunikacije između agenata je bitan segment simulacije i zanimljiv je izazov. Za komunikaciju između agenata odabran je Akka.NET, alat za izradu aplikacija vođenih događajima [9]. Akka.NET je izabran zbog svoje stabilnosti i velikog broja aplikacija koje ga uspješno koriste. Nadalje, omogućava mi da elegantno ostvarim komunikaciju između agenata te svoj fokus usmjerim ka implementiranju ostalih dijelova aplikacije. Danas je već opće poznato da kompleksna programska podrška mora imati dobro ostvarenu komunikaciju između svojih segmenata, u mom slučaju agenata.

4.4. Implementacija

U nastavku slijedi opis implementacije programa s naglaskom na inteligentne agente.

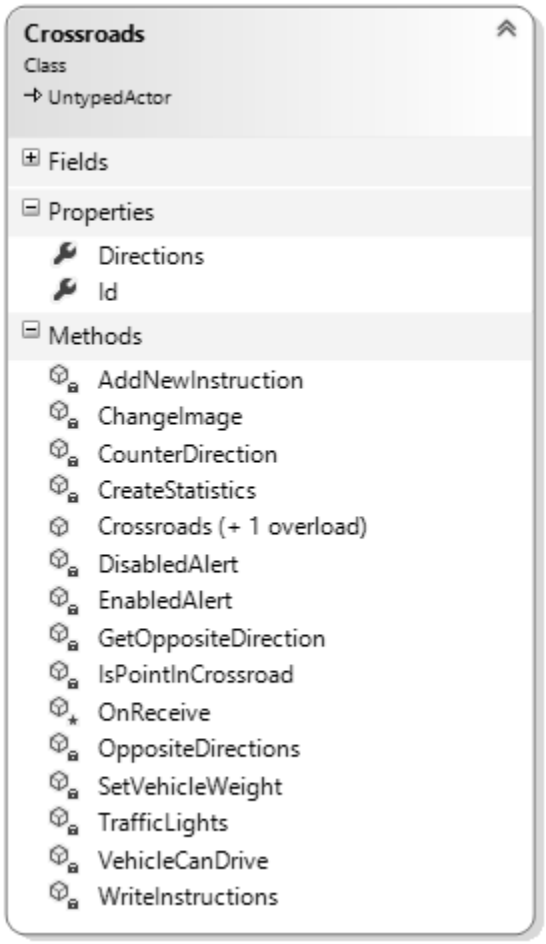
4.4.1. Inteligentni Agenti

Kako je naglasak ovog završnog rada na inteligentnim agentima, u aplikaciji se nalazi 5 različitih inteligentnih agenata. Prvi inteligentni agent predstavlja prometnu signalizaciju, preciznije semafor. U aplikaciji je prikazan klasom pod nazivom „TrafficLight“. Implementirane varijable, metode i klase su imenovane engleskim jezikom. (Slika 9) prikazuje semafor ili „TraficLight“ klasu. Semafor ima mogućnost promjene boje, crvena ili zelena i odgovoriti na upit o tome koja je trenutno boja prikazana na njemu. Direktno komunicira s kontrolnom stanicom, a putem nje može slati poruke do raskrižja.



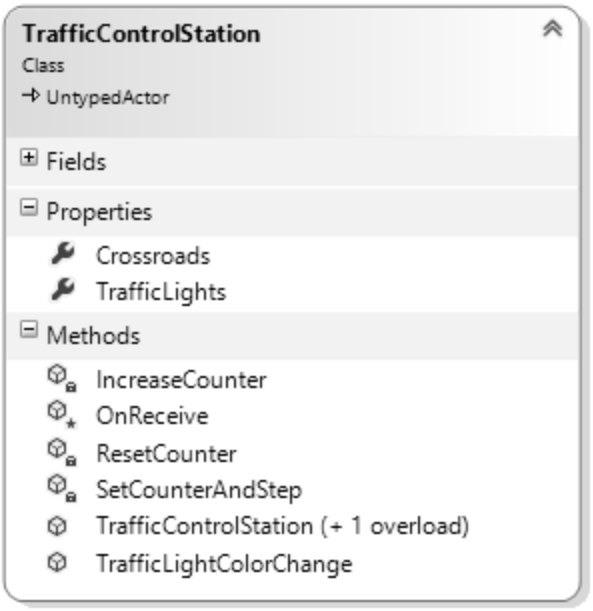
Slika 9 Klasa „TrafficLight“

Svaki semafor se nalazi na nekom raskrižju. Prema tome, u aplikaciji se nalazi i raskrižje kao inteligentni agent. Predstavljeno je klasom pod nazivom „Crorrsrads“ (Slika 10). Raskrižje može direktno komunicirati s kontrolnom stanicom. A putem kontrolne stanice komunicira s drugim raskrižjima , semaforima, vozilima ili samo jednim vozilom. Raskrižje ima mogućnost izdavanja uputa vozilu u kojem bi smjeru bilo najbolje da ide. Svako raskrižje zna koje upute je dalo određenom vozilu, korisnik im može pristupit odgovarajućim upitom. Raskrižje je svjesno vozila koji se na njemu nalaze i zna smjer iz kojega dolaze. Također raskrižje je svjesno i smjerova u koje može uputiti vozila koja mu pošalju upit za istim. Raskrižje može biti u onemogućeno, što simulira radove na cesti, tako da je promet u tom raskrižju otežan, zbog toga ostala raskrižja daju upute vozilima da izbjegavaju raskrižje toga tipa. Dakle, očita je adaptivnost raskrižja, prilagođavaju se trenutnom stanju u prometu i u skladu s tim donose odluke.



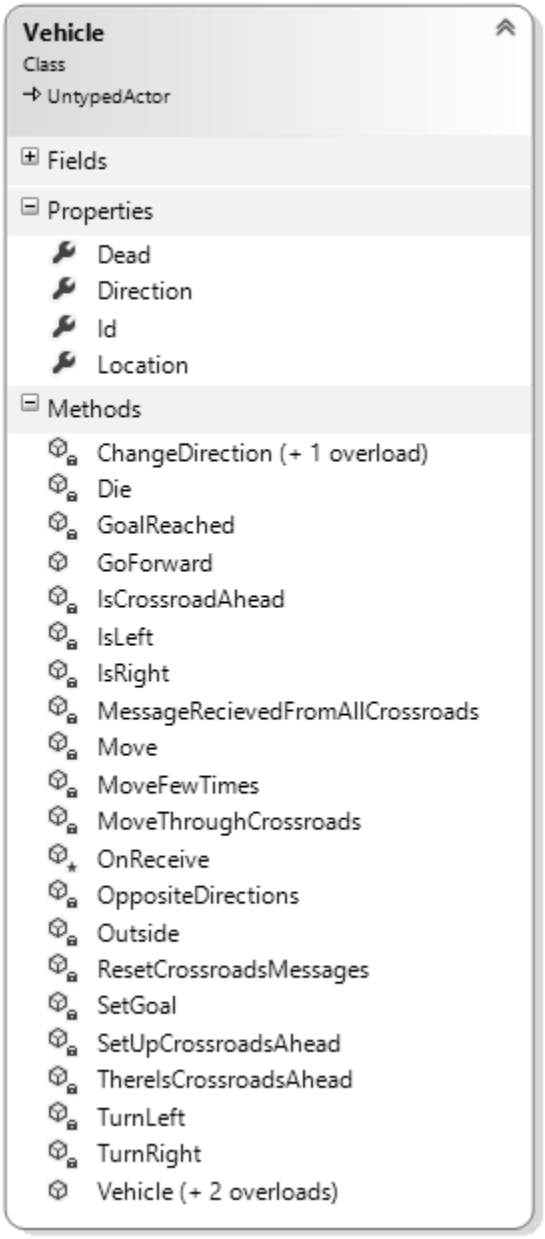
Slika 10 Klasa „Crossroads“

Kako je već spomenuto, komunikaciju između raskrižja i semafora omogućava kontrolna stanica. Ona je u programu predstavljena klasom pod nazivom „TrafficControlStation“. Posjeduje listu semafora i raskrižja s kojima je u izravnoj komunikaciji. Izravno komunicira i s inteligentnim pod nazivom „Vehicles“ kada on to zatraži, ali više o tome ćemo nešto poslije. Kontrolna stanica ima iznimnu važnost u ovoj simulaciji. Ona omogućava korisniku interakciju s simulacijom, vozilima daje potrebne informacije i usklađuje procese koji se izvršavaju unutar simulacije. Glavna uloga ovog inteligentnog agenta je da koordinira procesima kako u simuliranom prometu tako i u cijeloj aplikaciji. U nastavku, (Slika 11) predstavlja klasu „TrafficControlStation“.



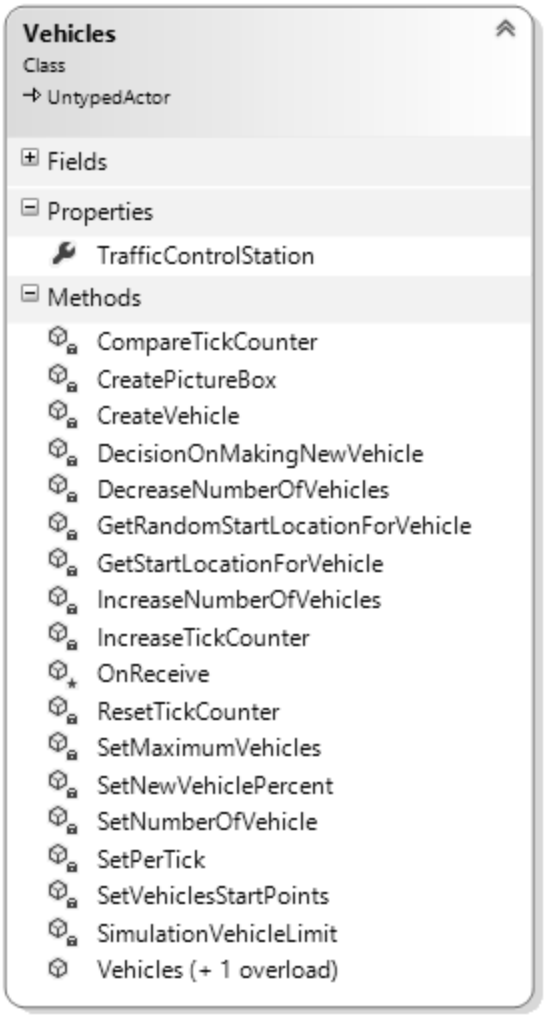
Slika 11 Klasa „TrafficControlStation“

Prometna infrastruktura je riješena, sada nam trebaju još samo vozila. Svako vozilo predstavljeno je klasom pod nazivom „Vehicle“. Inteligentni agent tipa vozilo ima mogućnost kretanja po prometu. Autonomno izvršava svoje akcije vezane za kretanje kao što su „idi ravno“ li „skreni“. Svako vozilo ima svoj cilj, dolaskom na cilj nestaje iz simulacije. Izravno komunicira s inteligentnim agentom „Vehicles“ a putem njega svoje poruke prima i šalje do ostalih agenata. Vozila se unutar simulacije stvaraju na različitim lokacijama, mogu biti slučajno odabrane ili unaprijed određene. Vozila su svjesna svoje lokacije i mogu je usporediti s svojim ciljem. U simulaciji se kreću poštujući prometnu signalizaciju. Kako je naglasak ovog završnog rada na implementaciji inteligentnih agenata i izgradnja događajem vođenog programa, kada dođu do raskrižja, ostvare komunikaciju s raskrižjem i razmjenom poruka dobiju informaciju što prometna signalizacija pokazuje, odnosno pravilo koje vrijedi za samo vozilo. Kada dobiju informaciju da mogu proći kroz raskrižje, komunikacija se nastavlja s drugim ciljem. Treba odrediti smjer kojim će se vozilo kretati. Vozilo šalje upit raskrižju s ciljem dobivanja smjera za nastavak kretanja. Raskrižje ovisno kako su parametri simulacije postavljeni, ili računa najkraći put vozila do cilja ili slučajno odabire smjer, smjer koji je odabran prosljeđuje vozilu i ono se nastavlja kretati.



Slika 12 Klasa „Vehicle“

U programu postoji i klasa pod nazivom „Vehicles“ koja predstavlja inteligentnog agenta zaduženog za komunikaciju vozila, agenta tipa „Vehicle“ s ostalima. Osim toga, zadaća ovog inteligentnog agenta je da stvara nova vozila koja se uključuju u promet. Ukratko, unutar simulacije zadužen je za komunikaciju i stvaranje novih vozila te svih procesa vezanih za isto. (Slika 13) predstavlja klasu „Vehicles“.



Slika 13 Klasa „Vehicles“

Tip inteligentnog agenta	**Broj inteligentnih agenata u simulaciji**
TrafficLight	više
Crossroads	više
TrafficControlStation	1
Vehicle	više
Vehicles	1

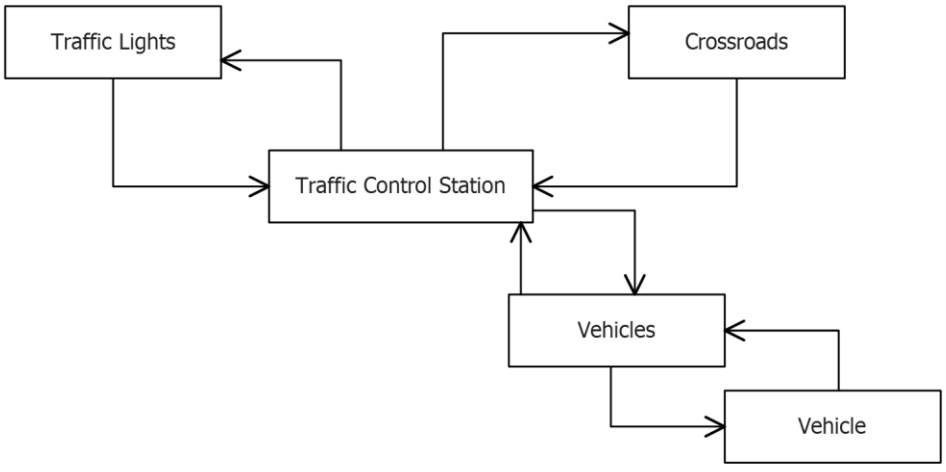
Tablica 5 Broj inteligentnih agenata u simulaciji

4.4.2. Okolina

Kako imamo inteligentne agente, treba ih staviti u okruženje. Riječ je o simulaciji pa su agenti smješteni u virtualnu okolinu. Okolinu predstavljaju ceste, križanja, prometna signalizacija itd. Kako su sva raskrižja na početku simulacije dostupna, a postoji mogućnost da korisnik onemogući odvijanje prometa na određenom križanju, riječ je o dinamičkoj okolini. Iako na prvi pogled izgleda da je riječ o diskretnom okruženju, okruženje je kontinuirano što se vidi po ponašanju inteligentnih agenata tipa „Vehicle“. Okolina je također pristupačna, sve informacije su dostupne i potpune, ali i epizodna što znači da ako je neko raskrižje prije bilo onemogućeno pa se promet unutar njega nije odvijao, a sada je omogućeno, stanje iz njegove prošlosti ne utječe na nove izračune najkraćeg puta za vozila.

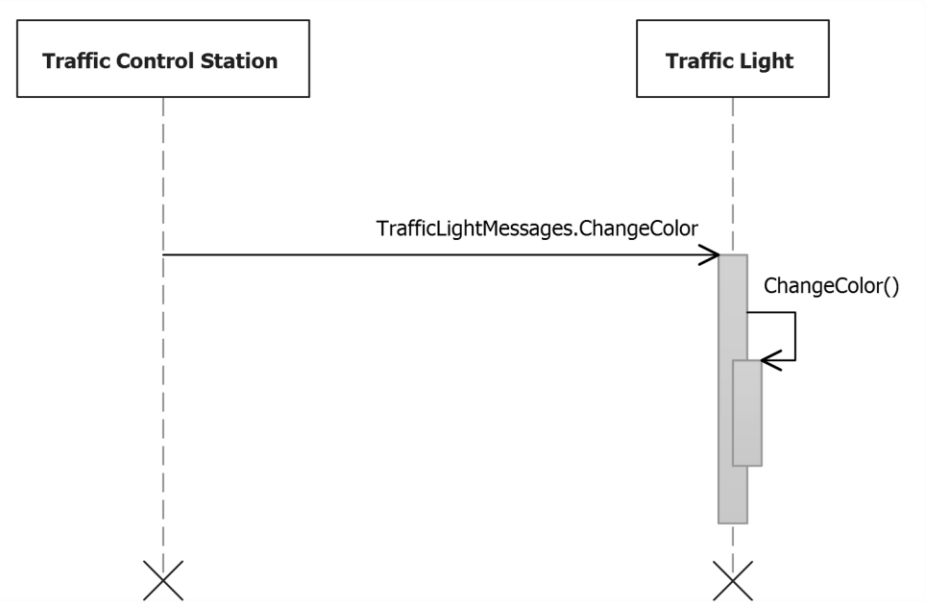
4.4.3. Komunikacija inteligentnih agenata

Do sada su objašnjeni inteligentni agenti koji se nalaze u programu i okolina u kojoj se nalaze. Vrijeme je da se prikaže komunikacija u navedenom višeagentskom sustavu. Već je ranije spomenuto kako poruke između agenata putuju od jednog do drugog. Svaki agent može komunicirati s bilo kojim drugim agentom u simulaciji, a komunikacija se odvija po shemi (Slika 14).



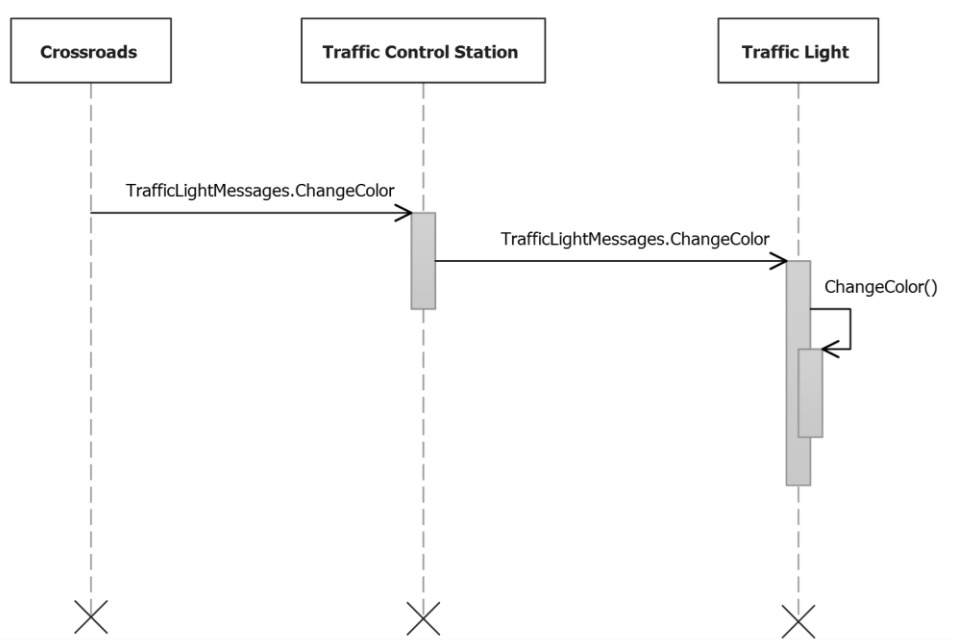
Slika 14 Komunikacija inteligentnih agenata

Navest ćemo nekoliko primjera radi jednostavnijeg shvaćanja komunikacije između agenata. (Slika 15) prikazuje komunikaciju kontrolne stanice i semafora. Kontrolna stanica javlja semaforu da promijeni boju, nakon čega semafor izvršava traženu akciju.



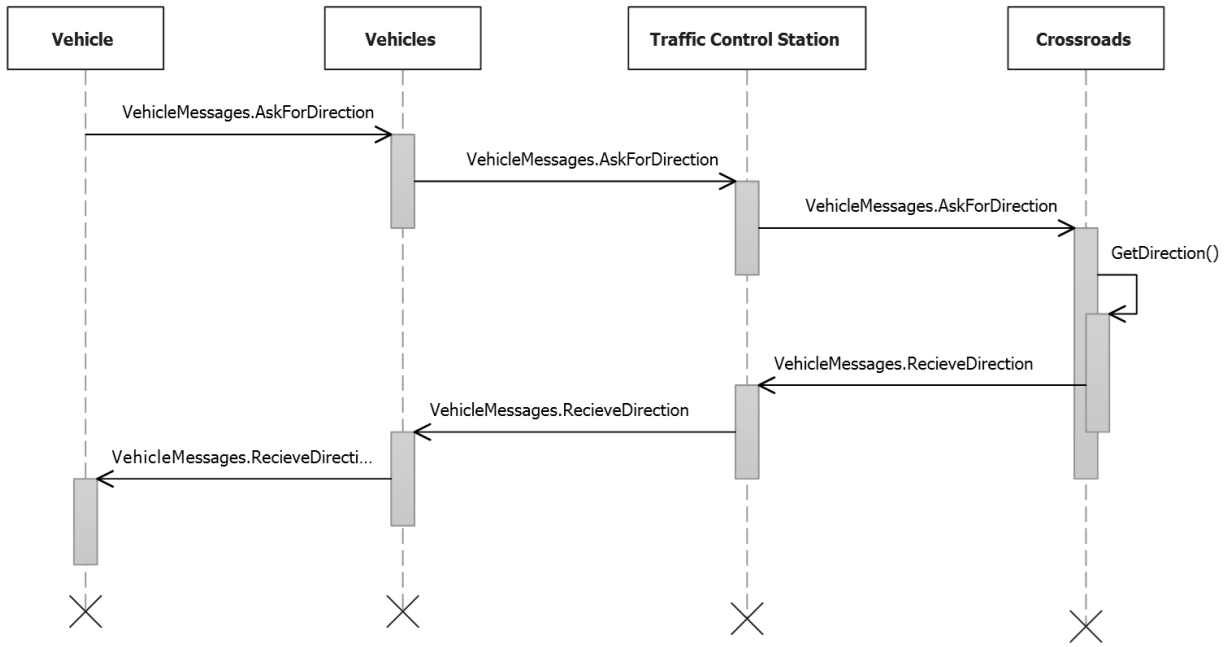
Slika 15 Komunikacija kontrolne stanice i semafora

Kako se semafor nalazi na križanju, a u simulaciji su raskrižje i semafor inteligentni agenti. Raskrižje ima mogućnost reći semaforu da promijeni boju. Takva komunikacija se odvija preko kontrolne stanice (Slika 16).



Slika 16 Komunikacija križanja i semafora

Komunikacije između vozila i raskrižja je nešto kompliciranija (Slika 17).

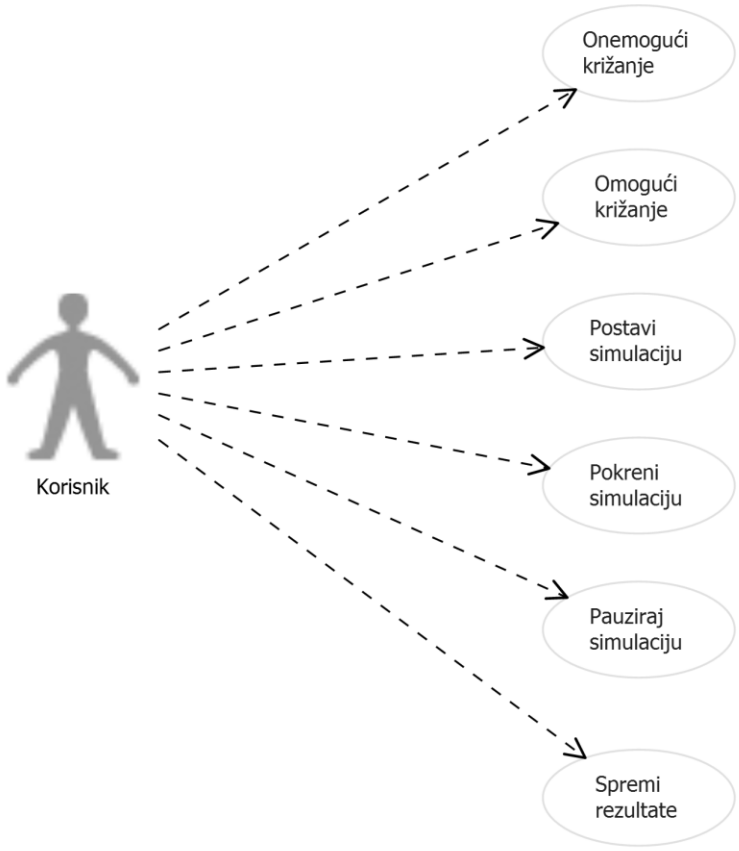


Slika 17 Komunikacija vozila i raskrižja

Vozilo prvo svoj upit pošalje agentu „Vehicles“, on ga proslijedi do kontrolne stanice koja ga konačno proslijedi do raskrižja. Kada raskrižje dobije upit, računa najkraći put, podatak stavlja u poruku i šalje prema vozilu na način da poruku pošalje prvo kontrolnoj stanici, ona ga proslijedi do agenta „Vehicles“ koji poruku dostavlja vozilu. Možda bi bilo bolje da agenti izravno komuniciraju jedan s drugim, ali za potrebe ovog završnog komunikacija ostvarena na navedeni način.

4.4.4. Uloga korisnika

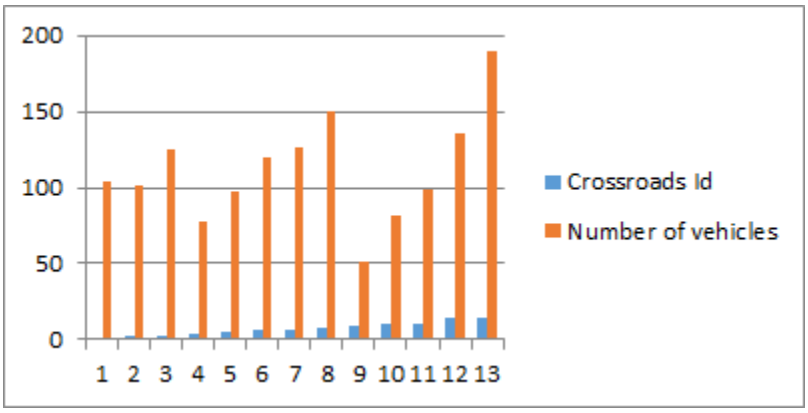
Korisnik ima mogućnost postaviti parametre simulacije kako bi dobio simulirani promet kakav želi. Nakon što postavi i pokrene simulaciju, može sudjelovati u njoj kao što je npr. opcija omogućavanja ili onemogućavanja protoka prometa na određenom raskrižju. Također može tražiti grafički prikaz toka prometa ili uputa koje određeno raskrižje poslalo vozilima. Na kraju, može spremiti dobivene rezultate.



Slika 18 Uloga korisnika u simulaciji

4.4.5. Rezultati simulacije

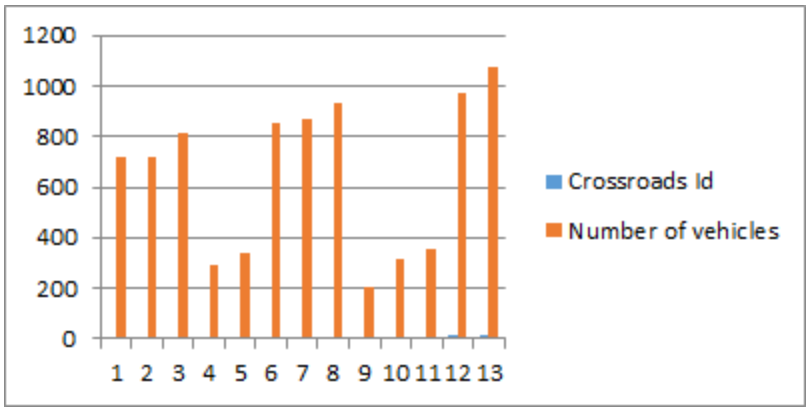
Rezultati se spremaju u obliku excel i bmp datoteke kako bi se mogli koristiti kasnije. Također rezultati se mogu zatražiti i prikazati za vrijeme simulacije, u tom slučaju, u prozoru simulacije se iscrtava traženi dijagram. U nastavku slijedi nekoliko primjera rezultata simulacije.



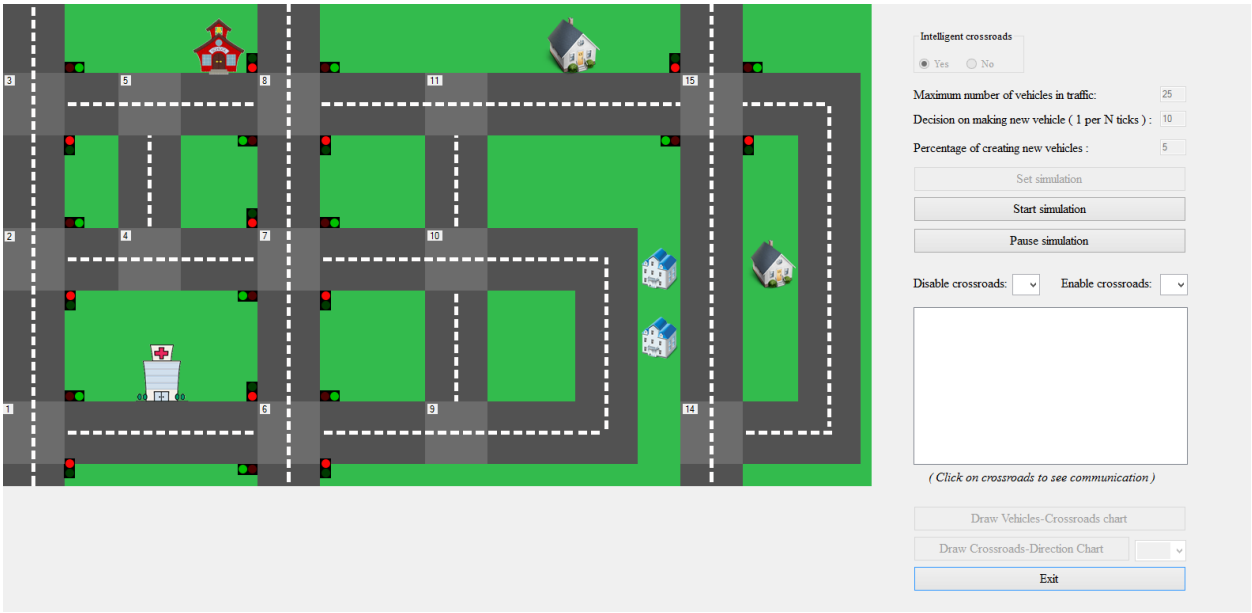
Slika 19 Rezultat simulacije 1



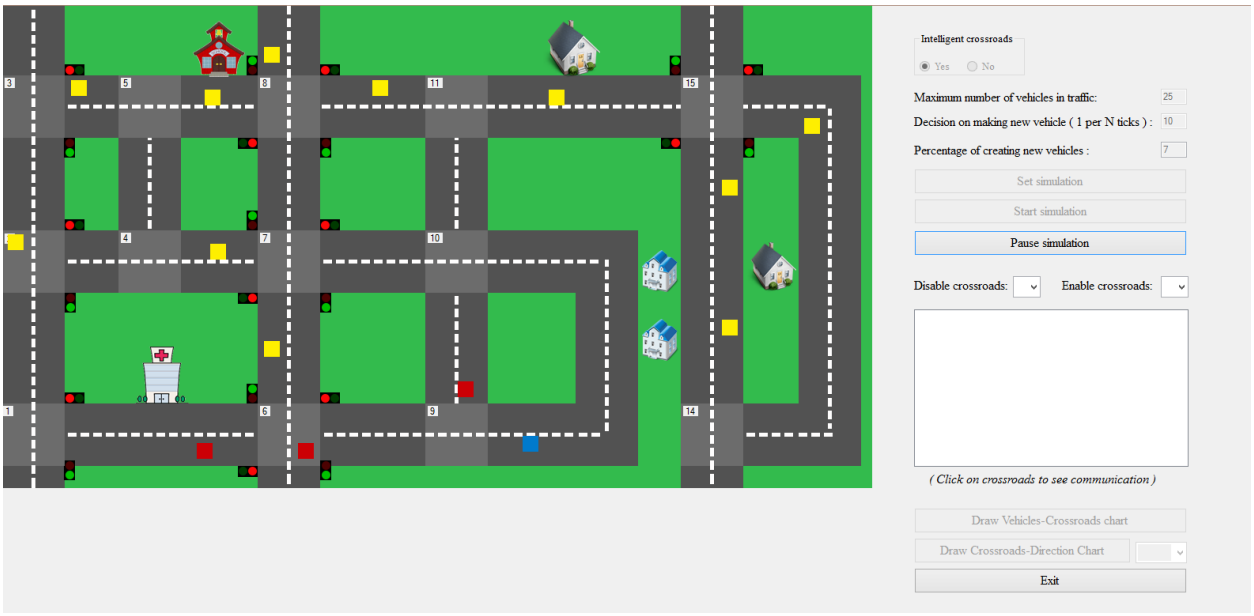
Slika 20 Rezultat simulacije 2



Slika 21 Rezultat simulacije 3



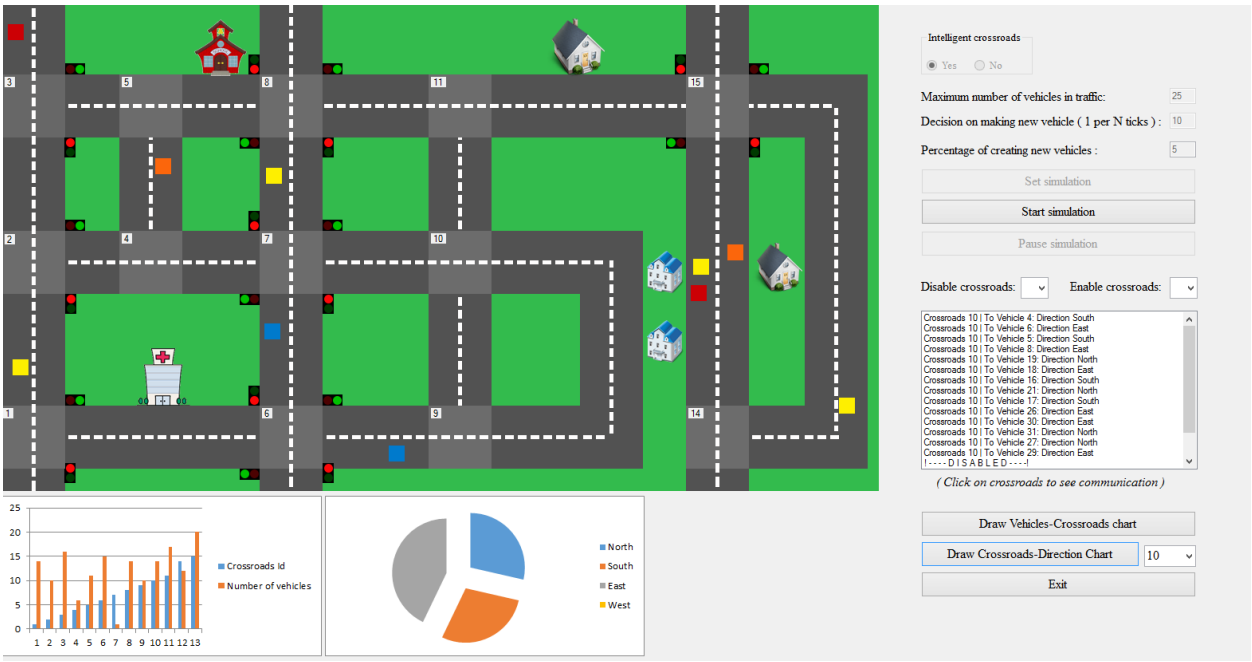
Slika 22 Slika iz simulacije 1



Slika 23 Slika iz simulacije 2



Slika 24 Slika iz simulacije 3



Slika 25 Slika iz simulacije 4

Zaključak

Inteligentni agenti su donijeli veliki napredak u rješavanju problema iz svakodnevnog života. Uspješno se koriste u mnogobrojnim aplikacijama različitih namjena. Kada uzmemo u obzir da su nova vozila opremljena velikim brojem kvalitetnih senzora i računalima, logičan korak je uvođenje inteligentnih agenata u promet. Ali nisu samo vozila ta koja mogu bit inteligentna, nema razloga da to ne budu i ostali elementi prometa kao što su raskrižja. Simulacija napravljena za potrebe ovog završnog rada pružila je mogućnost raskrižjima da budu inteligentni u virtualnom okruženju. Pokazalo se da je to dobra praksa. Raskrižja međusobno komuniciraju i svjesni su stanja u prometu ne samo lokalno u njihovoj blizini, nego imaju sliku odvijanja cijelog prometa. Tako npr. u simulaciji ukoliko dođe do rada na cesti, raskrižja preusmjeravaju vozila na ceste gdje je veća protočnost prometa. Simulacija ima mogućnost nadogradnje i nije idealna. Tako se npr. može implementirati komunikacija između vozila, u tom slučaju bi se mogli spriječiti potencijalni sudari. Također bi se mogla napraviti i bolja raskrižja, npr. ona s mogućnošću učenja. Inteligentni agenti su nam otvorili širok spektar novih mogućnosti, na nama je da ih iskoristimo. To nisu samo

mogućnosti za rješavanje problema u prometu. Veliki korak je napravljen i u svijetu informatike, pružaju novi pogledat na razvoj programske podrške.

Slike

Slika 1 Odnos inteligentnog agenta i okolice 8

Slika 2 Jednostavan reaktivan agent 9

Slika 3 Cilju orijentirani agent 10

Slika 4 Koristi orijentiran agent 11

Slika 5 Komunikacija dva agenta 12

Slika 6 Interferencija dva procesa 13

Slika 7 Primjer KQML dijaloga 16

Slika 8 FIPA `inform` poruka 18

Slika 9 Klasa „TrafficLight“ 22

Slika 10 Klasa „Crossroads“ 23

Slika 11 Klasa „TrafficControlStation“ 24

Slika 12 Klasa „Vehicle“ 25

Slika 13 Klasa „Vehicles“ 26

Slika 14 Komunikacija inteligentnih agenata 27

Slika 15 Komunikacija kontrolne stanice i semafora 28

Slika 16 Komunikacija križanja i semafora 29

Slika 17 Komunikacija vozila i raskrižja 29

Slika 18 Uloga korisnika u simulaciji 30

Slika 19 Rezultat simulacije 1 31

Slika 20 Rezultat simulacije 2 31

Slika 21 Rezultat simulacije 3 32

Slika 22 Slika iz simulacije 1 32
Slika 23 Slika iz simulacije 2 33
Slika 24 Slika iz simulacije 3 33
Slika 25 Slika iz simulacije 4 34

Tablice

Tablica 1 Inteligentni agenti - PACO (eng. PAGE) 7
Tablica 2 Parametri KQML poruke 14
Tablica 3 Performative KQML poruke 14
Tablica 4 FIPA performative 17
Tablica 5 Broj inteligentnih agenata u simulaciji 26

Kod

Kod 1 Primjer KQML poruke 15
Kod 2 Primjer KQML dijaloga 16
Kod 3 Primjer FIPA poruke 17

Literatura

[1] »Wikipedia,« [Mrežno]. Available: https://hr.wikipedia.org/wiki/Inteligentni_agenti.

[2] M. Wooldridge, An Introduction to MultiAgent Systems, Liverpool, 2009.

[3] J. M. Pujol, Structure in Artificial Societies, Barcelona, 2006.

[4] Y. Moses i M. Tennenholtz, »Artificial Social Systems,« 1995.

[5] »Foundation for Intelligent Physical Agents,« [Mrežno]. Available: http://www.fipa.org/.

[6] H. Helin, Supporting Nomadic Agent-based Applications in the FIPA Agent Arhitecture, Helsinki, 2003.

[7] S. el-Tantawy i B. Abdulhai, »IEEEE,« 2012. [Mrežno]. Available: http://ieeexplore.ieee.org.

[8] S. Ezell, Intelligent Transportation Systems, 2010.

[9] »Akka.NET,« [Mrežno]. Available: http://getakka.net/.

Sažetak

Inteligentnih agenti su programska podrška koja ima sposobnost da samostalno i bez intervencije korisnika izvršava postavljeni zadatak, u tom slučaju, krajnji korisnik se izvještava o završetku zadatka ili samoj pojavi očekivanog događaja. Danas se za rješavanje kompleksnih problema koriste višeagentski sustavi.

Summary

Intelligent agents are software which is capable of executing task independently without user intervention, in this case, user will be informed about task when it is completed or when expected event appears. Today we use multi-agent system to solve complex problems.

Skraćenice

FIPA ACL, MARLIN-ATSC, ITS