

Model poučavanja agentske paradigme korištenjem simulacijskih okruženja

Rakić, Krešimir

Doctoral thesis / Disertacija

2020

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:166:917546>

Rights / Prava: [Attribution-NonCommercial-ShareAlike 4.0 International](#)/[Imenovanje-Nekomercijalno-Dijeli pod istim uvjetima 4.0 međunarodna](#)

Download date / Datum preuzimanja: **2025-01-01**

Repository / Repozitorij:

[Repository of Faculty of Science](#)





PRIRODOSLOVNO-MATEMATIČKI FAKULTET

Krešimir Rakić

**MODEL POUČAVANJA AGENTSKE
PARADIGME KORIŠTENJEM
SIMULACIJSKIH OKRUŽENJA**

DOKTORSKI RAD

Split, svibanj 2020.



PRIRODOSLOVNO-MATEMATIČKI FAKULTET

Krešimir Rakić

**MODEL POUČAVANJA AGENTSKE
PARADIGME KORIŠTENJEM
SIMULACIJSKIH OKRUŽENJA**

DOKTORSKI RAD

MENTOR: prof. dr. sc. Marko Rosić

Split, svibanj 2020.



FACULTY OF SCIENCE

Krešimir Rakić

**MODEL OF TEACHING
AGENT PARADIGM USING
SIMULATION ENVIRONMENTS**

DOCTORAL THESIS

SUPERVISOR: professor Marko Rosić, Ph. D.

Split, May 2020.

ZAHVALE

Prije svih, iskreno se zahvaljujem svom mentoru, prof. dr .sc. Marku Rosiću na ukazanom povjerenju i podršci tijekom izrade ovog rada.

Veliko hvala izv. prof. dr. sc. Ivici Boljatu na savjetima i pomoći tijekom cijelog studija.

Hvala članovima povjerenstva: izv. prof. dr. sc. Ivici Boljatu, izv. prof. dr. sc. Tonću Marušiću, izv. prof. dr. sc. Ani Grubišić, prof. dr. sc. Mirjani Bonković i izv. prof. dr. sc. Saši Mladenoviću na svim komentarima, vremenu i trudu potrošenom na vrednovanju ovog rada.

Veliko hvala mojim dragoj kolegici i kolegama: dr. sc. Divni Krpan, doc. dr. sc. Tomislavu Volariću, Danijelu Vasiću, prof. i Danijelu Zeleniki, mag. ing. comp. koji su pomogli te rado sudjelovali u istraživanjima. Zahvaljujem i svim studentima koji su sudjelovali u istraživanjima.

TEMELJNA DOKUMENTACIJSKA KARTICA

Sveučilište u Splitu

Doktorska disertacija

Prirodoslovno-matematički fakultet

Poslijediplomski sveučilišni studij

"Istraživanje u edukaciji u području prirodnih i tehničkih znanosti"

MODEL POUČAVANJA AGENTSKE PARADIGME KORIŠTENJEM SIMULACIJSKIH OKRUŽENJA

Krešimir Rakić

Sveučilište u Mostaru, Fakultet strojarstva, računarstva i elektrotehnike

Matice hrvatske b.b., 88000 Mostar, Bosna i Hercegovina

Sažetak

Agentska paradigma objedinjuje istraživanja u kognitivnoj znanosti i umjetnoj inteligenciji. Uključivanje interaktivnih tehnologija, koje omogućavaju učenicima svih uzrasta aktivno sudjelovanje u relevantnim i autentičnim nastavnim aktivnostima, značajno doprinosi kvaliteti nastavnog procesa i usvajanju nastavnih sadržaja. Vizualizacija nastavnih sadržaja pomaže boljem razumijevanju i usvajanju gradiva, posebno u situacijama kad je prisutna visoka razina apstrakcije sadržaja. Ne postoje značajnije razlike u postignućima učenika koji su bili u kontaktu s fizičkim tehnološkim uređajima i učenika koji su kao nastavno pomagalo koristili simulacijsko okruženje. Prema navedenom postavljena su dva osnovna cilja istraživanja: a) utvrditi utjecaj primjene postojećih simulacijskih okruženja na usvajanje i razumijevanje koncepata agentske paradigme kod studenata u visokom obrazovanju, b) oblikovati model poučavanja agentske paradigme koji će sadržavati karakteristične primjere i zadatke iz ovog područja prikazane korištenjem simulacijskih okruženja. Prema ciljevima osmišljeno je istraživanje u tri faze koje se provodilo tijekom tri akademske godine na visokoškolskim ustanovama. Rezultati cjelokupnog istraživanja pokazali su kako studenti postižu bolje rezultate kada se za poučavanje agentske paradigme koristi simulacijsko okruženje. Rezultati istraživanja mogu biti smjernice za razvoj novih načina poučavanja agentske paradigme na visokoškolskoj razini.

(206 stranica, 59 slika, 32 tablice, 234 literaturnih navoda, jezik izvornika: hrvatski). Rad je pohranjen u Sveučilišnoj knjižnici u Splitu, Ruđera Boškovića 31, Split te u Nacionalnoj i sveučilišnoj knjižnici, Ul. Hrvatske bratske zajednice 4, Zagreb.

Ključne riječi: model poučavanja, agentska paradigma, simulacijska okruženja, informatika, umjetna inteligencija

Mentor: prof. dr. sc. Marko Rosić

Ocjenjivači:

1. izv. prof. dr. sc. Ivica Boljat
2. izv. prof. dr. sc. Tončo Marušić
3. izv. prof. dr. sc. Ani Grubišić
4. prof. dr. sc. Mirjana Bonković
5. izv. prof. dr. sc. Saša Mladenović

Rad prihvaćen: 20. svibnja 2020.

BASIC DOCUMENTATION CARD

University of Split
Faculty of Science
Doctoral Programme
"Education Research in Natural and Technical Sciences"

Doctoral Thesis

MODEL OF TEACHING AGENT PARADIGM USING SIMULATION ENVIRONMENTS

Krešimir Rakić

University of Mostar, Faculty of Mechanical Engineering, Computing and Electrical Engineering
Matice hrvatske b.b., 88000 Mostar, Bosna i Hercegovina

Abstract

The agent paradigm integrates research in cognitive science and artificial intelligence. The inclusion of interactive technologies, which enable students of all ages to participate actively in relevant and authentic teaching activities, significantly contributes to the quality of the teaching process and the acquisition of teaching content. Visualization of teaching content helps to better understand and adopt the material, especially with highly abstract content. There are no significant differences in the achievement of students who have been in contact with physical technological devices and students using the simulation environment as a teaching tool. According to the above, two basic research goals are set: a) to determine the impact of the application of existing simulation environments on the higher education students' adoption and understanding of the agent paradigm concepts, b) to model an agent paradigm teaching model that will contain characteristic examples and tasks in this field, presented using simulation environment. According to the goals, a three-phase research was designed and conducted over three academic years at higher education institutions. The results of the overall research showed that students achieve better results when a simulation environment is used to teach the agent paradigm. The results of the research may be guidelines for the development of new ways of teaching the agent paradigm at a higher education level.

(206 pages, 59 figures, 32 tables, 234 references, original in Croatian) Thesis is deposited in the University Library of Split, Ruđera Boškovića 31, Split and National and University Library, Ul. Hrvatske bratske zajednice 4, Zagreb

Keywords: model of teaching, agent paradigm, simulation environment, informatics, artificial intelligence

Supervisor: Marko Rosić, Ph. D.

Reviewers:

1. Ivica Boljat, Ph. D.
2. Tončo Marušić, Ph. D.
3. Anđelko Grubišić, Ph. D.
4. Mirjana Bonković, Ph. D.
5. Saša Mladenović, Ph. D.

Thesis accepted: May, 20th 2020.

SADRŽAJ:

1. UVOD	1
2. DOSADAŠNJE SPOZNAJE	6
2.1. Znanje i teorije učenja	6
2.1.1. Definicija znanja	6
2.1.2. Teorije učenja	7
2.1.3. Konstruktivistički pristup učenju i poučavanju.....	8
2.2. Agentska paradigma	15
2.2.1. Definicija i osnovna svojstva agenata	16
2.2.2. Matematički model funkcionalnosti agenta	19
2.2.3. Klasifikacija agenata	20
2.2.4. Klasifikacija okruženja.....	30
2.2.5. Višeagentski sustavi	34
2.2.6. Primjena agentskih i višeagentskih sustava	40
2.3. Simulacija.....	43
2.3.1. Definicija i osnovna svojstva simulacije.....	43
2.3.2. Kada primijeniti simulaciju?	45
2.3.3. Područja primijene simulacije.....	48
2.3.4. Primjena simulacije u obrazovanju	49
2.3.5. Alati za modeliranje i simuliranje zasnovani na agentima.....	52
2.4. Modeli poučavanja	63
3. METODOLOGIJA ISTRAŽIVANJA	68
3.1. Predmet i cilj istraživanja	68
3.2. Istraživačka pitanja	68
3.3. Paradigma i metodologijski pristup.....	69
3.4. Nacrt istraživanja	70
3.5. Mjerni instrumenti i analiza podataka	72

4. MODEL POUČAVANJA AGENTSKE PARADIGME KORIŠTENJEM SIMULACIJSKIH OKRUŽENJA	75
4.1. Prvi dio istraživanja	75
4.1.1. Inicijalna provjera znanja	76
4.1.2. Definicija, osnovna svojstva i vrste agenata (NJ1)	78
4.1.3. Definicija, osnovna svojstva i vrste okruženja (NJ2).....	91
4.1.4. Interakcija agenata i okruženja i međusobna interakcija agenata u višeagentskim sustavima (NJ3)	103
4.1.5. Područja primjene agenata (NJ4)	112
4.1.6. Analiza ukupnih rezultata provjera znanja nakon nastavnih jedinica	121
4.1.7. Kvalitativna analiza nakon prvog dijela istraživanja	123
4.2. Drugi dio istraživanja	126
4.2.1. Istraživanje u okviru predmeta Uvod u računarstvo na preddiplomskom studiju informatike	126
4.2.2. Istraživanje u okviru predmeta Umjetna inteligencija na diplomskim studijima strojarstva i računarstva	128
4.2.3. Analiza ujednačenosti skupina	131
4.2.4. Dodatna analiza postignutih rezultata	133
4.3. Treći dio istraživanja	134
4.3.1. Inicijalna provjera znanja	134
4.3.2. Istraživanje u okviru predmeta Umjetna inteligencija na diplomskom studiju računarstva	136
4.3.3. Provjera znanja s vremenskim odmakom.....	139
4.4. Prijedlog modela poučavanja agentske paradigme korištenjem siumlacijskih okruženja	142
5. ZAKLJUČAK	145
6. LITERATURA	148

PRILOZI.....	167
ŽIVOTOPIS I POPIS JAVNO OBJAVLJENIH RADOVA	201

POPIS SLIKA I TABLICA:

Popis slika:

Slika 1. Shematski prikaz agenta.....	19
Slika 2. Shematski prikaz jednostavnog reaktivnog agenta	21
Slika 3. Pseudokod agentske funkcije jednostavnog reaktivnog agenta	22
Slika 4. Shematski prikaz reaktivnog agenta sa stanjima.....	23
Slika 5. Pseudokod agentske funkcije reaktivnog agenta sa stanjima.....	23
Slika 6. Shematski prikaz agenta usmjerenog prema cilju.....	24
Slika 7. Pseudokod agentske funkcije agenta usmjerenog prema cilju	24
Slika 8. Shematski prikaz agenta usmjerenog prema uspješnosti	25
Slika 9. Pseudokod agentske funkcije agenta usmjerenog prema cilju	25
Slika 10. Shematski prikaz agenta koji uči.....	26
Slika 11. Trodimenzionalna matrica za klasifikaciju agentskih sustava	28
Slika 12. Shematski prikaz klasifikacije agenata na osnovi minimalne skupine svojstava	29
Slika 13. Shematski prikaz prvog dijela istraživanja	71
Slika 14. Shematski prikaz drugog dijela istraživanja	71
Slika 15. Shematski prikaz trećeg dijela istraživanja.....	72
Slika 16. Pseudokod agentske funkcije jednostavnog reaktivnog agenta	79
Slika 17. Jednostavni reaktivni agent koji pokreće autonomno vozilo u <i>Cellularu</i>	79
Slika 18. Jednostavni reaktivni agent koji pokreće autonomno vozilo u <i>NetLogu</i>	80
Slika 19. Jednostavni reaktivni agent koji pokreće rover na Marsu u <i>Cellularu</i>	81
Slika 20. Jednostavni reaktivni agent koji pokreće rover na Marsu u <i>NetLogu</i>	82
Slika 21. Pseudokod agentske funkcije reaktivnog agenta sa stanjima.....	82
Slika 22. Reaktivni agent sa stanjima koji pokreće rover na Marsu u <i>Cellularu</i>	83
Slika 23. Reaktivni agent sa stanjima koji pokreće rover na Marsu u <i>NetLogu</i>	84
Slika 24. Reaktivni agent sa stanjima koji pokreće pametni usisavač u <i>Cellularu</i>	84
Slika 25. Reaktivni agent sa stanjima koji pokreće pametni usisavač u <i>NetLogu</i>	85
Slika 26. Pseudokod agentske funkcije agenta usmjerenog prema cilju.....	86
Slika 27. Agent usmjeren prema cilju koji pokreće pametna kolica u trgovini u <i>Cellularu</i>	86
Slika 28. Agent usmjeren prema cilju koji pokreće pametna kolica u trgovini u <i>NetLogu</i>	87
Slika 29. Pseudokod agentske funkcije agenta usmjerenog prema uspješnosti	88
Slika 30. Agent usmjeren prema uspješnosti koji određuje optimalnu rutu vozila u <i>Cellularu</i>	89

Slika 31. Agent usmjeren prema uspješnosti koji određuje optimalnu rutu vozila u <i>NetLogu</i>	89
Slika 32. Determinističko okruženje po kriteriju determinističnosti u <i>Cellularu</i>	93
Slika 33. Determinističko okruženje po kriteriju determinističnosti u <i>NetLogu</i>	94
Slika 34. Stohastičko okruženje prema kriteriju determinističnosti u <i>Cellularu</i>	95
Slika 35. Stohastičko okruženje prema kriteriju determinističnosti u <i>NetLogu</i>	95
Slika 36. Potpuno vidljivo okruženje po kriteriju dostupnosti informacija u <i>Cellularu</i>	96
Slika 37. Potpuno vidljivo okruženje po kriteriju dostupnosti informacija u <i>NetLogu</i>	97
Slika 38. Djelomično vidljivo okruženje po kriteriju dostupnosti informacija u <i>Cellularu</i>	97
Slika 39. Djelomično vidljivo okruženje po kriteriju dostupnosti informacija u <i>NetLogu</i>	98
Slika 40. Epizodično okruženje po kriteriju epizodičnosti u <i>Cellularu</i>	99
Slika 41. Epizodično okruženje po kriteriju epizodičnosti u <i>NetLogu</i>	99
Slika 42. Sekvencijalno okruženje po kriteriju epizodičnosti u <i>Cellularu</i>	100
Slika 43. Sekvencijalno okruženje po kriteriju epizodičnosti u <i>NetLogu</i>	101
Slika 44. Neovisnost u višeagentskom sustavu u <i>Cellularu</i>	105
Slika 45. Neovisnost u višeagentskom sustavu u <i>NetLogu</i>	105
Slika 46. Jednostavna suradnja u višeagentskom sustavu u <i>Cellularu</i>	106
Slika 47. Jednostavna suradnja u višeagentskom sustavu u <i>NetLogu</i>	107
Slika 48. Opstrukcija u višeagentskom sustavu u <i>Cellularu</i>	108
Slika 49. Opstrukcija u višeagentskom sustavu u <i>NetLogu</i>	108
Slika 50. Čisto pojedinačno natjecanje u višeagentskom sustavu u <i>Cellularu</i>	109
Slika 51. Čisto pojedinačno natjecanje u višeagentskom sustavu u <i>NetLogu</i>	110
Slika 52. Simulacija kolonije mrava u prikupljanju hrane u <i>Cellularu</i>	114
Slika 53. Simulacija kolonije mrava u prikupljanju hrane u <i>NetLogu</i>	115
Slika 54. Simulacija epidemije zarazne bolesti u <i>Cellularu</i>	116
Slika 55. Simulacija epidemije zarazne bolesti u <i>NetLogu</i>	116
Slika 56. Simulacija segregacijskog modela Thomasa Schellinga u <i>Cellularu</i>	117
Slika 57. Simulacija segregacijskog modela Thomasa Schellinga u <i>NetLogu</i>	118
Slika 58. Simulacija širenja šumskog požara u <i>Cellularu</i>	118
Slika 59. Simulacija širenja šumskog požara u <i>NetLogu</i>	119

Popis tablica:

Tablica 1. Pregled istraživanja alata za modeliranje i simuliranje zasnovanih na agentima....	55
Tablica 2. Pregled ključnih značajki alata za modeliranje i simuliranje zasnovanih na agentima	60
Tablica 3. Analiza rezultata inicijalne provjere znanja	77
Tablica 4. Kruskal-Wallisov test za tri nezavisne skupine.....	77
Tablica 5. Analiza rezultata ispitivanja znanja nakon NJ1	90
Tablica 6. Kruskal-Wallisov test rezultata ispitivanja znanja nakon NJ1 za tri nezavisne skupine	91
Tablica 7. Mann-Whitneyev U test rezultata ispitivanja znanja nakon NJ1 za parove nezavisnih skupina	91
Tablica 8. Analiza rezultata ispitivanja znanja nakon NJ2	102
Tablica 9. Kruskal-Wallisov test rezultata ispitivanja znanja nakon NJ2 za tri nezavisne skupine	102
Tablica 10. Mann-Whitneyev U test rezultata ispitivanja znanja nakon NJ2 za parove nezavisnih skupine	103
Tablica 11. Analiza rezultata ispitivanja znanja nakon NJ3	111
Tablica 12. Kruskal-Wallisov test rezultata ispitivanja znanja nakon NJ3 za tri nezavisne skupine	111
Tablica 13. Mann-Whitneyev U test rezultata ispitivanja znanja nakon NJ2 za parove nezavisnih skupine	112
Tablica 14. Analiza rezultata ispitivanja znanja nakon NJ4	120
Tablica 15. Kruskal-Wallisov test rezultata ispitivanja znanja nakon NJ4 za tri nezavisne skupine	120
Tablica 16. Mann-Whitneyev U test rezultata ispitivanja znanja nakon NJ2 za parove nezavisnih skupine	121
Tablica 17. Analiza rezultata ispitivanja znanja nakon NJ4	127
Tablica 18. Rezultati Shapiro-Wilkovog testa normalnosti distribucije rezultata za kontrolnu i eksperimentalnu skupinu studenata na prvoj godini preddiplomskog studija informatike	128
Tablica 19. Neparni t-test rezultata ispitivanja znanja nakon nastavne jedinice za dvije nezavisne skupine studenata preddiplomskog studija informatike	128
Tablica 20. Analiza rezultata ispitivanja znanja nakon NJ4	130

Tablica 21. Rezultati Shapiro-Wilkovog testa normalnosti distribucije rezultata za kontrolnu i eksperimentalnu skupinu studenata na prvoj godini diplomskih studija strojarstva i računarstva	130
Tablica 22. Neparni t-test rezultata ispitivanja znanja nakon nastavne jedinice za dvije nezavisne skupine studenata diplomskih studija strojarstva i računarstva.....	131
Tablica 23. Analiza ostvarenih ocjena iz predmeta u okviru kojih je istraživanje provedeno	132
Tablica 24. Rezultati Shapiro-Wilkovog testa normalnosti distribucije rezultata za kontrolne i eksperimentalne skupine studenata uključenih u drugi dio istraživanja	132
Tablica 25. Neparni t-test ostvarenih ocjena za dvije nezavisne skupine studenata preddiplomskog studija informatike i dvije nezavisne skupine studenata diplomskih studija strojarstva i računarstva.....	133
Tablica 26. Neparni t-test rezultata ispitivanja znanja za dvije nezavisne kontrolne skupine studenata i dvije nezavisne eksperimentalne skupine studenata	134
Tablica 27. Analiza rezultata inicijalne provjer znanja.....	135
Tablica 28. Rezultati Shapiro-Wilkovog testa normalnosti distribucije rezultata za kontrolnu i eksperimentalnu skupinu studenata na prvoj godini diplomskog studija računarstva	136
Tablica 29. Neparni t-test rezultata ispitivanja znanja nakon nastavne jedinice za dvije nezavisne skupine studenata diplomskih studija strojarstva i računarstva.....	136
Tablica 30. Analiza rezultata ispitivanja znanja nakon NJ2	137
Tablica 31. Rezultati Shapiro-Wilkovog testa normalnosti distribucije rezultata za kontrolnu i eksperimentalnu skupinu studenata na prvoj godini diplomskog studija računarstva	138
Tablica 32. Neparni t-test rezultata ispitivanja znanja nakon nastavne jedinice za dvije nezavisne skupine studenata diplomskog studija računarstva.....	138

1. UVOD

Suvremeni život, život kojeg živimo u digitalnom dobu, izrazito je ovisan o tehnologiji. Trenutno se nalazimo u razdoblju najvećih fundamentalnih promjena još od Industrijske revolucije. Eksponencijalni tehnološki napredak omogućio je instrumentaliziranje, međusobno povezivanje i prožimanje inteligentnim dizajnom tehnoloških artefakata u cijelom nizu ljudskih djelatnosti: svakodnevnom životu, industriji, prijevozu, energetici, zdravstvu, pa i u obrazovanju, dajući im karakteristike inteligentnih ili pametnih ljudskih djelatnosti (Lévy i Bononno, 1998). Bogatstvo podataka i eksponencijalni rast novih znanja izazivaju obrazovne institucije da razmisle o novim modelima i pristupima učenju i poučavanju.

Današnji učenici i studenti pripadaju generaciji digitalnih urođenika, odraslih uz novu tehnologiju (Prensky, 2001). Oni su proveli čitav svoj život u okruženju i aktivnoj interakciji s računalima, video igrama, digitalnim glazbenim uređajima, video kamerama, mobilnim telefonima i svim drugim alatima i igračkama digitalnog doba.

S tehnologijom kao katalizatorom, obrazovanje se kreće od modela transfera znanja ka kolaborativnom, aktivnom, uključujućem modelu koji pomaže studentima da povećaju svoje znanje i razviju vještine potrebne za uspjeh u društvu koje uči (engl. *Learning Society*) (Selinger i sur., 2013) i pripremi ih za povećanu konkurenciju na tržištu rada. Tehnološka znanja i sposobnosti koje nadilaze usko specijalizirana područja postaju ključne osobine potrebne u modernoj industriji i životu općenito. Ovakve društvene okolnosti uvjetuju potrebu za obrazovnim metodama koje će omogućiti razvoj kreativnog razmišljanja, donošenja odluka, rješavanje problema, komunikacije i vještine timskog rada i njihovo stavljanje u kontekst stečenog znanja. Ove su vještine potrebne na radnom mjestu u 21. stoljeću.

Od iznimnog značaja je i korištenje suvremenih informacijskih i komunikacijskih tehnologija (engl. *Information and Communication Technologies*) u obrazovnom procesu. Glavni doprinos informacijskih i komunikacijskih tehnologija procesu učenja i poučavanja posljedica je njegovih tehnoloških karakteristika koje se odnose na upravljanje velikim količinama podataka i informacija u kratkom vremenu, njihov dinamični i interaktivni prikaz, te način komunikacije koji oni pružaju. Korištenje ovih mogućnosti informacijskih i komunikacijskih tehnologija omogućava aktivno sudjelovanje učenika i nastavnika, njihovu međusobnu interakciju, kao i interakciju s korištenim tehnološkim nastavnim pomagalicama te procese koji podržavaju stvaranje mentalnih modela (Mikropoulos i Bellou, 2013).

Obrazovna tehnologija predstavlja proučavanje i praktičnu primjenu tehnoloških procesa i resursa, njihovo kreiranje, upotrebu i upravljanje s ciljem olakšavanja učenja i poboljšanja performansi podučavanja (Richey, 2008). Ona uključuje sve valjane i pouzdane primijenjene obrazovne znanosti te procese i postupke koji su izvedeni iz znanstvenih istraživanja, kao i tehnološku opremu namijenjenu primjeni u obrazovnom procesu te teorijske, algoritamske ili heurističke procese koji se pri tome odvijaju.

Važna uloga koju u suvremenom obrazovanju ima obrazovna tehnologija ogleda se i u transformaciji tradicionalnog modela didaktičkog trokuta. Vrhove didaktičkog trokuta predstavljaju učenik, nastavnik i nastavni sadržaj. Korijene ovog pristupa možemo pronaći već u antičko doba (Zierer i Seel, 2012). Tako Aristotel razvija ideju retoričkog trokuta, kojeg čine govornik, publika i tema. Ideju o trokutu preuzeli su Ciceron u svom djelu *De Oratore* i Quintilian u *Institutio Oratoria*. Oni mu dodaju pedagoški i didaktički kontekst, jer u svojim djelima pišu o obrazovanju govornika te proširuju retorički trokut na pojam didaktičkog trokuta. Uz tradicionalne vrhove didaktičkog trokuta, tj. učenika, nastavnika i nastavnog sadržaja – koji predstavljaju odgovore na pitanja koga, kako i što se poučava, sve veći značaj dobiva i četvrti vrh, tj. tehnologija (Goodchild i Sriraman, 2012; Rezat i Strasser, 2012; Seghroucheni i sur., 2014). Ona dodaje novu dimenziju procesu učenja i poučavanja kreirajući didaktički tetraedar (Ruthven, 2012) čiji su vrhovi čvrsto povezani. Definiranjem pojedinog vrha mijenjaju se, tj. prilagođavaju i ostali vrhovi, pa će tako odabrani nastavni sadržaj odrediti karakteristike učenika kojima je namijenjen (starosnu dob, potrebno predznanje i sl.), strategiju poučavanja, kao i primjerenu tehnologiju kojom se nastavnik može poslužiti u nastavi.

Brz razvoj informacijskih i komunikacijskih tehnologija u posljednjih nekoliko desetljeća pružio je široj znanstvenoj i stručnoj zajednici mogućnost pristupa problemima na ranije nezamislive načine. Mnogim prirodnim, društvenim i tehničkim fenomenima i sustavima je teško pristupiti ili su oni u potpunosti nedostupni. Ovi fenomeni mogu biti opasni za promatranje i istraživanje ili apstraktni i nevidljivi ljudskom oku. Primjenom modernih računalnih tehnologija moguće je modelirati i simulirati takve pojave i sustave te manipulirati s njima te, na taj način, analizirati sustav i procijeniti strategije za njegovo funkcioniranje. Nuklearna fuzija i fisija, sunčeve baklje, mjesečeva gravitacija, laparoskopske operacije, redovi za čekanje u banci ili bolnici, javni transport ljeti i tereta, organizacija proizvodnje, rad sastavnica računala, funkcioniranje apstraktnih računalnih programa i dr. primjeri su fenomena i sustava koje je, uz pomoć tehnologije, moguće u nastavnom procesu predstaviti na vizualno prihvatljiv i zanimljiv način, a učeniku ili studentu omogućiti aktivno sudjelovanje u procesu

učenja i poučavanja. Važnost aktivnog angažmana osobe koja uči i konstruira znanje za sebe, nadograđujući novo znanje i iskustvo na ono već postojeće osnova je konstruktivističke teorije učenja.

Potreba za alatima programske podrške koji će omogućiti izučavanje fenomena i sustava korištenjem njihovih simulacijskih modela potaknula je razvoj velikog broja rješenja, različite osnovne svrhe i područja primjene. Različite skupine korisnika ističu važnost različitih aspekata ove skupine alata, u skladu s njihovim potrebama i mogućnostima. Znanstvenici, bez iskustva u programiranju, žele jednostavan alat s intuitivnim sučeljem. S druge strane, računalni stručnjaci i informatičari zainteresirani su za vrstu licence i otvorenost koda koji će im omogućiti manipuliranje alatom. Nastavnici, koji u nastavi koriste alate za modeliranje i simuliranje, žele jednostavne pakete koji učenicima omogućuju da prenesu znanje i iskustvo u stvarne situacije.

Fizički i apstraktni elementi ovih sustava mogu se implementirati kao entiteti programske podrške koji mogu percipirati svoju okolinu i samostalno reagirati na stimulaciju. Ovaj se opis u velikoj mjeri podudara sa široko prihvaćenom definicijom računalnih agenta, što izučavanju agentske paradigme, u okviru obrazovnog procesa, daje izuzetan značaj.

Još jedno potencijalno značajno područje primjene usvojenih znanja agentske paradigme je područje Interneta stvari (engl. *Internet of Things*). Osnovni koncept Interneta stvari je da se svakodnevni predmeti mogu opremiti identifikacijskim, senzorskim, mrežnim i procesnim mogućnostima koje će ih osposobiti za međusobnu komunikaciju, kao i komunikaciju s drugim uređajima i uslugama preko Interneta, kako bi se postigao neki koristan cilj (Ashton, 2009; Gubbi i sur., 2013). Internet stvari predstavlja evoluciju korištenja postojećih tehnologija: tehnologije radio frekvencijske identifikacije (engl. *Radio-Frequency Identification*), senzorskih mreža, međusobne komunikacije uređaja i sl., u smislu broja i vrsti uređaja, kao i povezivanja mreža tih uređaja preko Interneta. Od posebnog značaja za studente tehničkih fakulteta je i primjena koncepta Interneta stvari u industrijskoj proizvodnji, gdje on predstavlja komunikacijski kanal za suradnju pametnih proizvodnih sustava, međusobno i s ljudima. Ova primjena označava se kao Industrija 4.0 (Shrouf i sur., 2014). Kao inteligentno operacionalno proširenje Interneta stvari, javlja se u posljednjih nekoliko godina koncept Agenata stvari (engl. *Agents of Things*) (Mzahm i sur., 2013). Identifikacijske elektroničke oznake predmeta u Internetu stvari mogu se nadograditi softverskim agentima kako bi se, na osnovi njihove interakcije, unaprijedile usluge koje se mogu dobiti od objekata povezanih u Internet stvari.

Ovo dodatno potvrđuje potrebu za sustavnim pristupom izučavanju apstraktnih koncepata agentske paradigme, ali i metoda i tehnika modeliranja i simulacija realnih i apstraktnih sustava, kako bi se budući znanstvenici i stručnjaci upoznali s ovom tehnologijom budućnosti.

Sve gore navedeno poslužilo je kao motivacija za istraživanje kojim se htjelo potvrditi pozitivan učinak korištenja alata za modeliranje i simuliranje u izučavanju apstraktnih koncepata agentske paradigme.

Cilj istraživanja je utvrditi način na koji primjena postojećih simulacijskih okruženja utječe na usvajanje i razumijevanje koncepata agentske paradigme kod studenata u visokom obrazovanju. Prema rezultatima istraživanja oblikovat će se model poučavanja agentske paradigme koji će sadržavati karakteristične primjere i zadatke iz ovog područja prikazane korištenjem simulacijskih okruženja. Oblikovani model poučavanja agentske paradigme na visokoškolskoj razini učinit će se javno dostupnim.

Postavljena je polazna hipoteza prema kojoj će studenti informatike i računarstva postići bolje rezultate u usvajanju i razumijevanju osnovnih koncepata agentske paradigme uz poučavanje temeljeno na korištenju simulacijskih okruženja u odnosu na klasičan način poučavanja *ex cathedra*.

Istraživačka pitanja na koja će ovo istraživanje dati odgovor su:

- Utječe li način poučavanja agentske paradigme na uspješnost studenata u usvajanju i razumijevanju nastavnih sadržaja ovog područja?
- Postoje li razlike u uspješnosti usvajanja i razumijevanja nastavnih sadržaja iz područja agentske paradigme koje su posljedica odabranog simulacijskog okruženja?

Na početku rada dane su teoretske osnove područja obuhvaćenih ovim interdisciplinarnim istraživanjem. Ovo uključuje pojmove modeliranja i simuliranja, agentske paradigme i teorije znanja i učenja koje opravdavaju i podržavaju primjenu spomenute tehnologije u konkretnoj obrazovnoj situaciji. Predstavljen je i cjeloviti pregled istraživanja u spomenutim područjima te spona koje ih povezuju.

Modeliranje i simuliranje zasnovano na agentima (engl. *Agent-Based Modelling and Simulation*) predstavlja kategoriju računalnih modela koji simuliraju dinamičke akcije i reakcije agenata na podražaje iz okruženja, kao i protokole za međusobno komuniciranje agenata u zajedničkom okruženju, sa ciljem kritičkog osvrta i procjene njihovog dizajna i izvedbe, kao i uvida u njihovo ponašanje i svojstva (Abar i sur., 2017). Sa stajališta simulacije, funkcija pojedinog agenta može varirati od osnovnih uzročno-posljedičnih pravila do sofisticiranijih,

kognitivno bogatih modela ponašanja. Alati za modeliranje i simuliranje zasnovani na agentima (engl. *Agent-Based Modelling and Simulation Tools*) predstavljaju programsku podršku namijenjenu stvaranju i upravljanju ovih računalnih modela, te praćenju ponašanja realiziranih agenata u simuliranom okruženju.

U ovom radu je, između ostalog, dan i sveobuhvatan pregled dijela postojećih alata za modeliranje i simuliranje zasnovanih na agentima koji su svojom namjenom i izvedbom primjereni upotrebi u nastavi. Pri tome su navedene karakteristike alata koje mogu biti ključne prilikom izbora nekog od njih. U tom smislu, ovaj rad može poslužiti i kao katalog alata za modeliranje i simuliranje zasnovanih na agentima, koristan nastavnicima koji žele modeliranjem i simuliranjem obogatiti nastavni sadržaj koji predstavljaju učenicima.

Rad donosi i opsežan pregled provedenog istraživanja u nastavi, u okviru kojeg se poučavanju agentske paradigme pristupilo na novi, tehnološki suvremen način uz korištenje simulacijskih okruženja alata za modeliranje i simuliranje, kao i karakterističnih primjera kojima su predstavljeni apstraktni koncepti područja.

Provedena analiza ima za cilj, na osnovi prikupljenih rezultata, utvrditi pozitivan učinak primijenjene nastavne metode na pojedinim nastavnim jedinicama područja agentske paradigme, Na osnovi analize predložen je i detaljno razrađen model poučavanja agentske paradigme korištenjem simulacijskih okruženja.

2. DOSADAŠNJE SPOZNAJE

U ovom dijelu rada osvrnut ćemo se na dosadašnje spoznaje u područjima obuhvaćenim interdisciplinarnom prirodnom provedenog istraživanja.

2.1. Znanje i teorije učenja

Obrazovanje i znanje, kao posljedica obrazovanja, su pretpostavke razvitka suvremenog društva. Zahvaljujući njima, pojedinac se može aktivno uključiti u društvo, preuzeti različite društvene uloge i orijentirati se u njemu. Kako su znanja rezultat i znanstveno-tehnološkog razvitka, ona se ubrzano mijenjaju, što od pojedinca zahtijeva neprestano učenje i usavršavanje, tj. stjecanje novih znanja, vještina, kompetencija i sposobnost njihove primjene u praksi (Vilić, 2014).

U suvremenom društvu, društvu znanja, dolazi do transformacije obrazovanja prema širem konceptu učenja unutar i izvan institucionalnih, fizičkih i vremenskih okvira. To je omogućeno razvojem suvremenih informacijskih i komunikacijskih tehnologija. Pojedinci sada mogu usvajati nova znanja gdje im to odgovara, kada im to odgovara i na način na koji im to odgovara. Aktivnim sudjelovanjem u obrazovnom procesu, oni usvajaju znanja i vještine koje ih čine vrijednim resursom društva.

Obrazovanje u kojemu je aktivnost učenika još uvijek sporedan čimbenik ne postiže očekivane rezultate, a nedovoljna pripremljenost učenika za daljnji nastavak školovanja i nesnalaženje prilikom suočavanja sa životnim situacijama, postaje ključni problem današnjice (Vidulin-Orbanić, 2007).

U nastavku definiramo pojam znanja, objašnjavamo teorije učenja koje na različite načine objašnjavaju mentalne procese usvajanja znanja te naglašavamo konstruktivističku teoriju učenja koja najjače podržava model poučavanja korištenjem simulacijskih okruženja koji zahtijevaju aktivno sudjelovanje učenika u procesu učenja i poučavanja.

2.1.1. Definicija znanja

Pojam znanja možemo definirati kao skup činjenica, informacija i vještina stečenih iskustvom ili obrazovanjem. Znanje predstavlja poznavanje ili svijest o nekoj činjenici ili situaciji.

Tradicionalno, realističko poimanje koncepta znanja zasniva se na zdravorazumskom uvjerenju da stvarni svijet postoji bez obzira na to je li pojedinac za njega zainteresiran ili ga uopće ne primjećuje. Čovjek, iz ove perspektive, usvaja znanje praveći kopije ili replike stvarnosti u

svom umu. Ovakav pristup nužno vodi do pitanja koliko dobro naše znanje odgovara stvarnosti. Prema realističkoj perspektivi, odnos znanja i istina je pitanje korespondencije, tj. istinito je ono što odgovara objektivnoj stvarnosti, a lažno ono što joj ne odgovara (Smith, 1983).

Konstruktivistički model se može sažeti u tvrdnji da se znanje konstruira u umu učenika. Pri tome učenik izgrađuje razumijevanje opažene činjenice ili situacije. Ovaj proces nadilazi puko zrcaljenje i odražavanje onoga što mu je rečeno ili što je pročitao, jer učenik traži smisao i pokušava uklopiti novo znanje u postojeće, čak i u nedostatku potpune informacije (Resnick, 1983).

Konstrukciju znanja konstruktivisti opisuju kao potragu za prikladnošću, a ne podudaranjem sa stvarnošću (von Glasersfeld, 1986). Znanje nije preslika realnosti, nego se uklapa u nju na način specifičan za svakog pojedinca. Svatko od nas gradi vlastiti pogled na stvarnost pokušavajući uspostaviti red u kaosu podražaja kojima su obasuta naša osjetila. Pri tome je od ključne važnosti da znanje, koje pojedinac konstruira na osnovi neke informacije, zadovoljavajuće funkcionira u kontekstu u kojem se javlja.

Razliku između tradicionalnog i konstruktivističkog pogleda na znanje možemo prikazati i na sljedeći način. Tradicionalni, bihevioristički pogled na znanje gleda na um kao na crnu kutiju, pri čemu možemo točno procijeniti što u njega ulazi (podražaj okoline) i što izlazi (odgovor), ali možemo samo nagađati što se događa unutar kutije. Konstruktivističko shvaćanje znanja promatra okoliš kao crnu kutiju, pri čemu pojedinac zna što se događa u njegovom umu, ali može samo nagađati kakav je odnos njegovih mentalnih struktura i stvarnog svijeta (Bodner, 1986).

Način na koji definiramo i obrazložimo pojam znanja uvelike utječe i na principe tehnike i metode kojima želimo dovesti do usvajanja tog znanja. Na tim osnovama razvile su se različite teorije učenja.

2.1.2. Teorije učenja

Učenje je složeni psihički proces stjecanja znanja kojim se danas bave različite grane znanosti. Na osnovi različitih spoznaja o načinima i mogućnostima učenja, znanstvenici ga objašnjavaju teorijama učenja na kojima počivaju i sukladne didaktičke teorije (Jukić, 2013). Najčešće govorimo o biheviorističkom, kognitivističkom i konstruktivističkom pristupu objašnjavanja procesa učenja.

Biheviorizam se razvija u prvoj polovici 20. stoljeća, a prema njemu je učenje povezano s jačanjem veze između podražaja i reakcije na podražaj. Ovaj odnos se jača potkrepljivanjem,

odnosno nagrađivanjem, kako bi se postiglo željeno ponašanje (Ertmer i Newby, 1993). Tradicionalni bihevioristički model poučavanja zasniva se na pretpostavci da se znanje može prenijeti netaknuto iz uma učitelja u um učenika. Nastavnici bi se, prema ovom modelu, trebali usredotočiti na prenošenje znanja učenicima, a istraživači u području obrazovanja pokušati pronaći bolje načine za to (von Glasersfeld, 2008). Ipak, iskustva brojnih nastavnika čvrsto podržavaju hipotezu prema kojoj poučavanje i učenje nisu sinonimi, tj. nastavnik može poučavati, i to poučavati dobro i kvalitetno, a da pri tome izostane željeni rezultat, tj. da učenici ipak ne nauče.

Dok biheviorizam ističe središnju ulogu nastavnika i značaj okoline u izazivanju podražaja, kognitivizam naglašava ulogu učenika u procesu učenja na koji gledaju kao na proces organiziranja, povezivanja i generiranja informacija. Prema kognitivistima, učenje predstavlja razvijanje misaonih struktura kojima učenik predstavlja vanjske ili unutarnje procese i pojave. Nove misaone sheme koje pri tome nastaju, uklapaju se u postojeće znanje ili se postojeće znanje prilagođava kako bi se nove spoznaje u njega mogle uklopiti (Bower i Hilgard, 1981). Kognitivistički pristup, pri tome, zanemaruje kontekst i socijalne aspekte u kojima se učenje odvija (Chen, 2003)

Kao suprotnost biheviorizmu, po kojemu znanstveno znanje egzistira neovisno o percepciji i svijesti pojedinca, statično je i nepromjenjivo, tj. predstavlja utjelovljenje univerzalne istine (Murphy, 1997), nastaje konstruktivistički pristup učenju. Korijeni konstruktivizma mogu se pronaći u radovima kognitivnih psihologa, između ostalih J. Piageta, J. Brunera i U. Niessera, zbog čega njihov odnos prema konstruktivizmu ne podrazumijeva oprečnost već naglašavanje i proširivanje pojedinih elemenata kognitivističkog pristupa (Perkins, 1992). Konstruktivizam predstavlja teoriju učenja koja naglašava važnost aktivnog angažmana osobe koja uči i konstruira znanje za sebe, nadograđujući novo znanje i iskustvo na ono već postojeće.

Upravo je konstruktivizam, s naglaskom na aktivni angažman učenika kao preduvjet uspješnog usvajanja znanja, teorija na koju se najviše oslanja primjena simulacijskih okruženja, kao oblika obrazovne tehnologije, u nastavi.

2.1.3. Konstruktivistički pristup učenju i poučavanju

Konstruktivizam u pedagogiji podrazumijeva epistemologijsku i obrazovnu teoriju. Ta dva aspekta mogu biti odvojena i egzistirati posebno. Zagovaranje konstruktivističke teorije znanosti može istodobno značiti negiranje konstruktivističke pedagogije i obrnuto. Iako se konstruktivizam najprije pojavljuje kao teorija učenja, ideja se kasnije progresivno proširila na

niz drugih aspekata. Konstruktivizam postaje teorija poučavanja i teorija obrazovanja, odnosno pedagoška teorija (Babić, 2007), ali istovremeno i teorija kognicije, osobnog znanja, znanstvenog znanja, obrazovne etike i politike te pogled na svijet (Matthews, 2000).

Konstruktivizam je teorija učenja i poučavanja čiji je začetnik francuski psiholog Jean Piaget. Osnove razvoja konstruktivističke teorije mogu se pronaći u potrazi za načinom stjecanja znanja, s naglaskom na ponašanje učenika. Prema konstruktivizmu znanje nije moguće jednostavno prenositi, nego se ono izgrađuje u procesu, na osnovi iskustava iz stvarnog svijeta i uz aktivno sudjelovanje učenika (Piaget, 1974). Stjecanje znanja javlja se kroz stvaranje mentalnih modela, koji su rezultat kombiniranja prethodnih informacija s novima. Dakle, izgradnja znanja zasnovana je na iskustvu pojedinca i, kao takva, jedinstvena za svakog učenika. Do novih informacija može se doći i kognitivnim putem, tj. analizom, usporedbom i kombiniranjem prethodno dostupnih informacija (Ben-Ari, 2001).

Unutar konstruktivističke teorije razvijeni su različiti pristupi i interpretacije. U spektru konstruktivističkih pristupa i pogleda najčešće se ističu: trivijalni, radikalni, socijalni, kulturalni i kritički konstruktivizam (Dougiamas, 1998).

2.3.3.1. Trivijalni konstruktivizam

Najjednostavnija ideja konstruktivizma i korijen svih ostalih pogleda je trivijalni ili osobni konstruktivizam koji se temelji na Piagetom načelu prema kojem učenik aktivno izgrađuje znanje, a ne prima ga pasivno iz okruženja. Prethodno znanje učenika je neophodno za aktivno konstruiranje novih znanja procesima asimilacije i akomodacije. Trivijalni konstruktivizam, međutim, ne odgovara na brojna pitanja koja se nameću: o prirodi i karakteristikama okruženja i znanja, o njihovom međusobnom odnosu ili prednostima pojedinih okruženja u procesu učenja. Ove nedostatke trivijalnog konstruktivizma pokušavaju riješiti drugi konstruktivistički pogledi.

2.3.3.2. Radikalni konstruktivizam

Osnovu radikalnog konstruktivizma čine intrapsihički procesi koji su izvor konstrukcije stvarnosti u svijesti pojedinca (Jukić, 2013).

Radikalni konstruktivizam ima korijen u samoizgradnji smisla, značenja ili ideja o objektima i pojavama oko nas. Ovaj proces konstruiranja znanja od strane pojedinca duboko je povezan s metaforičkim razmišljanjem. Kada pojedinac razmišlja o objektima ili fenomenima, shvaća ih u kognitivnim konceptima, percepcijama ili konstruktima u svijesti. Pri tome prolazi kroz proces refleksije i apstrakcije koji predstavljaju neku vrstu metaforičkog razmišljanja jer

razmišlja o iskustvima u smislu metafore i dostiže dubinu apstrakcije kroz uporabu metafora (Belbase, 2011). Ove metafore postaju sredstvo konceptualizacije ideja, odnosno konstrukcije novih znanja. Metafore se ogledaju u razmišljanju o jednom objektu ili pojavi kroz perspektivu drugog objekta ili pojave, razmišljanje o jednoj ideji kroz aspekt druge ideje, konceptualizaciju jednog objekta u smislu drugog objekta, razumijevanje novog odnosa u smislu ranije poznatih odnosa te povezivanje jedne ideje s drugom mostom koji povezuje jednostavno i složeno, konkretno s apstraktnim i konačno s beskonačnim (Lakoff i Johnson, 1980).

Paradigma radikalnog konstruktivizma gleda na ljudski um kao na organizam koji prolazi kroz evolucijski proces. Metafora evolucije uma predstavlja prilagodbu na bolje i jasnije koncepte ili mentalne percepcije kroz reorganizaciju iskustvenog svijeta (Ernest, 1995). Pri tome je iskustveni svijet ustvari subjektivni svijet pojedinca. Objektivna stvarnost možda i postoji, ali ne postoji način da se to uistinu i potvrdi. Sve što prihvaćamo kao stvarnost, odnosno sve što znamo o svijetu, predstavlja iskustveni svijet pojedinca.

Konstrukcija značenja stvari i pojava u tom svijetu je, za radikalne konstruktiviste, individualni proces. Jedinka konstruira svoju stvarnost autonomno (Palekčić, 2007). Učenje predstavlja proces samoregulacije i tvorbe konceptualnih struktura putem refleksije i samoregulacije (von Glasersfeld, 2001), a znanje se konstruira izvan utjecaja socijalno-kulturnog konteksta.

2.3.3.3. Socijalni konstruktivizam

Osnovu socijalnog konstruktivizma čine interpsihički procesi, odnosno oblici i sadržaji diskursa između pojedinaca, koji vode konstrukciji stvarnosti u njihovoj svijesti (Jukić, 2013).

Socijalni konstruktivizam, s druge strane, odražava one teorije razvitka čovjeka koje pojedinca smještaju u sociokulturni kontekst. Učenje, prema socijalnim konstruktivistima, nije samo prihvaćanje i uklapanje novih znanja u postojeća, ono je i proces u kojem su učenici integrirani u zajednicu. Pri tome individualni razvoj pojedinca proizlazi iz njegovih socijalnih interakcija (Cole i Wertsch, 1996; Richardson, 1997). Znanje se stječe, odnosno konstruira, komunikacijom i interakcijom pojedinaca s okruženjem u kojem se nalazi.

Svijet oko nas je društveno konstruirani svijet. Individualni svjetovi svakog pojedinca dijelovi su kolektivnog društvenog svijeta. Ne postoji izolirana individualna stvarnost, izdvojena od društveno i kulturno konstruiranog svijeta. Socijalni konstruktivizam ne raspravlja o prirodi stvarnosti izvan zajedničkog društvenog i kulturnog svijeta. Osobna iskustva pojedinaca postaju društvena i kolektivna iskustva kada se dijele, komuniciraju, prenose, rekonstruiraju i

zadržavaju kao znanje. Stoga je znanje o svijetu konstruirano iz zajedničkih iskustava, društva i kulture te fizičkog svijeta.

Konstruirano znanje nema jednosmjernan tijek, procesuiranjem se informacije dovode u vezu s postojećim konceptima, prethodno stečenim iskustvom i tek tada dobivaju svoje značenje (Jukić, 2013). Kao polazište učenja i poučavanja uzima se višestruka stvarnost (Murphy, 1997), ovisna o pojedincu i njegovoj konstrukciji subjektivnog iskustvenog svijeta. U odnosu pojedinca i njegovog društvenog i kulturnog miljea, iz kojeg proizlazi i njegov kognitivni razvoj, posebnu ulogu imaju jezik i kultura (Vygotsky, 1978). Jezične sposobnosti omogućavaju čovjeku komunikaciju i sagledavanje svijeta oko sebe uz prevladavanje ograničenja koje mu nameće vlastite percepcija nametanjem društveno definiranog smisla i značenja.

Socijalni konstruktivizam na um metaforički gleda kao na povezanu mrežu pojedinca i društva u kojem djeluje. Komunikacija i interakcija unutar zajednice igraju značajnu kognitivnu ulogu. Na um se ne gleda samo u pojedinačnom kontekstu, već se promatra širi društveni i kulturni kontekst, a konstrukcija novog znanja smatra se društvenim fenomenom. Uloga individualnog uma u konstrukciji značenja vrednuje se u širem kontekstu, u odnosu na druge osobe i okruženje (Belbase, 2011). Um predstavlja društveni entitet koji stvara značenje kroz razgovor, dijalog, interakciju i društvenu i kulturnu razmjenu ideja (Ernest, 1995).

2.3.3.4. Kulturalni konstruktivizam

Kulturalni konstruktivizam sagledava konstruktivističke ideje u širem kontekstu kulturnog i biološkog nasljeđa, običaja, religije, jezika i alata za učenje.

Alati koje koristimo, uključujući jezik, sustav simbola i fizičke alate, utječu na način razmišljanja. Oni raspodjeljuju kognitivno opterećenje zadatka između ljudi i alata. Osim toga, upotreba alata može utjecati na učenika i šire od konteksta trenutnog učenja, mijenjanjem vještina, perspektive i načina predstavljanja svijeta (Salomon i Perkins, 1998). Više mentalne funkcije su, po definiciji, kulturno posredovane. One ne uključuju izravno djelovanje na svijet, već neizravno, ono koje uzima prethodna znanja i primjenjuje ih kao aspekt trenutnog djelovanja. Naglašena je potreba razmatranja kako svijeta područnog znanja, tako i učenikove mentalne slika tog svijeta (Cobern, 1993).

2.3.3.5. Kritički konstruktivizam

Kritički konstruktivizam predstavlja socijalnu epistemologiju koja se bavi društveno-kulturnim kontekstom izgradnje znanja (Taylor, 1996).

Ovaj konstruktivistički pogled promiče stvaranje uvjeta za uspostavljanje dijaloga orijentiranog na postizanje međusobnog razumijevanja. Ovi uvjeti uključuju primarnu brigu za održavanje suosjećajnih i pouzdanih odnosa, opredjeljenje za dijalog čiji je cilj postizanje recipročnog razumijevanja ciljeva, interesa i standarda te briga i kritičku svijest o često nevidljivim pravilima učionice, uključujući društvene i kulturne mitove (Taylor, 1998).

2.3.3.6. Konstruktivizam kao teorijska podloga obrazovnog procesa

Postoje značajne razlike među konstruktivistima različitih pravaca u tumačenju stvaranja znanja. Za radikalne konstruktiviste znanje je predstavlja izgradnju povezane, neproturječne iskustvene stvarnosti pojedinca, neovisnog o društvu. S druge strane konstruktivističkog spektra, socijalni konstruktivisti tumače stjecanje znanja kao izgradnju u društvu dogovorene, tj. društveno konstruirane stvarnosti (Purković, 2016). Neovisno o navedenim teorijskim razlikama, postoji konsenzus u stručnim i znanstvenim pedagoškim krugovima prema kojemu konstruktivistički pristup učenju i poučavanju, neovisan o pravcu, može nadopuniti i oplemeniti obrazovni proces i donijeti mu novu vrijednost.

Usprkos postojećim razlikama, među konstruktivistima različitih pravaca postoji zajednički stav kojim se ističe osam bitnih čimbenika koji utječu na uspješnost obrazovnog procesa (Brooks i Brooks, 1993). Pri tome se ne radi o isključivo konstruktivističkim načelima, već o čimbenicima koje su i drugi teoretičari predlagali tijekom vremena.

- Učenje se treba odvijati u stvarnom, autentičnom okruženju. Stvarno i autentično iskustvo predstavlja bitan poticaj za aktivnosti na osnovi kojih se izgrađuje novo znanje, odnosno konstruira model stvarnosti, neovisno o tome radi li se o osobnom ili konsenzusnom modelu. Izgradnja znanja u svijesti učenika je uspješnija kada je potaknuta autentičnim okruženjem. Konstruktivisti naglašavaju važnost autentičnog okružja za učenje i poučavanje jer se u njemu izgrađuju smislene mentalne strukture, održive u realnim situacijama.
- Učenje treba uključivati posredovanje i socijalno pregovaranje. Znanja i vještine neophodne za potpuni razvoj osobe koja će uspješno zauzeti svoje mjesto u društvu, kao što su, između ostalih, sposobnost društvenog pregovaranja i posredovanja, komunikacijske vještine, pravilna upotreba stručnog jezika i terminologije i vještine suradničkog rada, mogu se steći i vrednovati samo kroz socijalne oblike nastavnog rada, u realnoj socijalnoj sredini ili kvalitetno simuliranoj realnosti.
- Nastavni sadržaji trebaju biti relevantni za učenika. Konstruktivisti ističu adaptivnu funkciju znanja. Izgrađeno znanje treba pomoći prilagodbi učenika za djelovanje u

radnom i socijalnom okruženju. Kako bi se to u potpunosti ostvarilo, učenik treba razumjeti svrhu učenja, potrebu i razlog za usvajanje novih znanja, vještina te društveno prihvatljivih vrijednosti i stavova. Tek tada će nastavni sadržaji i vještine, koje se usvajaju u nastavnom procesu, biti motivirajući za učenika.

- Nove nastavne sadržaje i ciljne vještine treba promatrati iz pozicije učenikovog prethodnog znanja i stečenih vještina. Prema konstruktivističkoj teoriji, izgradnja novog znanja uvijek započinje na osnovama postojećeg znanja i vještina pojedinca. Novo znanje mora se nadograditi na već postojeće znanje, koje bi nastavnik trebao otkriti ili prepoznati. Bitnu ulogu ima i način na koji učenik razmišlja, jer to u velikoj mjeri određuje optimalan put do stjecanja novih znanja. Poznajući ove karakteristike učenika, nastavnik mu može osigurati individualizirani skup iskustava koja će mu dopustiti ispravnu mentalnu konstrukciju odgovarajućeg koncepta (Doolittle i Camp, 1999).
- Ocjenjivanje treba biti formativno kako bi učenicima pružilo informacije važne za buduće iskustveno učenje. Stjecanje znanja i razumijevanje sadržaja je, prema konstruktivistima, proces koji ima svoj tijek, a etape tog procesa u velikoj mjeri ovise o prethodno stečenom znanju učenika. Kako bi mogao osigurati optimalne uvjete za konstruiranje novih znanja kod svakog pojedinog učenika, nastavnik treba kontinuirano vrednovati njihova postignuća, jer znanje i razumijevanje nisu izravno vidljivi iz samog tijeka nastave.
- Učenike treba ohrabrivati na postizanje samoregulacije, samoposredovanja i samosvijesti. Osnovna tvrdnja konstruktivizma je da se znanje i razumijevanje izgrađuje aktivnošću učenika. Te aktivnosti uključuju mentalnu manipulaciju i samoorganizaciju iskustva, zahtijevaju od učenika regulaciju vlastitih kognitivnih funkcija, posredovanje pri stvaranju novih značenja iz postojećih znanja i formiranje svijesti o aktualnim strukturama znanja (Doolittle i Camp, 1999). Samoregulacija, samoposredovanje i samosvijest su elementi metakognicije, tj. samospoznaje o vlastitim kognitivnim procesima (Gordon, 1996), i predstavljaju bitan aspekt konstruktivističkih teorija znanja i obrazovanja (Piaget, 1977; Vygotsky, 1978). Razvoj metakognicije, kao najviše dimenzije znanja, uvijek se javlja kao rezultat konkretne aktivnosti učenika.
- Nastavnik treba biti vodič i organizator učenja, a ne instruktor. Uloga nastavnika je da učeniku ponudi mogućnost uključivanja u praktična istraživanja i pruži mu alate za izgradnju znanja u razrednom okruženju (Alimisis, 2013). Nastavnik prenesenim značenjem nastoji motivirati i potaknuti učenika za rad, i olakšati mu taj rad

odgovarajućim primjerima, raspravom i podrškom. Pri tome on ne pokušava izravno prenositi znanje, odnosno, pružati gotova rješenja (von Glasersfeld, 1998).

- Nastavnik treba osigurati i poticati višestruka stajališta i načine predstavljanja sadržaja. Iskustvo povezano s različitim pogledima na neku pojavu ili raznolikost predstavljanja sadržaja, pruža učeniku osnovi za različita tumačenja, odnosno različite načine konstrukcije znanja i mogućnost razvitka složenih mentalnih struktura). Ne postoje povlaštene istine, već samo uočljive spoznaje koje mogu, u manjoj ili većoj mjeri, dokazati svoju održivost (Doolittle i Camp, 1999). Učenik će bolje razumjeti nastavni sadržaj ako ga može sagledati s više različitih stajališta ili o njemu učiti iz više izvora. Na taj način moći će konstruirati vlastiti održivi modela iskustava i socijalnih interakcija.

Ova načela predstavljaju okosnicu konstruktivističke pedagogije i osnovi na kojoj se treba zasnivati učenje i poučavanje. Poseban naglasak je dan na aktivnu ulogu učenika u nastavnom procesu, bez koje će željeni rezultati nesumnjivo izostati.

Vođeni principom konstrukcije didaktičkog tetraedra, najprije ćemo definirati nastavni sadržaj čije poučavanje želimo ovim istraživanjem oplemeniti suvremenim tehnološkim nastavnim pomagalicama. Odlučili smo se za relativno novo područje agentske paradigme, koja u posljednje vrijeme nalazi sve širu primjenu u različitim granama ljudske djelatnosti, a, prema nekim predviđanjima, njen će značaj će u budućnosti samo rasti.

2.2. Agentska paradigma

Nastavni sadržaj, za koji će biti predložen novi model poučavanja, je agentska paradigma, tj. područje računarstva koje se bavi izučavanjem inteligentnih agenata, te objedinjuje istraživanja u kognitivnoj znanosti i umjetnoj inteligenciji (Wooldridge i Jennings, 1994). Kognitivna znanost pruža model obrade informacija mentalnih procesa koji koriste opisivanje racionalnog ponašanja agenata u funkcionalnom smislu. Područje umjetne inteligencije (engl. *Artificial Intelligence*) pruža tehničke vještine za prevođenje željenih vrsta ponašanja agenata u programski jezik, kao i odgovarajuću sklopovsku i programsku arhitekturu za implementaciju agenta u stvarnom ili u simuliranom svijetu.

Kada govorimo o agentima, ponajprije mislimo na virtualne, programske agente. Takav agent je ustrajan, cilju orijentiran računalni program koji reagira na svoju okolinu i djeluje bez kontinuiranog izravnog nadzora te obavlja određene funkcije za krajnjeg korisnika ili za drugi računalni program u određenom okruženju. Ovakvo samostalno djelovanje podrazumijeva ovlasti za izbor prikladne akcije u određenoj situaciji, ukoliko ona postoji. Agent je inhibiran drugim procesima i agentima, ali je također u stanju učiti iz svog iskustva funkcioniranja u okruženju tijekom dužeg vremenskog razdoblja.

Programski agent je računalni analogon autonomnog robota. Iz ove činjenice proizlazi i kolokvijalni naziv za agente, tj. *Bot*, kao izvedenica riječi robot. Robotsko tijelo, upareno s funkcionalnošću programskog agenta, može predstavljati njegovo otjelovljenje.

Programski agenti mogu biti samostalni ili djelovati u suradnji s drugim agentima ili ljudima. U interakciji s ljudima agenti mogu posjedovati kvalitete karakteristične ljudima, kao što su razumijevanje prirodnog jezika i govora, osobnost ili humanoidno otjelovljenje.

Programski agenti predstavljaju evolucijski korak u odnosu na konvencionalne računalne programe. Oni se mogu samostalno aktivirati i izvoditi te ne zahtijevaju unos ili interakciju s ljudskim korisnikom. Ovi agenti također mogu pokrenuti, nadgledati i prekinuti druge programe, agente ili uređaje u svom okruženju..

Povezani i izvedeni pojmovi su:

- inteligentni agenti koji iskazuju određeni aspekt umjetne inteligencije, npr. učenje ili zaključivanje,
- autonomni agenti koji mogu modificirati način na koji su postižu svoje ciljeve,
- distribuirani agenti koji se izvode na fizički različitim računalima,
- više-agentski sustavi, tj. skupina distribuirani agenata koji rade zajedno kako bi postigli cilj koji pojedinačni agent koji djeluje samostalno ne može ostvariti.

2.2.1. Definicija i osnovna svojstva agenata

Agent, kao središnji pojam agentske paradigme, predstavlja fizički i/ili virtualni sustav koji opaža svoje okruženje pomoću svojih senzora i na njega povratno djeluje preko svojih izvršnih organa, tj. aktuatora. Agenti djeluju u svom okruženju, a ono može uključivati ljude, robote, računalne programe, druge agente, ali i neke jednostavne naprave.

Pojam agenta prvi put možemo susresti u ranim danima istraživanjima distribuirane umjetne inteligencije (engl. *Distributed Artificial Intelligence*), odnosno u modelu *Aktora* (Hewitt, 1977). U ovom modelu, Hewitt predlaže koncept samostalnog, interaktivnog objekta kojeg naziva *Aktorom* i koji se istodobno izvršava sa skupinom istovrsnih objekata. Ovakav objekt ima uključena unutarnja stanja i može odgovoriti na poruke drugih, istovrsnih objekata. *Aktori* predstavljaju računalne agente koja međusobno komuniciraju prosljeđivanjem poruka i mogu istovremeno obavljati svoje aktivnosti.

Danas ne postoji univerzalno prihvaćena definicija pojma agenta. Osobina koja agente izdvaja iz mnoštva umjetnih sustava, bilo da se radi o računalnim programima ili elektromehaničkim robotima, dizajniranim i realiziranim od strane znanstvenika i inženjera je činjenica da agent može, u određenoj mjeri, kontrolirati svoju sudbinu i prilagoditi načela ponašanja (Steels, 1995). Kako bi to bilo moguće, agent mora imati mehanizme koji mu omogućavaju opažanje okoline i djelovanje na nju, ne zahtijevajući intervenciju drugih agenata pri tome. U literaturi nailazimo i na karakterizaciju agenata kao entiteta koji opažaju svoju okolinu sensorima i na nju djeluju putem aktuatora (Russell i Norvig, 2016). Sljedeća definicija promatra agente kao programe koji aktivno sudjeluju u dijalozima, pregovaraju i koordiniraju prijenos informacija (Coen, 1995). Agente se definira i kao sklopovske i/ili programske računalne sustave koji iskazuju svojstva autonomnog djelovanja, društvene sposobnosti, reaktivnosti i proaktivnosti (Wooldridge i Jennings, 1995) ili se nabrajaju slične skupine bitnih karakteristika. Agentima se mogu smatrati računalni sustavi koji nastanjuju složeno dinamično okruženje, osjećaju i djeluju autonomno u tom okruženju, i tako shvaćaju skup ciljeva ili zadataka za koje su dizajnirani

(Maes, 1995). Nwana, u svom često citiranom radu, daje definiciju agenta kao softverske ili hardverske komponente koja je sposobna usmjereno djelovati kako bi izvršila određene zadatke u ime svog korisnika (Nwana, 1996). Ipak, ograđuje se navodeći da je pojam agenta krovni pojam koji obuhvaća cijeli niz specifičnih tipova agenata i preporuča spuštanje preciznije definicije pojma na niži nivo pojedinog tipa.

Svi agenti su računalni programi, ali svi programi nisu agenti. Ključne karakteristike koje razlikuju agente od proizvoljnih programa su: reakcije na okruženje, autonomija, orijentiranost cilju i upornost (Franklin i Graesser, 1996). Agenti se razlikuju od objekata u objektno-orijentiranim računalnim programima svojom autonomnošću i fleksibilnošću, kao i postojanjem vlastite kontrolne strukture. Oni se također razlikuju od ekspertnih sustava klasične umjetne inteligencije izravnom interakcijom s okruženjem, a ne samo obradom simbola koje daju ljudi, kao i njihovim autonomnim učenjem (Florian, 2000).

Ipak postoji konsenzus da je autonomija, tj. sposobnost djelovanja bez ljudske intervencije ili intervencije drugih sustava ključna značajka agenta. Ostali atributi imaju različitu važnost u ovisnosti o domeni djelovanja agenta.

U svom radu Wooldridge i Jennings (1995) iznose dvije uobičajene uporabe pojma agent. Prva je slaba i relativno nesporna, dok je druga jača i potencijalno više sporna.

Najčešći način korištenja pojma agenta, onaj slabi i nesporni, određuje ga kao računalni sustav (obično) zasnovan na softveru koji iskazuje sljedeća svojstva:

- **Svojstvo autonomije.** Autonomni agenti, u najmanju ruku, imaju kontrolu nad svojim unutarnjim stanjem i djelovanjem, tj. djeluju bez izravne ljudske ili neke druge intervencije (Castelfranchi, 1994). Ipak, u praksi, često uzimamo širu definiciju autonomije, koja pretpostavlja da autonomni agent ima svoje uvjerenja, želje i namjere (engl. *Belief, Desire, Intentions* – BDI), koji nisu podložni onima drugih agenata. To ne znači nužno da su te karakteristike zastupljene unutar agenta, nego da svaki agent ima svoj plan rada koji može, ali i ne mora, biti u suglasju s planom rada drugih agenata.
- **Svojstvo reaktivnosti.** Biti reaktivan znači percipirati svoje okruženje, koje može biti fizički svijet, korisnik putem grafičkog korisničkog sučelja, skupina drugih agenata, Internet ili možda sve to kombinirano, i pravovremeno odgovarati na promjene koje se javljaju u tom okruženju. Kako u svakodnevnom životu, tako i svijetu agenata planovi rijetko teku glatko. Često su ometeni zbog ćudljivosti okruženja ili uplitanja drugih. Kad postanemo svjesni da su naši planovi ugroženi, ne zanemarujemo tu činjenicu, nego reagiramo odabirom

alternativnog načina djelovanja. Te reakcije mogu biti jednostavne, tipa *poticaj - reakcija*, ili uključivati daljnje promišljanje. Projektiranje čisto reaktivnog sustava, koji jednostavno reagira na podražaje okoline, nije teško. Takav sustav moguće je realizirati tablicom mogućih reakcija koja preslikava stanja okruženja na akcije sustava za pretraživanje, koja jednostavno okružuje stanja okoliša izravno na akcije. Ovakve sustave često karakteriziramo kao sustave usmjerene ciljem. Složenija je, međutim, implementacija sustava koji postiže učinkovitu ravnotežu ciljanog i reaktivnog ponašanja.

- ***Svojstvo proaktivnosti.*** Djelovanje proaktivnih agenata ne javlja se samo kao odgovor na njihovo okruženje, nego oni mogu iskazati i ponašanje usmjereno prema cilju preuzimanjem inicijative. Štoviše, proaktivnost uključuje i iskorištavanje povoljnih slučajnosti, a isključuje posve pasivne sustave koji nikada ne pokušavaju ništa učiniti.
- ***Svojstvo socijalne sposobnosti.*** Socijalna sposobnost je, s jedne strane, trivijalna jer milijuni računala diljem svijeta svakodnevno razmjenjuju informacije s ljudima i drugim računalima. Međutim, ova sposobnost razmjene niza bitova, u stvari nije socijalna sposobnost. U svakodnevnom životu, malo toga možemo postići bez interakcije s drugim ljudima i organizacijama, za koje se ne može pretpostaviti da dijele naše ciljeve. Analogno stvarnom životu, pojedini agenti su autonomni, svaki sa svojim rasporedom i ciljevima. Kako bi postigli te ciljeve, ponekad moraju pregovarati i surađivati s ostalim agentima, razumjeti i razmišljati o njihovim uvjerenjima i ciljevima i obavljati radnje koje inače ne bi odabrali, kako bi ostvarili suradnju i, putem nje, postigli svoje ciljeve. Ova vrsta socijalne sposobnosti, tj. interakcije s drugim zainteresiranim agentima (i ljudima) putem jezika za komunikaciju agenata, mnogo je složenija i mnogo manje razumljiva nego jednostavna razmjena niza bitova (Genesereth i Ketchpel, 1994).

Za neke istraživače, prema Wooldridgeu i Jenningsu (1995), osobito one čije je središte zanimanja umjetna inteligencija, pojam agent ima jači i specifičniji smisao nego što je gore opisano. Oni smatraju da je agent računalni sustav koji je, pored posjedovanja gore navedenih svojstava, koncipiran ili implementiran korištenjem koncepata koji se češće odnose na ljude. Na primjer, u području umjetne inteligencije se agenti često karakteriziraju pomoću psiholoških pojmova, kao što su znanje, uvjerenje, namjera i obveza (Shoham, 1993).

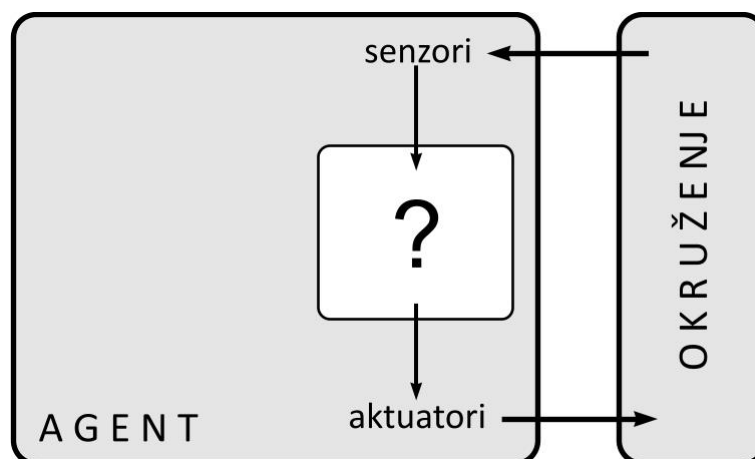
Drugi način davanja obilježja agentima sličnim ljudskima je njihovo vizualno predstavljanje, npr. pomoću grafičke ikone ili animiranog lica (Maes, 1994). Takvi agenti su od posebnog značaja istraživačima zainteresiranim za sučelja čovjek – računalo.

Autori navode nekoliko dodatnih atributa o kojima se raspravlja u kontekstu agenata. Na primjer:

- ***mobilnost*** je sposobnost agenta da se kreće kroz elektroničku mrežu (White, 1994);
- ***istinitost*** je pretpostavka da agent neće svjesno komunicirati lažne podatke (Galliers, 1988);
- ***dobronamjernost*** je pretpostavka da agent nema međusobno suprotstavljene ciljeve i da će, prema tome, svaki agent uvijek pokušati učiniti ono što se od njega traži (Rosenschein i Genesereth, 1985);
- ***racionalnost*** je, pojednostavljeno, pretpostavka da će agent djelovati kako bi postigao svoje ciljeve i neće djelovati na način da spriječi postizanje tih ciljeva – barem u mjeri u kojoj je to moguće.

2.2.2. Matematički model funkcionalnosti agenta

U osnovi, agenta možemo sagledati kroz njegovo opažanje okoline putem senzora i djelovanje u takvoj okolini i na takvu okolinu putem aktuatora. Shematski prikaz ove ideje dan je na Slici 1.



Slika 1. Shematski prikaz agenta

Termin opažaj koristimo za sve perceptivne ulaze agenta tijekom vremena. Slijed opažanja predstavlja cjelokupnu povijest svega što je agent ikada opazio. Izbor akcije kojom agent u nekom trenutku djeluje može ovisiti o posljednjem opažaju ili o cijelom slijedu opažanja. Ako možemo odrediti izbor akcije agenta za svaki mogući slijed opažanja, onda smo u potpunosti definirali njegovu funkcionalnost.

Matematički rečeno, kažemo da je ponašanje agenta opisano agentskom funkcijom koja preslikava bilo koji slijed opažanja u odgovarajuću akciju. Ako s P označimo skup svih

mogućih opažanja, a s A skup akcija koje agent može izvesti onda agentsku funkciju, prema Russellu i Norvigu (2016), možemo zapisati u obliku:

$$f: P^* \rightarrow A,$$

gdje je P^* skup svih sljedova opažanja proizvoljne duljine. Agentska funkcija je apstraktni koncept koji može sadržavati različita načela donošenja odluka kao što su, među ostalim, izračunavanje korisnosti pojedinih opcija, dedukcija na osnovi logičkih pravila i neizrazita logiku (Salamon, 2011).

Ukoliko bismo agentsku funkciju htjeli predstaviti tablicom, radilo bi se o vrlo velikoj tablici. U stvari, ona bi bila beskonačna, osim ako ne ograničimo dužinu slijeda opažanja koji želimo razmatrati. Tablicu je moguće konstruirati isprobavanjem svakog mogućeg slijeda opažanja i bilježenjem akcije koju, kao odgovor na odgovarajući slije opažanja, agent izvodi. Ovakav tabelarni zapis agentske funkcije predstavlja vanjsku karakterizaciju agenta.

Interno, agentska funkcija svakog agenta implementirana je agentskim programom. Pri tome je važno razlikovati ova dva pojma. Agentska funkcija predstavlja apstraktni matematički opis funkcionalnosti agenta, koja u obzir uzima cijeli slijed opažanja. Nasuprot njoj, agentski program je njena konkretna implementacija koja radi na arhitekturi agenta i kao ulaz uzima samo trenutno opažanje, jer ništa više nije dostupno iz okruženja. Ukoliko akcije agenta ovise od cjelokupnoj povijesti opažanja, u tom slučaju agent mora pamtititi opažanja. Agentski program preslikava opažanja, tj. stanje okruženja, na akcije agenta. Općeniti i prilagodljiv pristup izgradnje agentskog programa uključuje stvaranje općenitog tumača pravila stanje-akcija, a zatim i skupa konkretnih pravila za određeni zadatak agenta u okruženju u kom se nalazi. Pri tome, agentski program mora odgovarati stvarnoj arhitekturi agenta, kako bi u potpunosti i optimalno mogao iskoristiti sve njegove mogućnosti.

2.2.3. Klasifikacija agenata

Slično kao i kod definicije agenata, tako i u slučaju njihove klasifikacije, ne postoji konsenzus. U ovom radu odlučili smo se za primjenu klasifikacije koja slijedi.

Russell i Norvig (2016) navode pet osnovnih tipova agenata karakteriziranih stupnjem izraženog inteligentnog ponašanja i sposobnosti:

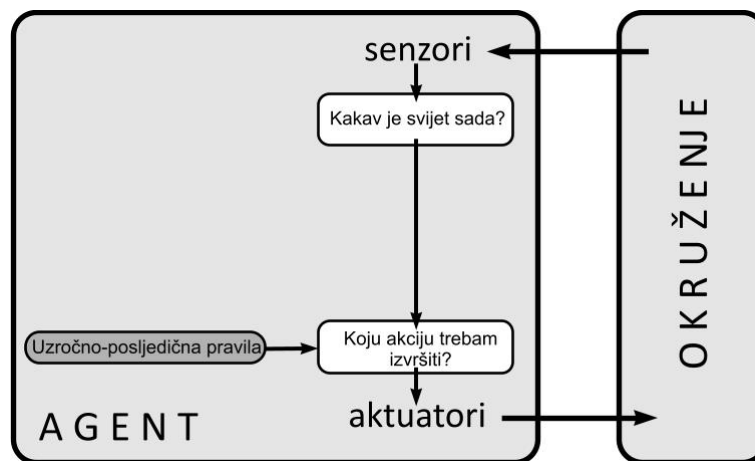
- jednostavni reaktivni (refleksivni) agenti (engl. *Simple Reflex Agents*) koji akciju odabiru na osnovi trenutnog opažanja, zanemarujući povijest opažanja;
- reaktivni agenti sa stanjima (engl. *Model-Based Reflex Agents*) koji akciju odabiru na osnovi povijesti opažanja;

- agenti usmjereni prema cilju (engl. *Goal-Based Agents*) koji odabiru akciju kojom će postići određeni cilj;
- agenti usmjereni prema uspješnosti (korisnosti) (engl. *Utility-Based Agents*) koji biraju akciju koja će postići maksimalni stupanj uspjeha;
- svi ovi agenti mogu biti i agenti koji uče (engl. *Learning Agents*) koji analiziraju iskustvo kako bi izabrali akciju.

U nastavku ćemo detaljnije opisati navedene klase agenta,

2.2.3.1. Jednostavni reaktivni agenti

Jednostavni reaktivni agenti reagiraju samo na trenutni podražaj, ne vodeći računa o prethodnim opažanjima. Funkcionalnost ovih agenata zasniva se na skupu unaprijed definiranih pravila stanje-akcija. Na osnovi trenutnog opažanja agenti traže odgovarajuća pravila. Učinkoviti su u potpuno vidljivom okruženju, tj. okruženju koje agent može u potpunosti spoznati. Shematski prikaz jednostavnog reaktivnog agenta dan je na Slici 2.



Slika 2. Shematski prikaz jednostavnog reaktivnog agenta

Primjer agentskog programa jednostavnog reaktivnog agenta, u pseudokodu, prikazan je na Slici 3. Funkcija *INTERPRET-INPUT*, na osnovi trenutnog opažaja, interpretira stanje okruženja. Ovo stanje, zajedno sa skupom unaprijed definiranih pravila stanje-akcija, služi funkciji *RULE-MATCH* da pronade prvo pravilo u skupu koje odgovara interpretiranom opisu stanja okruženja. Na osnovi odabranog pravila određuje se akcija koju će agent poduzeti, kao rezultat funkcije *RULE-ACTION*.

```

function JEDNOSTAVNI-REAKTIVNI-AGENT (opažaj)
  static: pravila /* skup unaprijed definiranih pravila stanje-akcija */

  stanje ← INTERPRET-INPUT (opažaj)
  pravilo ← RULE-MATCH (stanje, pravila)
  akcija ← RULE-ACTION (pravilo)
  return akcija

```

Slika 3. Pseudokod agentske funkcije jednostavnog reaktivnog agenta

Izvedba ove vrste agenata je jednostavna, ali je njihova inteligencija vrlo ograničena. U djelomično vidljivom okruženju često su u radu agenata ovog tipa neizbježne beskonačne petlje. Mogućnost slučajnog odabira akcije agenta može omogućiti izlaz iz ovakvih situacija.

2.2.3.2. Reaktivni agenti sa stanjima

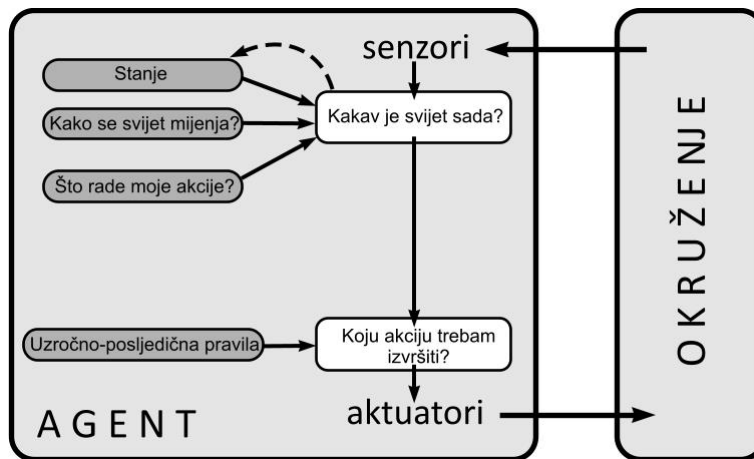
Reaktivni agenti sa stanjima nazivaju se još i reaktivnim agentima zasnovanim na modelu. Ovi agenti mogu uspješno djelovati i u djelomično vidljivom okruženju. Trenutno stanje okruženja je pohranjeno unutar agenta održavajući, na taj način, određenu strukturu, tj. model svijeta, koja opisuje onaj njegov dio koji agent ne može spoznati.

Ovakav interni model ovisi o povijesti opažanja pa se, prema tome, odražava na barem neke od neopaženih aspekata trenutnog stanja. Model zahtijeva dva tipa informacija na osnovi kojih obnavlja interni svijet:

- Kako se okruženje mijenja neovisno o djelovanju agenta?
- Kako djelovanje agenta utječe na okruženje?

Trenutni opažaj, u kombinaciji s prethodno pohranjenim modelom stanja okruženja, generira ažurirani opis trenutnog stanja. Pomoću ovog internog modela može se odrediti povijest opažanja i utjecaj djelovanja agenta na okruženje, te na osnovi njih odabrati djelovanje na isti način kao i kod jednostavnih reaktivnih agenata. Shematski prikaz reaktivnog agenta sa stanjima dan je na Slici 4.

Pseudokod agentskog programa reaktivnog agenta sa stanjima prikazan je na Slici 5. U odnosu na jednostavnog reaktivnog agenta, zanimljivo proširenje agentskog programa je funkcija *UPDATE-STATE*, koja je odgovorna za generiranje ažuriranog unutarnjeg opisa stanja okruženja. Osim što tumači novi opažaj iz perspektive postojećeg znanja o stanju okruženja, ona koristi i informacije o tome kako se okruženje razvija, kako bi se vodilo računa o njegovim dijelovima koji nisu vidljivi, kao i o posljedicama djelovanja agenta na okruženje.



Slika 4. Shematski prikaz reaktivnog agenta sa stanjima

```

function REAKTIVNI-AGENT-SA-STANJIMA (opažaj)
  static: stanje /* opis trenutnog stanja okruženja */
           pravila /* skup unaprijed definiranih pravila stanje-akcija */

  stanje ← UPDATE-STATE (stanje, opažaj)
  pravilo ← RULE-MATCH (stanje, pravila)
  akcija ← RULE-ACTION (pravilo)
  stanje ← UPDATE-STATE (stanje, akcija)
  return akcija

```

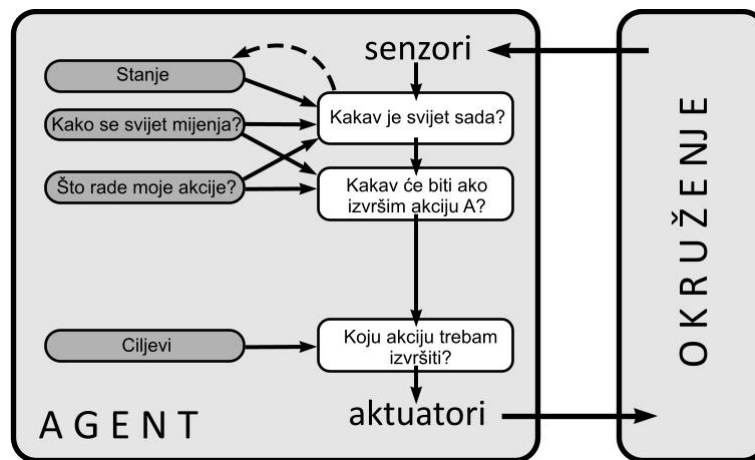
Slika 5. Pseudokod agentske funkcije reaktivnog agenta sa stanjima

2.2.3.3. Agenti usmjereni prema cilju

Agenti usmjereni prema cilju dodatno proširuju mogućnosti reaktivnih agenata sa stanjima, pomoću informacije o "cilju" djelovanja. Ova informacija opisuje poželjne situacije i na taj način omogućava agentu da između više mogućnosti odabere onu koja vodi ka ciljanom stanju. Pronalaženje sekvence djelovanja agenta kojom se postiže zadani cilj ostvaruje se pomoću pretraživanja i planiranja, disciplina umjetne inteligencije. Ovaj tip agenta je fleksibilniji od ranije navedenih jer je znanje koje pomaže pri donošenju odluke eksplicitno predstavljeno i podložno promjeni. Razlika u odnosu na pravila stanje-akcija je da se na ovaj način u obzir uzimaju i budući ciljevi. Shematski prikaz agent usmjerenog prema cilju dan je na Slici 6.

Dakle, osim opisa trenutnog stanja, agentu usmjerenom prema cilju su potrebne i informacije koje opisuju poželjne situacije, tj. ciljane stanja. Agentski program ove vrste agenta kombinira informacije o ciljanim stanjima s informacijama o rezultatima mogućih radnji, koje je agentski program reaktivnog agenta sa stanjima koristio za ažuriranje unutarnjeg opisa stanja okruženja, kako bi odabrao akcije koje postižu cilj. Ponekad će to biti jednostavno, kada zadovoljavajući rezultat proizlazi iz jedne akcije, a ponekad će to biti zahtjevnije, kada agent mora uzeti u obzir

duge sljedove akcija i njihovih posljedica kako bi pronašao način za postizanje cilja. Pseudokod agentskog programa agenta usmjerenog prema cilju dan je na Slici 7.



Slika 6. Shematski prikaz agenta usmjerenog prema cilju

```

function AGENT-USMJEREN-PREMA-CILJU (opažaj)
  static: stanje /* opis trenutnog stanja okruženja */
           pravila /* skup unaprijed definiranih pravila stanje-akcija */
           cilj /* skup definiranih ciljanih stanja */

  stanje ← UPDATE-STATE(stanje, opažaj)
  pravilo ← RULE-MATCH(stanje, pravila)
  akcija ← RULE-ACTION(pravilo)
  stanje ← UPDATE-STATE(stanje, akcija)
  if (stanje in cilj) then
    return akcija
  else
    opažaj ← OBTAIN-PERCEPT(stanje, cilj)
    return AGENT-USMJEREN-PREMA-CILJU (opažaj)

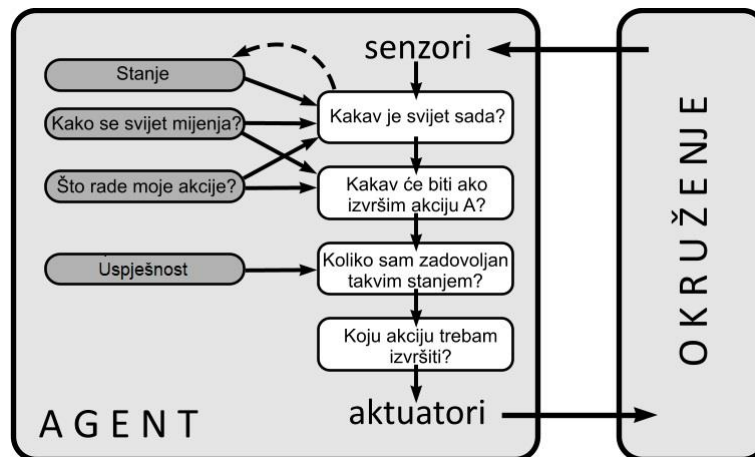
```

Slika 7. Pseudokod agentske funkcije agenta usmjerenog prema cilju

2.2.3.4. Agenti usmjereni prema uspješnosti

Sami ciljevi nisu dovoljni za realizaciju visokog kvaliteta ponašanja agenta u većini okruženja. Ponekad može postojati mnogo sljedova akcija agenta koje će dovesti do cilja, ali su neki od njih brži, sigurniji, pouzdaniji ili jeftiniji od drugih. Agenti usmjereni prema uspješnosti razlikuju ciljana stanja od onih koja to nisu, ali je, također, moguće definirati mjerilo koliko je neko stanje poželjno. To se ostvaruje korištenjem funkcije uspješnosti, koja preslikava pojedino stanje u realan rezultat, tj. mjeru njegove uspješnosti. Funkcija uspješnosti omogućava donošenje racionalnih odluka u dva tipa slučajeva kada samo poznavanje cilja nije dovoljno, tj. kod postojanja konfliktnih ciljeva ili kod postojanja višestrukih ciljeva.

Općenitiji pristup trebao bi omogućiti usporedbu različitih stanja svijeta prema stupnju zadovoljstva agenta što se nalazi u tim stanjima. Racionalni agent usmjeren prema uspješnosti odabire onu akciju koja daje najbolji rezultat, ponderiran vjerojatnošću dolaska do cilja. Odabirom prave funkcije uspješnosti možemo napraviti agenta koji se ponaša racionalno. Shematski prikaz agenta usmjerenog prema uspješnosti dan je na slici 8



Slika 8. Shematski prikaz agenta usmjerenog prema uspješnosti

Koristeći funkciju uspješnosti *OBTAIN-SCORE*, kao i model okruženja u kojem djeluje, agent usmjeren prema uspješnosti mjeri rezultat onih stanja okruženja koja su moguće posljedice njegovih akcija. Nakon toga bira onu akciju koja dovodi do najvećeg mogućeg rezultata uspješnosti. Pseudokod agentskog programa agenta usmjerenog prema cilju dan je na Slici 9.

```

function AGENT-USMJEREN-PREMA-USPJEŠNOSTI (opažaj)
  static: stanje /* opis trenutnog stanja okruženja */
           pravila /* skup unaprijed definiranih pravila stanje-akcija */
           cilj /* skup definiranih ciljanih stanja */

  stanje ← UPDATE-STATE(stanje, opažaj)
  pravilo ← RULE-MATCH(stanje, pravila)
  akcija ← RULE-ACTION(pravilo)
  stanje ← UPDATE-STATE(stanje, akcija)
  rezultat ← OBTAIN-SCORE(stanje)
  if (stanje in cilj) and BEST-SCORE(rezultat) then
    return akcija
  else
    opažaj ← OBTAIN-PERCEPT(stanje, cilj)
    return AGENT-USMJEREN-PREMA-USPJEŠNOSTI (opažaj)

```

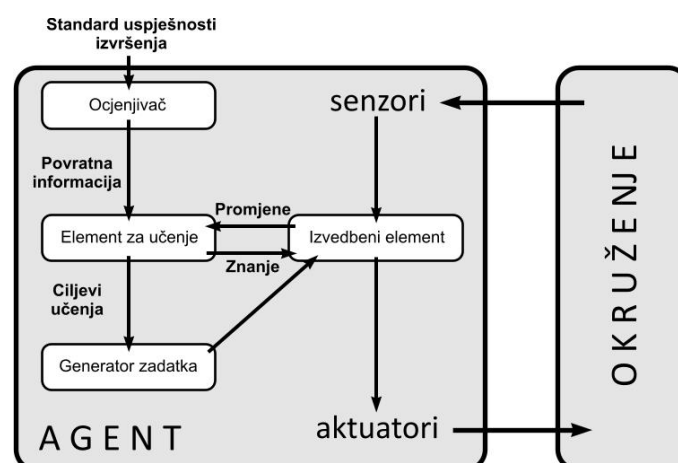
Slika 9. Pseudokod agentske funkcije agenta usmjerenog prema cilju

2.2.3.5. Agenti koji uče

U svom čuvenom radu, Turing razmatra ideju manualnog programiranja inteligentnih strojeva i, zaključujući da za takav način treba previše vremena, predlaže izgradnju strojeva koji mogu učiti (Turing, 1950). Ovakav pristup danas je vrlo zastupljen u mnogim područjima istraživanja umjetne inteligencije. Sposobnost učenja omogućava inteligentnom stroju da u početku djeluje u nepoznatoj sredini, a da vremenom postane sposobniji nego što mu to njegovo početno znanje dopušta.

Agenti koji uče predstavljaju moguću nadogradnju svakog od ranije navedenih tipova agenata. Konceptualno su podijeljeni u četiri komponente.

Najvažnija razlika, u odnosu na prethodne vrste agenata, je element za učenje, koji je odgovoran za poboljšanje funkcionalnosti agenta i izvedbeni elementa, koji je odgovoran za odabir vanjskih aktivnosti agenta. Izvedbeni element, u stvari, predstavlja ono što smo kod prethodnih tipova nazivali samim agentom, on prihvaća opažaje i odlučuje o akcijama. Element za učenje koristi povratne informacije od ocjenjivača o tome koliko dobro agent radi, u odnosu na standard uspješnosti izvršenja, i određuje način na koji se izvedbeni element treba mijenjati kako bi u budućnosti bio uspješniji. Ocjenjivač je neophodan kod ovog tipa agenata, jer opažaji sami po sebi daju indikaciju uspješnosti agenta. Važno je da standard uspješnosti izvršenja bude fiksiran. Konceptualno, treba ga razmatrati kao nešto u potpunosti izvan samog agenta, jer ga agent ne smije prilagođavati svom ponašanju. Posljednja komponenta agenta koji uči je generator zadatka, odgovorna je za predlaganje akcija koje će dovesti do novih i informativnih iskustava. Shematski prikaz agenta koji uči dan je na Slici 10.



Slika 10. Shematski prikaz agenta koji uči

2.2.3.6. Druge klasifikacije agenata

Brenner i sur. (2012) definiraju skupinu karakterističnih svojstava potrebnih za razumijevanje zadataka i funkcioniranja inteligentnih agenata. Ova svojstva čine razliku između inteligentnih agenata i tradicionalnih aplikacija programske podrške. Definirana svojstva su:

- autonomija,
- karakter,
- komunikacija,
- koordinacija,
- mobilnost,
- proaktivnost,
- reaktivnost,
- sposobnost učenja,
- suradnja i
- usmjerenost prema cilju.

Pri tome agent ne mora nužno imati sva navedena svojstva.

Autori ova svojstva grupiraju u dvije kategorije: unutarnja i vanjska.

Unutarnja svojstva su ona koja tvore "unutrašnje biće" agenta, odnosno svojstva koja određuju djelovanje unutar agenta. Ona uključuju svojstva autonomije, mobilnosti, proaktivnosti, sposobnosti učenja i usmjerenosti prema cilju.

Vanjska svojstva uključuju sve one karakteristike koje utječu na interakciju između agenata ili komunikaciju između ljudi i agenata. Na primjer, to su sposobnosti kao što su koordinacije, komunikacije i suradnje.

Pri tome se sva obilježja ne mogu svrstati u samo jednu od skupina, već neka od njih mogu dijelom pripadati objema skupinama. Karakter agenta koji određuje unutrašnje ponašanje agenta, ali također igra i važnu ulogu u vanjskoj komunikaciji, primjer je takvog svojstva.

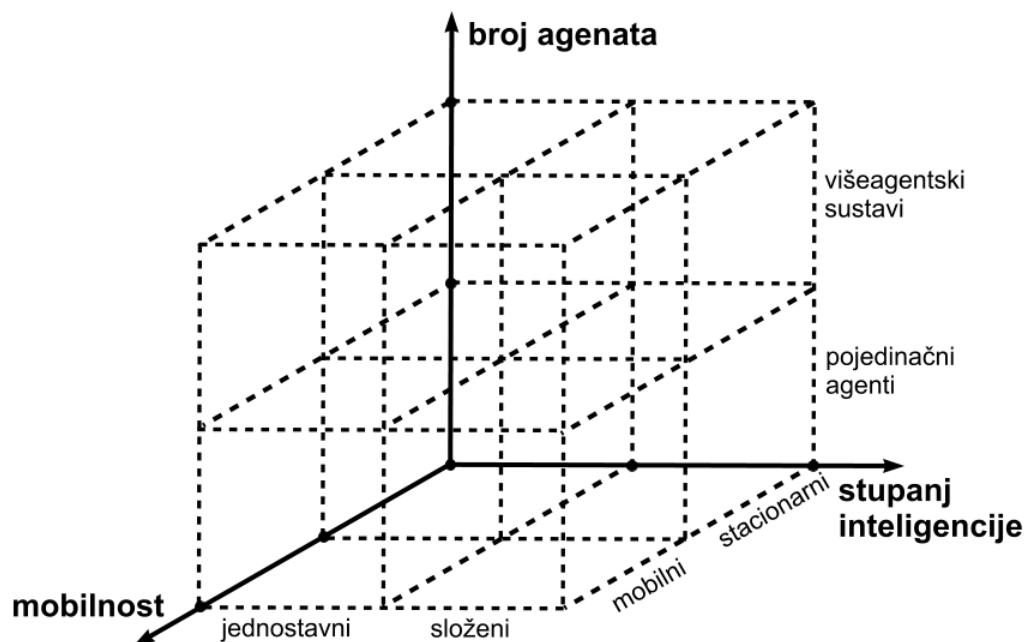
Prema autorima, idealna klasifikacija bi uključivala sva navedena svojstva. Svakom agentu bi se pridružila višedimenzionalna matrica s rezultatima njegovih vrijednosti za svako od svojstava. Međutim, zbog velikog broja svojstava i njihovih mogućih vrijednosti, ovakva matrica bila bi teško razumljiva i ne bi imala praktičnu vrijednost. Zbog toga autori definiraju tri meta-svojstva, svako s dvije moguće vrijednosti, koja uključuju sva ranije navedena svojstva.

Tri meta- svojstva i njihove moguće kategorije vrijednosti su:

- broj agenata (pojedinačni agenti / višeagentski sustavi),
- mobilnost (stacionarni agenti / mobilni agenti) i
- stupanj inteligencije (jednostavni agenti / složeni agenti).

Ranije navedena svojstva karaktera agenta, proaktivnosti, reaktivnosti, sposobnosti učenja i usmjerenost prema cilju u velikoj mjeri određuju stupanj inteligencije agenta. Komunikacija je potrebna kako pojedinačnim agentima, tako i u višeagentskim sustavima, te se može pridružiti kategoriji broja agenata. Sposobnosti koordinacije i suradnje utječu na kriterije stupnja inteligencije i broja agenata, dok autonomija ima utjecaj na inteligenciju i mobilnost agenta.

Agente je moguće klasificirati korištenjem trodimenzionalne matrice, čije dimenzije predstavljaju definirana meta-svojstva. Trodimenzionalna matrica za klasifikaciju agentskih sustava prikazana je na Slici 11.



Slika 11. Trodimenzionalna matrica za klasifikaciju agentskih sustava

Konkretna svojstva svakog pojedinog agenta određuju njegovu poziciju u ovako definiranoj klasifikacijskoj matrici.

Pokušajem klasifikacije agenata, odnosno proučavanjem tipova ovih apstraktnih entiteta programske podrške, u svom radu bavi se i Nwana (1995). On izdvaja nekoliko dimenzija klasifikacije.

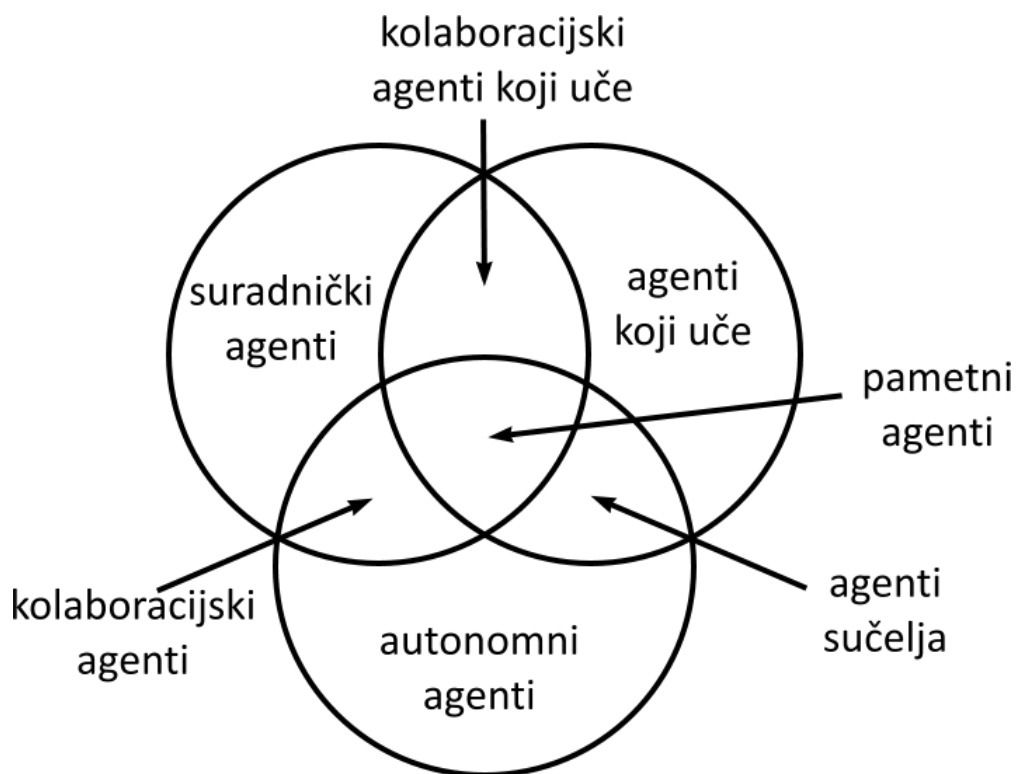
Agente se, prema autoru, može klasificirati na osnovi niza atributa koje bi oni trebali iskazivati. Odabrani broj atributa, naravno, određuje i složenost klasifikacije. Ne umanjujući značaj izbora bilo koje druge skupine atributa, Nwana identificira minimalni popis koji čine:

- autonomija,
- sposobnost učenja i
- suradnja.

Koristeći ovu minimalnu skupinu svojstava izdvojena su četiri tipa agenata:

- kolaboracijski agenti,
- kolaboracijski agenti koji uče,
- agenti sučelja i
- uistinu pametni agenti.

Shematski prikaz ove klasifikacije dan je na Slici 12.



Slika 12. Shematski prikaz klasifikacije agenata na osnovi minimalne skupine svojstava

Pri tome, granice između tipova agenata nisu stroge. Tako je, npr. kod kolaboracijskih agenata, veći naglasak na suradnji i autonomiji nego na sposobnosti učenju. To, naravno, ne implicira da kolaboracijski agenti nikada ne uče. Isto tako je, za agente sučelja, veći naglasak na autonomiji i sposobnosti učenja nego na suradnji. Autor ne smatra agentima ništa što se nalazi

izvan presjeka područja, odnosno ništa što ne pripada jednom od četiri navedena tipa agenata. Kao primjer navode ekspertne sustave koji su većinom autonomni, ali obično ne iskazuju svojstva suradnje i sposobnosti učenja. U idealnom slučaju, agent bi trebao iskazivati sva tri svojstva iz minimalne skupine i to predstavlja cilj kojemu, prilikom realizacije agenata, treba težiti.

Bez obzira na prihvaćenu klasifikaciju i kategoriju u koju svrstavamo konkretnog agenta, zajednička karakteristika svih agenata je da djeluju u okruženju u kojem se nalaze. Postoje različite vrste okruženja koje izravno određuju s kakvim problemima se agent mora nositi. Zato prilikom dizajniranja agenata uvijek u obzir treba uzeti par koji čine agent i okruženje u kojem će djelovati. Iako je ponekad teško odrediti gdje završava okruženje, a počinje agent, korisno je odrediti kriterije okruženja. Na taj način moguće je predvidjeti težinu zadatka za umjetnu inteligenciju.

U nastavku donosimo klasifikaciju okruženja agenata.

2.2.4. Klasifikacija okruženja

Prilikom dizajniranja rješenja umjetne inteligencije koja uključuju agente, obično se najviše vremena posvećuje karakteristikama realiziranih agenata ili prirodi podataka koje će oni prihvaćati i obrađivati. Značajno manje pozornosti posvećuje se prirodi okruženja u kojem dizajnirano rješenje djeluje. Ispostavlja se, međutim, da su karakteristike okruženja jedan od ključnih elemenata za određivanje pravog modela rješenja.

Postoji nekoliko aspekata koji utječu na karakteristike okruženja. Priroda problema, opseg dostupnih informacija i znanja o okruženju, promjene koje u okruženju nastaju djelovanjem agenta samo su neki od elemenata koji određuju prirodu okruženja. Razumijevanje karakteristika okruženja jedan je od prvih zadataka s kojim se treba suočiti kako bi se uhvatili u koštac s određenim problemom iz područja umjetne inteligencije, odnosno agentske paradigme.

Iz te perspektive, postoji nekoliko kategorija koje koristimo za klasificiranje okruženja umjetne inteligencije. Russell i Norvig (2016) uvode sedam načina za klasificiranje okruženja umjetne inteligencije, koje se mogu zapamtiti pomoću mnemoničke sekvence *D-SOAKED*. Svaki od njih dijeli okruženja u dva disjunktna podskupa. Kriteriji klasifikacije su:

- determinističnost (engl. *Deterministicness*), koji okruženja dijeli na deterministička i nedeterministička,
- statičnost (engl. *Staticness*), koji okruženja dijeli na statička i dinamička,

- dostupnost informacija (engl. *Observability*), koji okruženja dijeli na potpuno vidljiva i djelomično vidljiva.
- broj agenata (engl. *Agency*), koji okruženja dijeli na osnovi broja agenata koji u njemu djeluju na jednoagentska i višeagentska.
- znanje o okruženju (engl. *Knowledge*), koji okruženja dijeli na osnovi znanja koje agenti imaju o okruženju.
- epizodičnost (engl. *Episodicness*), koji okruženja dijeli na epizodična i sekvencijalna.
- diskretnost (engl. *Discreteness*), koji okruženja dijeli na diskretna i kontinuirana.

U nastavku ćemo se detaljnije osvrnuti na navedene kriterije klasifikacije.

2.2.4.1. Determinističnost

Za okruženje čije je sljedeće stanje u potpunosti određeno trenutnim stanjem i djelovanjem agenta koji se nalazi u okruženju kažemo da je determinističko. U determinističkom okruženju agent ne treba voditi računa o neizvjesnosti koja je posljedica prirode okruženja. Primjer determinističkog okruženja je šah. U ovoj igri sljedeća pozicija na ploči je u potpunosti određena prethodnom pozicijom i potezom koji igrač, odnosno agent koji igra šah, odigrava.

Ukoliko, međutim, nakon primjene određene akcije agenta postoji neizvjesnost oko stanja koje u okruženju nastaje, kažemo da je to okruženje nedeterminističko ili stohastičko. Ukoliko su neki aspekti okruženja nedostupni agentu, često će takvo okruženje biti stohastičko. To je naročito slučaj kod složenih okruženja. Igra pokera, u kojoj postoji neizvjesnost prilikom podjele karata igračima / agentima, primjer je stohastičkog okruženja

Ako je okruženje agenta u višeagentskom sustavu determinističko, osim za akcije drugih agenata, kažemo da je takvo okruženje strateško.

2.2.4.2. Statičnost

Ako se okruženje ne može promijeniti dok agent djeluje, reći ćemo da je takvo okruženja statičko. Statička okruženja su jednostavna za rukovanje, jer agent ne mora promatrati svijet oko sebe dok odlučuje o sljedećoj akciji, niti treba brinuti o vremenskom tijeku. Filtriranje elektroničke pošte, u potrazi za neželjenom poštom, ili analiza govora su problemi koji se odvijaju u statičkom okruženju.

Okruženje koje se može promijeniti dok agent razmatra sljedeću akciju nazivamo dinamičkim okruženjem. U višeagentskim sustavima učinak drugih agenata može okruženje učiniti

dinamičkim. Primjer dinamičkog okruženja je inteligentni sustav računalnog vida u bespilotnim letjelicama, koji se bavi često promjenjivim podacima,

2.2.4.3. Dostupnost informacija

Ako raspoloživi senzori daju agentu u svakom trenutku potpuni uvid u stanje okruženja u kojem djeluje, tada kažemo da je okruženje zadatka potpuno vidljivo. Okruženje zadatka je potpuno vidljivo ako senzorski aparat agenta može detektirati sve aspekte okruženja relevantne za njegov izbor akcije. Potpuno vidljiva okruženja su pogodna, jer agent ne treba održavati bilo kakvo unutrašnje stanje kojim bi pratio svijet oko sebe. Agent koji obavlja prepoznavanje slike djeluje u potpuno vidljivoj okruženju.

Okruženje može biti djelomično vidljivo zbog nepreciznih senzora ili smetnji u opažanju, ali i zbog dijelova stanja okruženja koje senzori ne mogu opaziti. Agent koji upravlja automatskim usisivačem koji ima samo lokalni senzor prljavštine, ne može reći da li prljavštine u drugim, udaljenim dijelovima prostorije.

U ekstremnom slučaju, kada agent nema uvid u niti jedan relevantan aspekt okruženja, kažemo da je okruženje nevidljivo. U nevidljivom okruženju nije moguće djelovanje agenata.

2.2.4.4. Broj agenata

S obzirom na broj agenata koji u nekom okruženju djeluju, razlikujemo jednoagentska i višeagentska okruženja.

Iako ova razlika može izgledati jednostavna, postavlja se pitanje koji entiteti se moraju promatrati kao agenti. Agentima možemo smatrati one entitete koji se mijenjaju iz koraka u korak. Jača definicija zahtijeva da ti entiteti mogu opažati i djelovati.

Agent koji rješava križaljku djeluje u jednoagentskom okruženju, dok su agenti koji sudjeluju u partiji šaha u dvoagentskom, odnosno višeagentskom okruženju.

Problemi dizajniranja agenata koji se javljaju u višeagentskim okruženjima su često potpuno drugačiji od onih u jednoagentskim okruženjima. U okruženjima u kojima djeluje više agenata, važno je promatrati i njihovo međusobnu komunikaciju i međudjelovanje u odnosu na pojedinačni ili zajednički cilj. Njihov odnos, u takvim situacijama, može biti ravnodušan, suradnički ili natjecateljski. O vrstama međudjelovanja agenata u višeagentskim sustavima će biti više riječi u nastavku.

2.2.4.5. Znanje o okruženju

Strogo govoreći, ovaj kriterij se ne odnosi na samo okruženje, nego na stanje znanja agenta (ili programera koji dizajnira agenta) o zakonitostima koji upravljaju događajima u okruženju. Smatra se da je okruženje poznato ako agent razumije zakone koji upravljaju ponašanjem okruženja. U poznatom okruženju su poznati ishodi ili vjerojatnosti ishoda (u slučaju stohastičkog okruženja) za svaku radnju. Agent koji igra šah će znati da je pojedena šahovska figura uklonjena iz igre. Na ulici, agent koji upravlja vozilom trebao bi znati da kada pada kiša, ulice postaju skliske.

S druge strane, ako agent ne poznaje ili ne razumije zakone koji upravljaju stanjem okruženja, kažemo da je okruženje djelomično poznato ili čak nepoznato. Ako je okruženje nepoznato, agent će morati naučiti kako se ono ponaša, kako bi mogao donositi dobre odluke. Agenti koji uče su pogodni za djelovanje u nepoznatim okruženjima, jer im njihova sposobnost učenja omogućuje da s vremenom postanu kompetentniji nego što im je to dopuštalo samo početno znanje.

Potrebno je naglasiti kako razlika između poznatih i nepoznatih okruženja nije ista kao ona između potpuno vidljivih i djelomično vidljivih okruženja. Sasvim je moguće da je poznato okruženje djelomično vidljivo. Primjer takvog okruženja je igra pasijansa kod kojeg, iako poznajemo pravila, na vidimo karte koje još uvijek nisu okrenute.

Nasuprot tome, nepoznato okruženje može biti potpuno vidljivo. Ekran u nekoj novoj video igri može prikazivati sve karakteristike trenutnog stanja igre, ali igraču još uvijek nije poznato što koja tipka radi, sve dok to ne isproba.

2.2.4.6. Epizodičnost

U epizodičnom okruženju, iskustvo agenta je podijeljeno na osnovne, atomarne epizode. U svakoj epizodi agent opaža svoje okruženje i, na osnovi tog opažaja, izvodi jednu radnju. Pri tome, sljedeća epizoda ne ovisi o akcijama koje su poduzete u prethodnim epizodama, nego samo o informacijama dobivenim putem senzora o stanju u kojem se okruženje nalazi. Mnogi zadaci klasifikacije su epizodni. Na primjer, agent koji mora otkriti neispravne proizvode na traci za montažu zasniva svaku odluku na trenutnom proizvodu bez obzira na prethodne odluke. Također trenutna odluka ne utječe na odluke za sljedeće proizvode.

S druge strane, u sekvencijalnom okruženju, trenutna odluka može utjecati na sve buduće odluke. Agent koji igra šah ili upravlja vozilom djeluje u sekvencijalnom okruženju. U oba

slučaja kratkoročne akcije mogu imati dugoročne posljedice. Epizodična okruženja su mnogo jednostavnija od sekvencijalnih, jer agent koji u njima djeluje ne mora misliti unaprijed niti voditi računa o prethodnim opažajima i akcijama.

2.2.4.7. Diskretnost

Razlika između diskretnih i kontinuiranih okruženja odnosi se na stanja okruženja, način obrade vremena, kao i na opažaje i akcije agenta.

Diskretno okruženje ima konačan broj stanja, iako ponekad vrlo veliki. Vrijeme se kreće u diskretnim intervalima, obično s jednim opažanjem i jednom akcijom po svakom intervalu, iako je moguć i izostanak akcije. Primjer diskretnog okruženja je partija šaha u kojoj postoji konačno stanja na ploči i određeni broj mogućih akcija u svakom potezu.

Kontinuirana okruženja, s druge strane, oslanjaju se na nepoznate i brzo mijenjajuće izvore podataka. Signali okruženja neprekidno. Ovakva okruženja mogu se kvantitativno izmjeriti do željene razine preciznosti. Signali stalno dolaze do senzora agenta, uzrokujući promjene na njegovim postupcima. Sustavi računalnog vida u bespilotnim letjelicama ili samoupravljujućim vozilima rade u kontinuiranom okruženju, s kontinuiranom izmjenom stanja tijekom vremena. Njihova brzina i lokacija prolaze kroz niz kontinuiranih vrijednosti tijekom vremena.

2.2.5. Višeagentski sustavi

Mnoge realne situacije moguće je opisati pojmovima agentske paradigme. Agente možemo zamisliti u gotovo svakom području ljudskog života i djelovanja. Iako agenti mogu djelovati pojedinačno kako bi riješili određeni problem, vrlo često se agentski sustavi sastoje od nekolicine agenata. Većina problema iz stvarnog svijeta zahtijeva ili uključuje više agenata koji predstavljaju decentraliziranu prirodu problema, višestruke perspektive ili konkurentne interese (Krcadinac i sur., 2009). Sustavi u kojima djeluje više autonomnih agenata s mogućnošću interakcije i komunikacije nazivaju se višeagentskim sustavima (engl. *Multi-Agent Systems* - MAS).

Ovi sustavi povijesno pripadaju području distribuirane umjetne inteligencije (Bond i Gasser, 1998), a koriste se za rješavanje složenih problema koji nadilaze mogućnosti svakog od uključenih agenata (Durfée i Lesser, 1989). Osnovne karakteristike višeagentskih sustava su (Jennings i sur., 1998):

- svaki agent ima nepotpune informacije ili sposobnosti za rješavanje problema te, na osnovi tih ograničenja, i vlastiti pogled na problem,

- ne postoji globalna kontrola sustava,
- podaci su decentralizirani,
- računanje je asinkrono.

Dakle, višeagentski sustavi, kao i ranije uvedeni koncept višeagentskih okruženja, uključuju djelovanje više od jednog agenta. Interakcija se događa kada se dva ili više agenata dovode u dinamički odnos kroz skup uzajamnih akcija. Interakcije se razvijaju iz niza akcija čije posljedice utječu na buduće ponašanje agenata. Tijekom interakcija agenti su u međusobnom kontaktu, bilo da je taj kontakt izravan ili se odvija preko drugog agenta ili kroz okoliš (Ferber, 1999).

Interakcije među agentima određene su ponajprije ciljevima i sposobnostima pojedinačnih agenata, kao i raspoloživim resursima u okruženju višeagentskog sustava. Pri tome interakcija može biti pozitivna i negativna.

Pozitivna interakcija je situacija u kojoj djelovanje jednog agenta (ili skupine agenata) ne utječe negativno na djelovanje drugog agenta (ili skupine agenata).

Negativna interakcija je situacija u kojoj uspješno djelovanje jednog agenta (ili skupine agenata) uzrokuje neuspješno djelovanje drugog agenta (ili skupine agenata).

Razlikujemo više vrsta interakcija u višeagentskim sustavima, zasnovane na ciljevima i sposobnostima agenata te resursima dostupnim u okruženju (Ferber, 1999).

Ukoliko su ciljevi agenata u višeagentskom sustavu kompatibilni, a sposobnosti svakog od njih dovoljne za ostvarivanje tih ciljeva te ako u okruženju ima dovoljno raspoloživih resursa za sve agente tada su agenti ravnodušni, odnosno nezainteresirani za djelovanje ostalih agenata u sustavu. Ravnodušnost je situacija u kojoj djelovanje jednog agenta (ili skupine agenata) ni na koji način ne utječe na djelovanje drugog agenta (ili skupine agenata). Ovakav tip interakcije nazivamo neovisnošću.

Suradnju agenata definiramo kao uspješno djelovanje jednog agenta (ili skupine agenata) potpomognuto djelovanjem drugog agenta (ili skupine agenata). Suradnja agenata u višeagentskom sustavu pretpostavlja kompatibilne ili zajedničke ciljeve (Fornara, 2003). U tom slučaju agenti mogu surađivati u pokušaju postizanja cilja koji bi im pojedinačno bio nedostižan (Cavezzali i sur., 2003). Ovo predstavlja situaciju pozitivne interakcije u kojoj dodavanje novog agenta omogućava primjetno povećanje razine izvedbe skupine ili izbjegavanje i/ili rješavanje potencijalnih ili stvarnih sukoba. Pri tome vrsta suradnje ovisi o stupnju međuovisnosti njihovih

aktivnosti, kao i o tome jesu li sposobnosti pojedinih agenata dovoljne za ostvarivanje ciljeva, odnosno postoji li dovoljno resursa u okruženju. Razlikujemo tri vrste suradnje: jednostavnu suradnju, opstrukciju i koordiniranu suradnju.

Nasuprot tome, ako su ciljevi pojedinih agenata ili skupina agenata nekompatibilni dolazi do situacije negativne interakcije (Kwon i Lee, 2002). Pri tome vrsta interakcije ovisi o raspoloživosti resursa. Ukoliko postoji dovoljno resursa u okruženju višeagentskog sustava, agenti se međusobno natječu kako bi ostvarili svako svoj cilj. Natjecanje može biti individualno ili kolektivno, u ovisnosti o sposobnostima koje agenti posjeduju. Ako resursi nisu dovoljni za ostvarivanje ciljeva svih agenata u sustavu, dolazi do sukoba oko raspoloživih resursa (Clement i Durfee, 1999). U ovisnosti o raspoloživim vještinama, sukob oko resursa može biti individualni ili kolektivni.

Kako smo međusobne odnose agenata u višeagentskom sustavu odredili pomoću tri dimenzije (ciljevi, vještine i resursi) od kojih svaka ima dvije moguće vrijednosti, ukupno razlikujemo osam navedenih tipova interakcije u višeagentskim sustavima. U nastavku ćemo dati osnovne karakteristike svakog od njih.

2.2.5.1. Neovisnost

Neovisnost karakteriziraju kompatibilni ciljevi agenata u višeagentskom sustavu i činjenica da svaki od njih ima vještine potrebne za ostvarivanje tih ciljeva. Količina raspoloživih sredstva u okruženju dovoljna je za sve agente.

Situacija neovisnosti ne predstavlja nikakav problem sa stajališta više agenata i može se sažeti kao jednostavan skup usporednih akcija koje agenti provode neovisno jedni o drugima, bez učinkovite interakcije. Ljudi koji prolaze jedni pored drugih na ulici, znajući da ima dovoljno mjesta da se mimoiđu, ili inženjeri koji ne rade na istom projektu, nalaze se u situaciji potpune neovisnosti.

Ovdje se radi o neutralnoj situaciji, koja zapravo ne zahtijeva bilo kakvu specifičnu interakciju sa stajališta umjetne inteligencije, robotike ili distribuiranih sustava.

2.2.5.2. Jednostavna suradnja

Situacija u kojoj je količina raspoloživih sredstava u okruženju dovoljna za sve agente koji u njemu djeluju i kada su ciljevi agenata kompatibilni, ali agenti ne posjeduju vještine potrebne za ostvarivanje tih ciljeva može se riješiti jednostavnom suradnjom agenata u okruženju.

Jednostavna suradnja sastoji se od jednostavnog zbrajanja vještina pojedinih agenata, bez zahtjeva za dodatnim aktivnostima koordinacije među uključenim agentima.

To odgovara situaciji u kojoj se zadaci dodjeljuju članovima tima, ali nitko od članova tima nije dovoljno sposoban da sam izvrši zadatak. Pri tome postoji vrlo malo međuovisnosti među strankama koje surađuju, jer ne postoje potencijalni sukobi oko resursa ili autorizacije (Onken i Schulte, 2010).

Ovakva situacija je karakteristična za komunikacijske sustave u kojima se sva interakcija izražava u obliku raspodjele zadataka i dijeljenja znanja, kao i za tim stručnjaka koji mogu relativno samostalno raditi usporedno ili jedni iza drugih uz međusobno raspoređivanje zadataka i dijeljenje informacija tijekom rada.

2.2.5.3. Opstrukcija

Do opstrukcije dolazi u situacijama u kojima agenti imaju kompatibilne ciljeve i dovoljno sposobnosti za ostvarivanje tih ciljeva, međutim resursi u okruženju nisu dostatni za sve agente u višeagentskom sustavu.

Opstrukcija je karakteristična za sve situacije u kojima agenti ne trebaju jedni druge za realiziranje zadataka, ali utječu na izvedbu ostalih agenata. Do ovog oblika interakcije obično dolazi ako postoje članovi tima koji ovise o istom resursu. Zbog toga, dio suradnje unutar tima je nastojanje pronalaska rješenja s obzirom na koje članovi tima mogu nastaviti djelovanje s najmanjim gubitkom performansi. Dakle, ovdje se radi o tehnikama izbjegavanja i/ili rješavanja potencijalnih ili stvarnih sukoba između agenata u višeagentskom sustavu.

Tipični primjeri su automobili na cesti, kontrola zračnog prometa, optimalno korištenje resursa kao što su raspoređivanje vremena, upravljanje zalihama ili zadavanje zadataka računalnom procesoru.

2.2.5.4. Koordinirana suradnja

U situacijama u kojima su ciljevi agenata u višeagentskom sustavu kompatibilni, međutim niti njihove sposobnosti, niti raspoloživi resursi nisu dovoljni za ostvarivanje tih ciljeva, agenti moraju koordinirati svoje akcije kako bi osigurali sinergijske prednosti udruženih vještina. Također je moguće da, ako se javi potreba, pojedini agenti zamijene svoje uloge.

Koordinirana suradnja djeluje na osnovi unaprijed organizirane i/ili koordinacije u hodu, u svjetlu ograničenih resursa u okruženju. To može uključivati i raspodjelu uloga među članovima tima kako bi se izbjegao konflikt resursa putem organizacijskog dizajna.

Koordinirana suradnja je najsloženija situacija suradnje jer kombinira probleme raspodjele zadataka s aspektima koordinacije koje oblikuju ograničeni resursi.

Koordinirana suradnja javlja se u gotovo svim industrijskim aktivnostima koje zahtijevaju distribuirani pristup, kao što je kontrola mrežnih resursa, projektiranje i proizvodnja industrijskih proizvoda ili distribuirana regulacija ili implementacija čak i vrlo kompliciranih skupina autonomnih robota.

Tipičan primjer tima koji u velikoj mjeri provodi koordiniranu suradnju je pilotska posada od dva čovjeka u zrakoplovu. Dva pilota imaju različite uloge, tj. u svakom trenutku jedan od njih upravlja zrakoplovom i ima nad njim kontrolu, a drugi je odgovoran za pomoć u pripremi plana leta i komunikaciji s kontrolom zračnog prometa. Mogućnosti oba pilota su otprilike iste, tako da u određenim situacijama mogu zamijeniti svoje uloge.

2.2.5.5. Čisto pojedinačno natjecanje

Čisto pojedinačno natjecanje javlja se u situacijama u kojima su ciljevi pojedinačnih agenata nekompatibilni, pojedinačni agenti imaju sposobnosti da ostvare svoje ciljeve, a količina resursa u okruženju je dovoljna za sve agente.

Nekompatibilnost ciljeva pojedinih agenata implicira da ostvarivanje jednog cilja istovremeno uzrokuje otežano, ili čak onemogućeno, ostvarivanje ciljeva drugih agenata. Kada su ciljevi nespojivi, agenti se moraju međusobno natjecati ili pregovarati kako bi ih postigli.

Svi agenti imaju na raspolaganju iste, dostatne, resurse i postavljeni su u identične početne situacije. Kako nema potrebe za natjecanjem oko resursa, dolazi do čistog pojedinačnog natjecanja. Ne postoje specifični problemi vezani uz ovu vrstu situacije, nego pobjeđuje najbolji natjecatelj.

Primjer čistog pojedinačnog natjecanja je atletska utrka. U njoj svi natjecatelji imaju potrebne sposobnosti za dolazak do cilja, kao i resurse koji su im za to potrebni. Pobjeđuje onaj natjecatelj čije su sposobnosti najizraženije.

2.2.5.6. Čisto kolektivno natjecanje

Ako su ciljevi pojedinih agenata u višeagentskom sustavu nekompatibilni, a njihove pojedinačne vještine nedovoljne za ostvarivanje tih ciljeva, pri čemu u okruženju ima dovoljno resursa za potrebe svih agenata, javlja se interakcija koju nazivamo čisto kolektivno natjecanje.

Kada pojedinačne sposobnosti agenata nisu dostatne, oni se moraju grupirati u timove ili koalicije kako bi mogli ostvariti svoje ciljeve. Ovo formiranje skupina provodi se u dvije uzastopne faze. U prvoj fazi agenti teže formiranju skupina ujedinenih vezama koordinirane suradnje, dok u drugoj fazi imaju tendenciju suprotstavljanja ostalim formiranim skupinama.

Karakteristični primjeri ovakve situacije su ekipna natjecanja u kojima se timovi (obično) međusobno ne ometaju, kao što su štafetne utrke ili natjecanje u jedrenju.

2.2.5.7. Pojedinačni sukob oko resursa

Situacija u kojoj imamo nekompatibilne ciljeve pojedinih agenata u višeagentskom sustavu, pri čemu su sposobnosti svakog od njih dovoljne za ostvarivanje pojedinačnog cilja, a zajednički resursi u okruženju ne mogu zadovoljiti potrebe svih agenata dovodi do pojedinačnog sukoba oko resursa.

Kada se resursi ne mogu dijeliti, dolazi do tipičnu konfliktnu situaciju u kojoj pojedinac želi steći resurse samo za sebe, kako bi mogao ostvariti svoj cilj

Postoji velik broj primjera pojedinačnih sukoba oko resursa u životinjskom i ljudskom svijetu. Životinje brane svoju teritoriju ili se međusobno natječu za nju. S druge strane, pojedinci u kolektivu se natječu za financijski povoljan položaj u firmi, što označava i njihov napredak u hijerarhiji.

2.2.5.8. Kolektivni sukob oko resursa

U situaciji u kojoj pojedinačni agenti imaju nekompatibilne ciljevi i nedovoljne pojedinačne vještine potrebne za njihovo ostvarivanje, a pri tome u okruženju ne postoji dovoljno raspoloživih resursa za sve agente višeagentskog sustava, dolazi do kolektivnog sukoba oko resursa.

Ova situacija kombinira kolektivno natjecanje s pojedinačnim sukobima oko resursa. U prvom koraku se formiraju koalicije onih agenata čiji su ciljevi bliski, a suma vještina dovoljna za njihovo ostvarivanje. Nakon toga formirane koalicije se međusobno sukobljavaju kako bi stekle monopol nad resursima, tj. imovinom, teritorijem ili položajem.

Svi oblici rata, lova, industrijske konkurencije i drugih vrsta kolektivnih sukoba u kojima je cilj stjecanje posjeda nad teritorijem, imovinom ili bilo kojim drugim oblikom resursa karakteristični su primjeri ove vrste interakcije.

2.2.6. Primjena agentskih i višeagentskih sustava

Spomenuli smo ranije da je agentska paradigma relativno novo područje istraživanja i primjene umjetne inteligencije. Kako bi se bilo koja nova tehnologija smatrala korisnom ona mora ponuditi jednu od dvije stvari (Jennings i Wooldridge, 1998):

- sposobnost rješavanja problema koji su do tada bili izvan dosega automatizacije, bilo zato što se nijedna postojeća tehnologija nije mogla iskoristiti za rješavanje tih problema ili zato što se razvoj rješenja pomoću postojećih tehnologija smatrao preskupim, teškim, dugotrajnim i/ili rizičnim,
- sposobnost rješavanja problema koji se mogu riješiti i postojećim tehnologijama, ali na značajno bolji, jednostavniji, jeftiniji, učinkovitiji i/ili brži način.

Tehnologija računalnih agenta donijela je nova i efikasnija rješenja problema u različitim granama ljudske djelatnosti, kao i rješenja nekih, do tada, nerješivih problema. Danas se računalni agenti koriste praktično svugdje: u svakodnevnom životu, na poslu, u komunikaciji i zabavi.

Primjene agentske paradigme nalazimo u malim sustavima u kojima obično djeluje jedan agent, a njegov zadatak je relativno jednostavan. U okruženju u kojem agent djeluje rijetko dolazi do promjena i nepredviđenih situacija. Primjeri ovih sustava su:

- filteri elektronske pošte (Alshahad, 2009; Li i sur., 2009; Gomes, 2011),
- pretraživanje Interneta (Jansen, 1997; Stanley, 1997; Godoy i Amandi, 2000; Augar, 2003.),
- osobni asistenti (Wu i sur., 2006; Chen i Barthès, 2007; Pietroszek, 2007).

Nešto složenije uvjete djelovanja agenta nalazimo u srednje složenim sustavima. Razina inteligencije agenata koji u ovim sustavima djeluju je viša nego u prethodnim primjerima, što je uzrokovano i problemima koji se pred njih postavljaju. U ovu kategoriju sustava ubrajamo:

- raspoređivanje zadataka (Das i sur., 2001; Hylton i Sankaranarayanan, 2012; Kumari i sur., 2015)
- novinski servisi i filtriranje informacija (Kebreau i sur., 1998; Bohté i sur., 2000; Al Masum i sur., 2006),
- online kupovina (Lee i Liu, 2004; Xiao i Benbasat, 2007; Shanbhag i sur., 2016),

- posredovanje (Shen i Norrie, 1997; Saber i Okba, 2008; Walker-Roberts i Hammoudeh, 2018),
- Internet aukcije (Wellman, 1998; Maes i sur., 1999; Adomavicius, 2009),
- distribuirano upravljanje projektima (Jennings, 1996; Yan i sur., 2000; Lin i sur., 2015).

Složeni otvoreni sustavi podrazumijevaju visok stupanj inteligentnog ponašanja realiziranih agenata u, najčešće, višeagentskom sustavu. Važnost ispravnog funkcioniranja agenata u kategoriji složenih sustava se ogleda i u razini odgovornosti koje djelovanje ovakvih sustava ima. Primjeri složenih sustava, između ostalih, su:

- sustavi koji pregovaraju i surađuju (Jiao i sur., 2006; Rady, 2011; Kadar i Muntean, 2014),
- kontrola cestovnog, željezničkog i zračnog prometa (Nguyen-Duc i sur., 2003; Chen i Cheng, 2010; Gerla i sur., 2014; Mancilla i sur., 2019; van Wieringen, 2019).

Već je ranije istaknuto kako se agentska paradigma može vrlo učinkovito primijeniti u kombinaciji s još jednom novom, uzbudljivom i nadirućom tehnologijom, koja ubrzano ulazi u sve aspekte ljudskog života i rada – Internetom stvari. Fizički objekti, u potpunosti integrirani u informacijski autocestu, putem svojih senzora prihvaćaju obilje identifikacijskih, povijesnih i opisnih podataka, kao i podataka o položaju i okruženju (Cooper i James, 2009). Dodajući inteligenciju, koju sa sobom donose agenti, svakodnevnim predmetima povezanim na Internet, koncept Interneta stvari dobiva novu, višu dimenziju, a Internet stvari postaje Internet agenata (Yu i sur., 2013; Fortino i sur., 2017).

Integracijom Interneta stvari i agentske tehnologije u proizvodna postrojenja i ova grana ljudskog djelovanja dobiva atribut "pametna" i/ili "inteligentna". Na taj način industrija prolazi kroz četvrtu industrijsku revoluciju, a proizvodni strojevi nadograđuju se suvremenim informacijskim i komunikacijskim tehnologijama te tvore Industriju 4.0 (Adeyeri i sur., 2015; Roblek i sur., 2016; Wang i sur., 2016; Zhong i sur., 2017). Ovaj put čini se neizbježnim u stvaranju suvremene proizvodnje, konkurentne na globalnom svjetskom tržištu.

S obzirom na široku primjenu agentske paradigme u velikom broju grana ljudske djelatnosti, javlja se potreba da se s ovom novom, modernom i sve prisutnijom tehnologijom tijekom svog školovanja upoznaju budući stručnjaci i znanstvenici iz područja informatike, računarstva, strojarstva, prometa, ekonomije i drugih područja ljudskog rada.

Definicijom nastavnog sadržaja, u ovom slučaju područja agentske paradigme, kao prvog vrha didaktičkog tetraedra, u velikoj mjeri određena su i sljedeća dva vrha tetraedra, tj. učenik i nastavnik.

Visoka razina apstrakcije koncepata agentske paradigme, kao i predznanje potrebno za njihovo razumijevanje i aktivno sudjelovanje u nastavi, čini ovo područje primjerenim poučavanju u visokom obrazovanju, naročito na studijima računarstva i informatike.

Nastavnici koji mogu sudjelovati u nastavnom procesu su stručnjaci iz područja računarstva i informatike, s posebnim naglaskom na umjetnu inteligenciju koja čini okosnicu ove suvremene tehnologije.

2.3. Simulacija

Nakon definiranja nastavnog sadržaja, određivanja ciljane skupine studenta kojoj će se sadržaj nastavni predstaviti te nastavnika koji će sudjelovati u nastavnom procesu, ostaje nam definirati i nastavnu metodu koja će se primijeniti. Pokazalo se je da učenici svih uzrasta bolje usvajaju nastavne sadržaje aktivnim sudjelovanjem u relevantnim i autentičnim aktivnostima (McKeachie i Svinicki, 2013), nego kroz pasivno promatranje.

Konkretna iskustva dovode do boljeg razumijevanja novih koncepata, što je posebno izraženo kod visoke razine apstrakcije nastavnog sadržaja (Turkle i Papert, 1992). Ovakav pristup iziskuje visoku razinu uključenosti učenika u nastavni proces (Parsons i Taylor, 2011). Učenici žele iskusiti svoj rad kao smislen, žele raditi s idejama koje su važne i rješavati stvarne probleme (Dunleavy i sur., 2010).

Uključivanjem interaktivnih tehnologija moguće je značajno doprinijeti kvaliteti nastavnog procesa (Ramaley i sur., 2005). Vizualizacija nastavnih sadržaja značajno doprinosi boljem razumijevanju i usvajanju kao i postizanju boljih rezultata učenika, posebno u situacijama kad je prisutna visoka razina apstrakcije sadržaja (Min i sur, 2017).

Problem pri nabavci fizičkih artefakata, tj. tehnoloških nastavnih pomagala, često predstavlja njihova visoka cijena. U kombinaciji s, u pravilu, velikim brojem učenika, ovo postaje značajna prepreka aktivnom sudjelovanju učenika. Problem visokih troškova može se dijelom zaobići korištenjem virtualnih simulacijskih okruženja.

Istraživanja su pokazala da ne postoje značajnije razlike u postignućima učenika koji su bili u kontaktu s fizičkim tehnološkim uređajima i učenika koji su kao nastavno pomagalo koristili simulacijsko okruženje (Liu i sur., 2011; Rutten i sur., 2012; Liu, Newsom i sur., 2013; Brinson, 2015).

2.3.1. Definicija i osnovna svojstva simulacije

U svakodnevnom životu često se susrećemo s konceptom simulacije. Sinoptičari nam redovito prikazuju simulacije vremena u danima koji slijede. Djeca, a ponekad i odrasli, igraju se fizičkim modelima automobila, željeznica, brodova te, u novije vrijeme, i letjelica koji, daljinski upravljani, simuliraju ponašanje stvarnih vozila. Brojne računalne aplikacije simuliraju cijeli niz ljudskih aktivnosti, što nam omogućuje testiranje naših vještina kao vozača, istraživača, urbanista i sl.

S povijesnog stajališta, simulacije korištene u različitim područjima ljudskog djelovanja razvijale su se u velikoj mjeri neovisno. Ipak, istraživanja u području kibernetike i teorije sustava tijekom 20. stoljeća, u kombinaciji sa sve većom primjenom računala u svim tim područjima, dovele su do određenog unificiranja i sustavnog prikaza koncepta simulacije.

Simulacija je danas sve prisutnija metodologija rješavanja problema iz različitih područja ljudskog djelovanja. Koristi se za opisivanje i analizu ponašanja sustava, otkrivanje uzročno posljedičnih veza unutar sustava ali i kao pomoćni alat pri oblikovanju realnih sustava, kako postojećih tako i konceptualnih.

U svom najopćenitijem smislu simulacija se može definirati kao imitacija sustava.

Imitacija označava oponašanje ili kopiranje nečeg drugog. Primjerice, krivotvoritelj oponaša rad velikog umjetnika ili nastoji vjerno kopirati novčane apoene. Za sportaša koji se ponaša kao da je na njemu napravljen prekršaj, iako to nije slučaj, kažemo da simulira. Sustavi računalom potpomognutog dizajna (CAD sustavi) omogućuju imitaciju dizajna fizičkih artefakata ili cjelokupnih proizvodnih pogona. Međutim, ključna razlika između ovih imitacija i ranije opisanih primjera je što oni ne uključuju tijek vremena.

Postoji, dakle, razlika između koncepta statičke simulacije koji oponaša sustav u određenom trenutku i koncepta dinamičke simulacije koja uključuje i vremenski tijek (Law i Kelton, 2000). Izraz simulacija se uglavnom koristi u kontekstu dinamičke simulacije pa ju možemo definirati kao imitaciju djelovanja procesa ili sustava stvarnog svijeta tijekom vremena (Banks i sur., 2001).

Pojedine aspekte ove definicije ćemo dodatno analizirati. Najprije ćemo pojasniti pojam sustava. Općenito, sustav predstavlja skup komponenti, organiziranih s nekom svrhom (Coyle, 1996). Pri tome razlikujemo četiri glavne kategorije sustava (Checkland, 1981):

- Prirodni sustavi su sustavi čije je porijeklo vezano za prirodu oko nas (npr. atom, sustav vremena planete Zemlje, Sunčev sustav, ...).
- Umjetni fizički sustavi su sustavi koji nastaju kao rezultat ljudskog djelovanja, (npr. kuća, automobil, proizvodni pogon, ...).
- Umjetni apstraktni sustavi su oni sustavi koji su rezultat ljudskog djelovanja, (npr. matematika, književnost, ...).
- Sustavi ljudske aktivnosti predstavljaju sustave svjesno ili nesvjesno uređenih aktivnosti ljudi (npr. obitelj, grad, politički sustav, ...).

U praksi je moguće oponašati djelovanje svih navedenih vrsta sustava, tj. simulirati ih. Simuliranje realnog sustava najprije zahtijeva razvoj modela sustava. Model sustava predstavlja ključne karakteristike, ponašanja i funkcije odabranog fizičkog ili apstraktnog sustava ili procesa. Model, dakle, predstavlja sam sustav, dok simulacija predstavlja funkcioniranje sustava tijekom vremena.

Drugi aspekt definicije koji treba dodatno istražiti jest razmotriti svrhu simulacijskih modela. U svojoj općenitoj raspravi o modelima, Pidd kao svrhu izgradnje modela identificira potrebu za razumijevanjem, mijenjanjem, upravljanjem i nadzorom stvarnosti (Pidd, 2009). Govoreći o simulacijskim modelima, kao njihovu svrhu možemo navesti bolje razumijevanje sustava te uočavanje mogućih poboljšanja sustava. Bolje razumijevanje sustava, kao i identifikacija mogućih poboljšanja, važni su za buduće donošenje odluka u stvarnom sustavu.

Druga važna značajka modela je naglasak na pojednostavljenju. Model sustava, naime, rijetko uključuje sve aspekte sustava, nego se koncentrira na one koji su ključni za funkcioniranje sustava. Čak i ako je moguće obuhvatiti sve pojedinosti, to vjerojatno nije poželjno, jer bi zahtijevalo previše vremena za prikupljanje i uključivanje u model svih aspekata sustava.

Prilikom odluke o primjeni simulacije treba uzeti u obzir i svrhu modeliranja sustava. Simulacija predstavlja eksperimentalni pristup modeliranju u kojem korisnik simulacije mijenja ulaze sustava, odnosno istražuje alternativne scenarije, te promatra ponašanje modela dok ne stekne dovoljno razumijevanja o njegovom radu i identificira način na koji može poboljšati realni sustav. Primjerice, moguće je predvidjeti prosječno vrijeme čekanja za telefonske korisnike u pozivnom centru kada je zaposlen određen broj operatera, a zadatak osobe koja koristi simulacijski model je mijenjanje ulaza (broj operatera) i promatranje simulacijskog modela kako bi se utvrdio učinak izmjene.

Uz dodatno opisane bitne aspekte simulacije moguće je preciznije definirati simulaciju kao eksperimentiranje s pojednostavljenom (računalnom) imitacijom promatranog sustava i promatranje njegovog ponašanja tijekom vremena, sa svrhom boljeg razumijevanja i/ili poboljšanja sustava.

2.3.2. Kada primijeniti simulaciju?

Prilikom odluke o eventualnoj primjeni simulacije, potrebno je utvrditi je li ovaj pristup najprimjereniji problemu koji istražujemo. U tu svrhu neophodno je razmotriti karakteristike

promatranog sustava te utvrditi prednosti i nedostatke simulacije u odnosu na ostale metode analize sustava.

2.2.2.1. Karakteristike sustava

Mnogi realni sustavi su podložni varijabilnosti. To mogu biti predvidive (npr. mijenjanje broja operatera u pozivnom centru tijekom dana kako bi se zadovoljio promjenjivi broj poziva korisnika ili planirani prekidi u proizvodnom pogonu) ili nepredvidive promjene (npr. broj dolazaka pacijenata na hitni bolnički prijem ili razgradnje opreme u fleksibilnoj proizvodnoj ćeliji). Oba oblika varijabilnosti prisutna su u većini operacijskih sustava.

Sustavi su također i međusobno povezani, tj. njihove komponente ne funkcioniraju izolirano, već svojim djelovanjem utječu jedna na drugu. Pri tome promjena u jednom dijelu sustava dovodi do promjene u drugom dijelu sustava. Učinke međusobnih veza u sustavu teško je ispravno predvidjeti, posebno kada je prisutna varijabilnost.

Međusobne veze pojedinih komponenti unutar sustava često nisu jednosmjerne, naročito u složenim sustavima. U takvim sustavima često susrećemo strukture petlji ili povratne informacije koje mogu uključivati više komponenti sustava. Pri tome fizički objekti i informacije često putuju u suprotnim smjerovima, čime ponašanje sustava postaje još složenije i teže predvidivo.

Dakle, većina realnih sustava je po svojoj prirodi varijabilna, međusobno povezana i složena. Teško je predvidjeti ponašanje sustava koji su podložni bilo varijabilnosti, međusobnoj povezanosti ili složenosti, a gotovo nemoguće isto napraviti za sustave koji su potencijalno podložni svemu navedenom. Simulacijski modeli, međutim, mogu eksplicitno predstavljati varijabilnost, međusobnu povezanost i složenost sustava. Kao rezultat toga, moguće je simulacijom predvidjeti performanse sustava, usporediti alternativne izvedbe kao i odrediti učinak alternativnih pravila na performanse sustava. Zbog toga je simulacija često pogodan pristup modeliranja sustava.

2.2.2.2. Prednosti simulacije

Simulacija nije jedina metoda analize i istraživanja realnih sustava s ciljem njihova upoznavanja i poboljšanja. Konkretno, moguće je eksperimentirati s realnim sustavom ili koristiti neki drugi pristup modeliranja sustava. Navesti ćemo prednosti koje simulacija ima nad alternativnim pristupima.

Umjesto razvijanja i korištenja simulacijskog modela, eksperimenti se mogu provesti i na realnom sustavu (npr. dodatni prijavni šalteri u zračnoj luci, ...). Prednosti koje simulacija ima nad izravnim eksperimentiranjem s realnim sustavom uključuju (Robinson, 2004):

- Niže troškove. Eksperimentiranje s realnim sustavom je potencijalno skupo zbog potrebe prekidanja ustaljenih postupaka i isprobavanja novih ideja. Dodatno, ukoliko promjene nisu pozitivne, može doći do nezadovoljstva korisnika sustava.
- Kraće vrijeme eksperimentiranja. Izravno eksperimentiranje je vremenski zahtjevno i može potrajati duži vremenski period, dok simulacija može raditi brže od realnog vremena, što daje mogućnost istraživanja mnogih ideja u kraćem vremenskom periodu.
- Mogućnost kontroliranja eksperimentalnih uvjeta. Prilikom istraživanja učinka alternativnih ideja važno je kontroliranje uvjeta u kojima se eksperiment izvodi, kako bi se mogao izravno usporediti njihov učinak. Ovo je teško, ponekad čak i nemoguće, postići kada eksperimentiramo sa stvarnim sustavom, a jednostavno u slučaju eksperimenta sa simulacijskim modelom.
- U situaciji u kojoj realni sustav još ne postoji izravno eksperimentiranje nije moguće. Jedina alternativa je razvijanje vjerodostojnog modela sustava.

Kao što je ranije navedeno, simulacija nije jedini model koji se koristi za bolje razumijevanje i poboljšavanje realnih sustava. Ostali pristupi modeliranja uključuju jednostavne i složene proračune u obliku sustava matematičkih jednažbi, proračunske tablice, složenije matematičke i heurističke metode (linearno programiranje, dinamičko programiranje, genetski algoritmi, ...). Prednosti simulacije u odnosu na druge metode modeliranja posljedica su, u prvom redu, jednostavnijeg modeliranja varijabilnosti sustava i učinka koju ta varijabilnost uzrokuje. Većina alternativnih metoda modeliranja realnih sustava nije u mogućnosti uzeti u obzir i prirodnu varijabilnost realnog sustava koji opisuje, a kod metoda koje se mogu prilagoditi promjenjivoj prirodi opisanog sustava, takva prilagodba značajno povećava njihovu kompleksnost. Druga značajna prednost simulacije je činjenica da ona ne zahtjeva ograničavajuće polazne pretpostavke sustava, ili ih zahtjeva u minimalnom opsegu kao rezultat težnje za pojednostavljenjem modela ili kao nadomjestak za nedostajuće podatke. S druge strane, mnogi drugi pristupi modeliranju strogo zahtijevaju polazne pretpostavke koje su im neophodne za uspješan rad. Na kraju treba dodati i transparentnost simulacije koja je, u odnosu na druge modele modeliranja realnih sustava, intuitivnija i često može stvoriti animirani prikaz sustava što je čini vjerodostojnijom i pouzdanijom, naročito za korisnike koji nisu stručnjaci u modeliranom području.

2.2.2.3. Nedostatci simulacije

Naravno, postoje i slučajevi kada je neki drugi pristup modeliranju sustava primjeren i zadovoljavajući, odnosno slučajevi u kojima simulacija nije potrebna ili prikladna.

Prilikom korištenja simulacija treba voditi računa o nekoliko nedostataka ovog pristupa. Prije svega u obzir treba uzeti visinu troškova razvoja i upotrebe simulacijskog modela, koji mogu biti značajni. Osim toga, simulacija ne predstavlja brzo rješenje nego dugotrajan pristup, što dodatno povećava trošak njezine upotrebe.

Većina simulacijskih modela zahtijeva znatnu količinu podataka koji nisu uvijek dostupni i ponekad je potrebno dosta vremena da ih se transformira u oblik koji je pogodan za simulaciju. Simulacijsko modeliranje zahtijeva vještine i kompetencije konceptualnog modeliranja, validacije i statistike, a u nekim primjenama i vještine rada s ljudima i upravljanja projektima, koje nisu uvijek lako dostupne.

Također, simulacija može biti i pretjerano uvjerljiva zbog upotrebe animiranog, realnog prikaza sustava, što dodatno može osnažiti tendenciju da se računalnom obradom dobiveni rezultati nekritički prihvaćaju. Zbog toga je, prilikom tumačenja rezultata dobivenih simulacijom, nužno razmotriti valjanost osnovnog modela, pretpostavke i pojednostavljenja koja su napravljena.

Zbog svih navedenih nedostataka simulacijskog pristupa, preporuka je da se ona koristi kada druge, jednostavnije, brže i jeftinije, metode ne daju zadovoljavajući rezultat (Pidd, 2004).

2.3.3. Područja primijene simulacije

Područje primjene simulacije obuhvaća različite kontekste u kojima se javlja potreba za optimizacijom performansi promatranih sustava. Od iznimne su koristi u sustavima redova čekanja u kojima se entiteti obrađuju kroz niz etapa, s tim da se između svake etape formiraju redovi čekanja ukoliko sustav nema dovoljan kapacitet obrade. Mnogi sustavi mogu se prikazati kao sustavi redova čekanja, a entiteti koji se kreću kroz sustav pri tome mogu biti ljudi, fizički objekti ili informacije. Redovima čekanja mogu se opisati proizvodni sustavi, transportni sustavi, sustavi zdravstvene zaštite, sustavi prirodnih resursa, upravljanje poslovnim procesima, performanse računalnih sustava i sl.

Simulacija se koristi i za modeliranje prirodnih i društvenih sustava, kako bi se dobio uvid u njihovo funkcioniranje, ispitao i prikazao učinak alternativnih uvjeta ili načina djelovanja. Simulacija se također koristi u situacijama kada se u istraživanje ne može uključiti realni sustav, iz razloga njegove nedostupnosti, opasnog ili neprihvatljivog uključivanja, činjenice da je

sustav projektiran, ali još nije izgrađen, odnosno u situacijama kada je sustav apstraktan ili jednostavno ne postoji (Sokolowski i Banks, 2009).

Ključna pitanja u simulaciji uključuju pribavljanje valjane informacija o ključnim karakteristikama i ponašanju elemenata sustava o odgovarajućem obliku, korištenje aproksimacija i pretpostavki o za jednostavnije modeliranje sustava te provjera točnosti i valjanosti ishoda simulacije. Postupci i protokoli za provjeru i validaciju modela su kontinuirano područje akademskih istraživanja i razvoja tehnologije i prakse simulacije, pogotovo u području računalne simulacije.

2.3.4. Primjena simulacije u obrazovanju

Primjena simulacije u nastavnom procesu pruža učenicima studentima iznimno dragocjenu mogućnost osposobljavanja u realnim situacijama. Njeno korištenje u učionicama eksponencijalno raste u protekla dva desetljeća, u velikoj mjeri zbog napretka u tehnologiji. Ipak, ideja simulacijske obuke javlja se početkom 20. stoljeća, kroz simulacije koje su uključivale replike modela korištene stvarne opreme.

Veći dio napretka u simulacijskom treningu tijekom prošlog stoljeća ostvaren je u zrakoplovnoj industriji sa simulacijom letenja. Godine 1910. stvoren je prvi simulator letenja. Ovaj simulator se sastojao od dvije drvene polovice bačve, od kojih je jedna bila postavljena na postolje, a druga je predstavljala pilotsku kabinu. Pilot je sjedio u gornjoj polovici bačve, koja je ručno pomicala izvana od strane pomoćnog osoblja, kako bi se oponašalo različite scenarije prilikom leta.

Tijekom Drugog svjetskog rata piloti su morali naučiti letjeti u situacijama čestih izmjena okruženja i pri velikim brzinama. Zbog toga su se obavljali simulacijski treninzi na tlu, s cijelim posadama aviona, kako bi naučili letjeti kao tim.

Nakon rata, simulatori letenja su uvedeni i u civilno zrakoplovstvo. Curtiss-Wright iz Pan American Airwaysa proizveo je simulator Boeing 377 Stratocruiser. Ovaj simulator je bio nepomičan i nije imao vizualni sustav, ali je pilotska kabina bila potpuna replika kabine stvarnog aviona. Ovaj simulator se koristio kratko vrijeme, jer su se već šezdesetih godina proizveli simulatori leta koji su mogli pokretati kabinu gore-dolje, oponašajući na taj način turbulencije, polijetanje i slijetanje.

Tijekom sedamdesetih godina počeli su se pojavljivati i računalno generirani simulatori koji su se s vremenom postupno poboljšavali, usporedno s napretkom korištene tehnologija.

Računala se danas koriste kao sastavni dio mnogih učionica i laboratorija. Primjenu nalaze kao alat za učenje i poučavanje, odnosno u realnim i virtualne istraživanjima u cijelom nizu nastavnih predmeta. Koriste se za jednostavnije prikupljanje podataka, za prikaz podataka u stvarnom vremenu, za analizu tih podataka, kao i za simulaciju složenih pojava.

Brojna istraživanja su pokazala da je ovaj pristup nastavi, potpomognut računalima, jednako djelotvoran ili čak djelotvorniji od pristupa koji nije zasnovan na korištenju računala (Steinberg i sur, 1996; Trumper, 2003; Zacharia i Anderson, 2003).

Ova istraživanja su pokazala da računalne simulacije mogu biti jednako učinkovit alat za učenje kao i praktična oprema, s obzirom na iste nastavne planove i programe (Triona i Klahr, 2003). Studenti koji su koristili računalne simulacije umjesto stvarne opreme bolje su odgovarali na konceptualna pitanjima i razvili veću kompetenciju u rukovanju stvarnim fizičkim artefaktima (Finkelstein i sur., 2005).

Kao što je već spomenuto, simulacija se može učinkovito i djelotvorno koristiti u opisivanju sustava koji su nedostupni ili opasni, sustava koji obično ne dopuštaju uključenost učenika, apstraktne sustave ili sustave koji (još) ne postoje. Izučavanje ovih sustava, korištenjem simulacije, može bitno unaprijediti proces učenja i podučavanja.

Uz simulacijska okruženja učenici mogu mijenjaju silu gravitacije, istraživati nuklearnu fisiju na molekularnoj razini, pomicati tektonske ploče dok istražuju razlike između divergentnih i konvergentnih granica. Računalne simulacije daju nam interaktivnu, autentičnu i smislenu mogućnost učenja.

Još jedna prednosti računalne simulacije je mogućnost da se obično nevidljive pojave učeniku jasno predoče i prikažu (Windschitl, 1995). Studenti i istraživači mogu promatrati, istraživati, rekreirati i primati trenutne povratne informacije o stvarnim objektima, fenomenima i procesima koji bi inače bili previše složeni, dugotrajni ili opasni (Dwyer, 1978).

U simuliranoj okolini, tijekom vremena se može ubrzati ili usporiti; apstraktni koncepti mogu se učiniti konkretnim, a prikriivena ponašanja vidljivim. Nastavnici mogu usmjeriti pažnju učenika na konkretne ciljeve učenja, kada je realno okruženje pojednostavljeno, uzročnost događaja jasno objašnjena, a nepotrebni kognitivni zadaci smanjeni kroz simulaciju. U suprotnom, kada učenici imaju mnogo izbora unutar simulacije, sama mogućnost izbora predstavlja potencijalnu distrakciju (Chandler i Chaille, 1993), a previše detalja u vizualizaciji sustava smanjuje mogućnost uočavanja relevantnih detalja (Bell i Smetana, 2008).

Nepotrebne informacije u multimedijским nastavnim sadržajima, tj. riječi, slike i zvukove koji odvrćaju pažnju od bitnog sadržaja, poželjno je eliminirati ili svesti na najmanju moguću mjeru (Clark i Mayer, 2011). Dodavanje zanimljivih, ali nevažnih materijala, može odvratiti pažnju učenika i narušiti proces konstrukcije znanja, a pojačavati emocionalni učinak (Mateljan i sur., 2009). Ovo predstavlja načelo koherentnosti, jedno od načela za smanjivanje obrade suvišnih podataka u multimedijском učenju (Mayer, 2009).

Još jedna korisna značajka simulacije jest mogućnost primjene perceptivnih znakova koji bi usmjerili pozornost učenika na kritične značajke i događaje koji su funkcionalno nevidljivi (Carlsen, Andre, 1992).

Simulacije pružaju mogućnost za razumijevanje i prilagodljivo istraživanje raznolikog stvarnog svijeta i izgrađenih okruženja, čime se pruža širok spektar mogućnosti istraživanja, od znanstvenih do društvenih i umjetničkih. (Woolf, 2010)

Iz svih navedenih razloga, simulacija je u posljednje vrijeme postala jedan od najpopularnijih nastavnih alata za isporuku kvalitetne nastave. Kada se koriste realistične simulacije za proučavanje različitih pojava, studenti se potiču na primjenu stečenih vještina i motiviraju za napredno učenje (Lewis i sur., 1993; Hsu i Thomas, 2002; Weller, 2004; Moreno i Mayer, 2007). Iskustvo rada sa simulacijskim okruženjem pozitivno utječe na ispravno usvajanje koncepata i uspješnu primjenu naučenog materijala (Spinello i Fischbach, 2004; Widiyatamoko, 2018). Studija provedena sa srednjoškolcima u Velikoj Britaniji, koja je koristila simulaciju u učionici, pokazala je da je ovaj pristup inspirirao sudionike da izraze znanstveno razmišljanje i pristup analizi i procjeni prikupljenih podataka (Jarvis i Pell, 2005). Rezultati ovih i sličnih istraživanja ukazuju na veliki potencijal primjene tehnologije u nastavi i njezin utjecaj na postignuća učenika (Jonassen, 2003; Chen i Howard, 2010).

Sudjelovanje u tehnički obogaćenoj nastavi može pozitivno utjecati na stav učenika prema znanosti i potaknuti ih da izaberu STEM-orijentiranu karijeru. Visokotehnološka, računalom podržana kooperativna simulacija realnih situacija pomaže učenicima u primjeni i transferu stečenih znanja i vještina te formiranju njihovog interesa za znanost i profesionalnog identiteta (Chen i Howard, 2010).

Simulacijska okruženja predstavljaju bogati kontekst koji se može koristiti za poučavanje koncepata računarstva, uključujući i one složene i apstraktne (Papastergiou, 2009), a provedena istraživanja su pokazala da njihovo korištenje u nastavi ima pozitivan učinak na usvajanje novih znanja te da se ona mogu iskoristiti kao učinkovita i motivacijska okruženja za učenje (Liu,

Schunn i sur., 2013; Clayton i Gizelis, 2005). Ovakva okruženja pružaju veliku prednost u smislu dostupnosti i praktičnosti zbog financijskih, vremenskih i prostornih ograničenja u kojima se nastava često odvija.

Sustavi u kojima djeluju inteligentni agenti, kao i višeagentski sustavi, primjeri su apstraktnih sustava. Proučavanje tih sustava teško je izvodljivo na teorijskoj razini. Posebna poteškoća je predviđanje ponašanja sustava u različitim situacijama koje inteligentni agent, ili nekolicina agenata u višeagentskom sustavu, opaža svojim sensorima i odgovara na njih.

Postojeća programska rješenja, namijenjena simulaciji djelovanja agenata i višeagentskih sustava (Tisue i Wilensky, 2004b; Vidal i Buhler, 2002; Denzinger i Kidney, 2005), predstavljaju polaznu točku za oblikovanje modela poučavanja agentske paradigme, različitih kategorija agenata te načina njihovog djelovanja i međusobne komunikacije i suradnje zasnovane na primjeni simulacijskih okruženja.

Posljednjih nekoliko godina, uz sve veće zanimanje za korištenje alata za modeliranje i simuliranje zasnovanih na agentima, provedena su brojna istraživanja dostupnih alata. U sljedećem odjeljku dajemo pregled prethodnih relevantnih istraživanja.

2.3.5. Alati za modeliranje i simuliranje zasnovani na agentima

Široka upotreba računalnih agenata koja, uz sveprisutnu digitalizaciju, polako prodire u sve sfere života, također je uzrokovala i potrebu za istraživanjem ponašanja sustava u kojima pojedinačni agenti ili skupine agenata djeluju autonomno. Broj okruženja koja omogućuju modeliranje i simuliranje takvih sustava zasnovanih na agentima stalno se povećava. To je motiviralo brojne autore da provedu istraživanja pregled dostupnih alata, fokusirajući se na različite aspekte modeliranja i simuliranja sustava. Navesti ćemo ta istraživanja kronološki.

U svom pregledu, Serenko i Detlor istraživali su alate za modeliranje i simuliranje sustava zasnovane na agentima koji nalaze primjenu kao pedagoški alati koji se koriste u nastavne svrhe (Serenko i Detlor, 2002). Klasificirali su 20 dostupnih alata promatrajući četiri ključne značajke: sposobnost stvaranja mobilnih agenata, sposobnost razvoja višeagentskih sustava, sposobnost stvaranja različitih tipova agenata za različite svrhe i sposobnost dohvaćanja informacija. Također su istražili osnovni programski jezik koji se koristi u tim alatima. Autori su proveli anketu s 87 nastavnika koji su u svom radu koristili neki od alata. Anketom su ispitali zadovoljstvo korisnika funkcionalnošću alata, njegovom izvedbom i interakcijom s korisnikom.

Godine 2004. Tobias i Hofmann raspravljali su o četiri alata otvorenog koda: *Repast*, *Swarm*, *Quicksilver* i *VSEit*, te ih procijenili na temelju 19 karakteristika, uključujući opće kriterije, modeliranje i eksperimentiranje, podršku modeliranju i mogućnosti modeliranja (Tobias i Hofmann, 2004). Alate su rangirali dodjeljivanjem bodova prema svakom kriteriju. Glavno ograničenje ovog pregleda je činjenica da su autori ograničeni na alate koji koriste Javu kao glavni programski jezik, a uglavnom ih koriste stručnjaci i znanstvenici iz područja društvenih znanosti.

Istraživanje koje su proveli Railsback, Lytinen i Jackson, objavljeno 2006. godine, uspoređuje četiri osnovne platforme: *NetLogo*, *Mason*, *Repast* i *Swarm* (Railsback i sur., 2006). Autori su stvorili predložak, nazvan *StupidModel*, koji su koristili za procjenu i usporedbu tih alata na više razina. Sa svakom novom razinom, dodavali su više mogućnosti i, kroz ukupno 15 razina, istraživali različite značajke kao što su pitanja okoliša, struktura modela, planiranje rasporeda djelovanje agenata, ulaz i izlaz datoteka, generiranje slučajnih brojeva i statističke sposobnosti. Glavno ograničenje ovog pregleda je mali broj testiranih platformi i njihova slična priroda.

U sljedećem istraživanju, provedenom 2006. godine, Castle i Crooks istraživali su osam simulacijskih platformi (Castle i Crooks, 2006). Posebno se usredotočuju na procjenu njihovih geoprostornih mogućnosti. Također su obratili pozornost na datum stvaranja, jezik implementacije, potrebno iskustvo u programiranju te dostupnost demonstracijskih modela i praktičnih vodiča. I ovaj pregled uspoređuje alate specijalizirane za upotrebu u istraživanjima društvenih znanosti.

Značajno povećavajući broj promatranih simulacijskih alata, Nikolai i Madey ispituju pet ključnih karakteristika (Nikolai i Madey, 2009). Kod ukupno 53 alata, uspoređivali su programski jezik potreban za stvaranje modela ili simulacije, operacijski sustav potreban za pokretanje alata, vrstu licence, primarnu domenu za koju je alat namijenjen i stupanj podrške dostupne korisniku. Njihova je namjera bila pružiti korisnicima dovoljno informacija za odabir najprikladnijeg alata u obliku jednostavnog priručnika.

Na temelju osobnog iskustva i dostupnih informacija, Allen se osvrnuo na važne značajke postojećih alata (Allen, 2010). Autor je podijelio alate na simulacijska okruženja pojedinačnih inteligentnih agenata (ukupno 31 alat) i one dizajnirane da simuliraju višeagentske sustave (ukupno 13 alata). Ukazao je na važne nedostatke pojedinih alata, kao što su: složeno korištenje, nedovoljno razvijeni alati za izgradnju modela, posebno alati za predstavljanje prostora, nedostatak alata za izvođenje i promatranje simulacijskih eksperimenata i nedostatak alata za

dokumentiranje i komuniciranje programske podrške. Ovo istraživanje također navodi opća područja primjene ove skupine alata: biologiju i medicinu, fiziku i kemiju, sigurnost i kibernetičku sigurnost, okoliš, društveno i ekonomsko modeliranje te optimizaciju opskrbnih mreža i prometa.

U najopsežnijem pregledu alata za modeliranje i simuliranje zasnovanih na agentima, Abar, Theodoropoulos, Lemariner, i O'Hare se pregledali 85 postojećih alata (Abar i sur., 2017). Autori donese sažetu karakterizaciju postojećih alata, uključujući osnovni programski jezik, tip agenata koje sustav podržava, složenost sustava za krajnjeg korisnika i domenu za koju je sustav prvenstveno namijenjen, naglašavajući njihove prednosti i nedostatke. Kao izvor ovih podataka autorima su poslužili iz korisnički priručnici, popratni vodiči i dokumentacija alata za modeliranje i simuliranje obuhvaćenih ovim pregledom. Na taj način oni pružaju korisnu referencu inženjerima, istraživačima, studentima i nastavnom osoblju za odabir najpogodnijeg alata iz ove skupine, prilikom projektiranja i razvoja njihovih modela i prototipnih sustava.

Pozivajući se na većinu gore navedenih pregleda, Wikipedija nudi još jedan ažurirani pregled dostupnih alata, uključujući njihovu primarnu domenu, instituciju koja ih razvija, programski jezik i vrstu licence programske podrške, korisničku dokumentaciju, 3D i GIS mogućnosti te usklađenost s primjenjivim standardima ("Comparison of Agent-Based Modeling Software", 2019).

Kratak pregled gore navedenih istraživanja dan je u Tablici 1. Ova istraživanja pružaju korisne informacije o širokom rasponu alata za modeliranje i simuliranje zasnovanih na agentima. Oni također daju uvid u razdoblje aktivne upotrebe pojedinih alata, što realno ukazuje na njihovu objektivnu kvalitetu.

Od posebnog značaja za ovo istraživanje su oni alati koji se mogu uspješno koristiti u nastavne svrhe, za upoznavanje studenata s apstraktnim konceptima agentske paradigme.

Tablica 1. Pregled istraživanja alata za modeliranje i simuliranje zasnovanih na agentima

Autori	Godina istraživanja	Broj uključenih alata	Osnovna namjena alata
Serenko, Detlor	2002	20	Obrazovanje i nastava
Tobias, Hoffman	2004	4	Društvene znanosti
Railsback, Lytinen, Jackson	2006	4	Opća namjena
Caste, Crooks	2006	8	Geoprostorne mogućnosti
Nikolai, Madey	2009	53	Opća namjena
Allen	2010	31+13	Opća namjena
Abar, Theodoropoulos, Lemariner, O'Hare	2017-	85	Opća namjena
Wikipedia	2019	25	Opća namjena

2.3.5.1. Pregled ranijih istraživanja

Kako bismo identificirali i kritički se osvrnuli na ranije objavljene radove koji opisuju istraživanja u kojima se analizirala učinkovitost pregledane su relevantne znanstvene baza podataka (Web of Science, ScienceDirect, SCOPUS, Google Scholar). Znanstvene baze pretražene su po pojmovima ključnim za ovo interdisciplinarno istraživanje (Webster i Watson, 2002). U obzir su uzeti pojmovi pojmovi iz područja:

- agentske paradigme: *agent, intelligent agent, agent paradigm, agent theory*,
- simulacijskih okruženja: *simulation, simulation environment, agent based modelling and simulation tools*,
- obrazovanje: *education, teaching, learning*.

Detaljna analiza dobivenih rezultata pokazala je nedostatak istraživanja koji zadovoljavaju sve navedene kriterije pretrage.

Istraživanja koja opisuju primjenu alata za modeliranje i simuliranje zasnovanih na agentima u nastavi odnose se, između ostalih, na poučavanje, koncepata i fenomena prirodnih znanosti: (Stieff i Wilensky, 2003; Kottonau, 2011; Brewington i sur., 2013), društvenih znanosti: (Šperka, 2014; Taillandier i Adam, 2017), primjenjenih tehničkih znanosti (Blikstein i Wilensky, 2006; Blikstein i Wilensky, 2008; Zhu i sur., 2011; Farias i sur., 2019), medicine

(Bora i sur., 2019;) ili čak za poučavanje samog modeliranja i simuliranja (Kurkovsky, 2013; Pinder, 2013; Shiflet i sur., 2013). Može se, dakle, zaključiti da su alati za modeliranje i simuliranje zasnovani na agentima široko prihvaćeni u obrazovanju (Gu i Blackmore, 2015).

Radovi koji opisuju poučavanje agentske ili višeagentske paradigme ne koriste alate za modeliranje i simuliranje zasnovane na agentima (Bordini, 2005) ili ne uspoređuju učinkovitost provedene nastave sa nastavom izvedenom na klasičan način (Berge i sur., 2002; Sakellariou i sur., 2008). Također treba napomenuti da se u ovim istraživanjima radilo o naprednim konceptima agentske paradigme i da su uključeni studenti imali prethodna znanja iz područja agentske paradigme i umjetne inteligencije općenito. Sakellariou, Kefalas i Stamatopoulou proveli su anketu o zadovoljstvu studenata izvedenom nastavom uz korištenje alata NetLogo i dobili pozitivnu povratnu informaciju.

2.3.5.2. Izbor alata za modeliranje i simuliranje zasnovanog na agentima

Prethodno navedena istraživanja i pregledi poslužili su kao osnova za odabir alata za modeliranje i simuliranje zasnovanih na agentima potrebnih za ovo istraživanje. Također su u obzir uzeti i neki alati koji nisu obuhvaćeni ovim istraživanjima, ali se čine zanimljivim i vrlo primjenjivim.

Imajući na umu ograničenja korištenja softverskih alata u nastavne svrhe, posebna pažnja je posvećena nekolicini osnovnih karakteristika svakog od alata, te su alati klasificirani po nekoliko kategorija koje, prema našem iskustvu, igraju važnu ulogu u izboru programske podrške za učenje i poučavanje.

Jedna od kategorija je vrsta licence alata programske podrške. Ova značajka je od velike važnosti zbog financijskog aspekta organiziranja nastave.

Ostale važne kategorije su programski jezik za modeliranje i simuliranje sustava, kao i razvojno okruženje alata. Upoznavanje s programskim jezikom s kojim učenici do tada nisu imali nikakav kontakt bilo bi dugotrajno. Komplicirano razvojno okruženje također bi zahtijevalo previše vremena za prilagodbu učenika na sami alat. Na taj način ne bi preostalo dovoljno vremena za kvalitetno poučavanje agentske paradigme, koja je u središtu pozornosti planiranog nastavnog procesa.

Također smo obratili pažnju na platforme koja podržavaju rad alata ove skupine, kao i na potreban operacijski sustav. Ove karakteristike predstavljaju još jednu značajnu stavku s organizacijskog stajališta.

Kao posljednje obilježje, za svaki od alata ćemo ukazati na razinu složenosti razvoja modela. Ova značajka također ima značajan utjecaj na pogodnost izbora nekog za modeliranje i simuliranje zasnovanog na agentima alata u procesu poučavanja i učenja.

Prije kratkog pregleda odabranih značajki odabranih alata, dat ćemo nekoliko osnovnih informacija za svaki od njih. Alati koje su izdvojeni na osnovi njihove primjenjivosti kao nastavnog alata u procesu učenja i poučavanja su pobrojani u nastavku.

- *AgentScript* je minimalistički okvir za modeliranje sustava, zasnovan na agentima. Ovaj alat pripada skupini alata koji se temelje na *NetLogo* agentskoj semantici. Njegov je cilj promicanje agentski orijentiranog modela programiranja *CoffeeScript/JavaScript* ("Agentscript: Minimalist Agent Based Modeling Framework", 2013).
- *AgentSheets* je autorski alat koji korisnicima računala bez formalne obuke u programiranju omogućuje izgradnju i postavljanje interaktivnih simulacija na webu (Repenning i Sumner, 1995). Ovaj alat kombinira autorizaciju korištenjem Java programskog jezika, programibilne agente namijenjene krajnjem korisniku i tehnologiju proračunskih tablica (Repenning, 2000).
- *BehaviourComposer* je internetski alat za konstruiranje, pokretanje, vizualizaciju, analizu i dijeljenje modela zasnovanih na agentima (Kahn i Noble, 2010). Ovi modeli mogu biti konstruirani od strane korisnika koji nisu stručnjaci u području modeliranja sustava, sastavljanjem unaprijed izrađenih modularnih komponenti nazvanih mikro-ponašanja, koji predstavljaju male, koherentne i nezavisne programske fragmente (Kahn, 2007).
- *Cellular* predstavlja okruženja za programiranje slaganjem unaprijed definiranih ili korisnički definiranih programskih blokova, za stvaranje simulacija zasnovanih na agentima. Mogu se kreirati različite vrste agenata, a njihovo ponašanje se kontrolira konstruiranjem skripti pomoću vizualnog sučelja za programiranje (Lane i sur., 2012).
- *ExtendSim* je simulacijsko okruženje za modeliranje kontinuiranih, diskretnih i sustava zasnovanih na agentima. Njegov dizajn olakšava sve faze projekta simulacije, od stvaranja, procjenjivanja i provjere modela pa do izgradnje korisničkog sučelja (Krahl, 2008). Integrirana baza podataka omogućuje brže i lakše rukovanje simulacijskim modelima (Diamond i sur., 2010).
- *Framsticks* je sustav izgrađen kako bi podržavao širok raspon eksperimenata i pružio svu svoju funkcionalnost korisnicima, koji mogu koristiti ovaj otvoreni sustav na različite načine (Komosiński, 2003). Ovaj alat nastoji popuniti prazninu između često jednostavnih i znanstveno utemeljenih modela umjetnog života, s jedne strane, i

naprednih alata za simuliranje koji realistično prikazuju relativno sofisticirane modele ali koji nisu u potpunosti zasnovani na znanstvenim spoznajama, s druge strane (Komosinski i Ulatowski, 1999).

- *JAS-mine* je računalna platforma temeljena na Javi koja sadrži alate za potporu razvoja velikih, simulacija diskretnih događaja temeljenih na podacima (Richiardi i Richardson, 2017b). Platforma je posebno dizajnirana kako za modeliranje zasnovano na agentima, tako i za mikrosimulacijsko modeliranje, predviđajući konvergenciju između ova dva pristupa (Richiardi i Richardson, 2017a).
- *MIMOSE* je sustav programske podrške za modeliranje i simuliranje koji se sastoji od jezika za opis modela i eksperimentalnog okvira za simuliranje opisanih modela (Möhring, 1995). Svrha projekta *MIMOSE* bio je razvoj jezika za modeliranje koji razmatra posebne zahtjeve modeliranja u društvenim znanostima i podržava stvaranje strukturiranih, homogenih simulacijskih modela (Möhring i Ostermann, 1996).
- *MOBIDYC* je sustav koji pomaže korisniku u četiri glavna koraka pri izgradnji i korištenju modela za dinamiku populacije: definiranju bioloških entiteta, definiranju njihovog okruženja, pokretanju i kontroliranju simulacija te uređivanju rezultata (Ginot i Le Page, 1998). Sustav uključuje alate za svaki aspekt funkcioniranja agenta (Ginot i sur., 2002).
- *NetLogo* je višeagentski programski jezik i okruženje za modeliranje prirodnih i društvenih pojava. Posebno je prikladan za modeliranje složenih sustava koji se razvijaju tijekom vremena (Tisue i Wilensky, 2004a). Unatoč jednostavnom sučelju za programiranje, *NetLogo* je sposoban za vrlo sofisticirano modeliranje i omogućuje iskusnim korisnicima dodavanje vlastitih Java proširenja (Sklar, 2007).
- *Scratch* je vizualno programsko okruženje koje korisnicima omogućuje stvaranje interaktivnih, medijski bogatih projekata (Resnick i sur., 2009). Ti projekti uključuju animirane priče, igre, glazbene spotove, znanstvene projekte, praktične vodiče, simulacije i projekte zasnovane na senzorskim opažanjima (Maloney i sur., 2010).
- *SeSAM* pruža generičko okruženje za modeliranje i eksperimentiranje sa sustavima zasnovanim na agentima (Klügl i Puppe, 1998). Na temelju deklarativne, eksplicitne reprezentacije modela i vizualnog programiranja, ovaj alat omogućuje implementaciju modela na razini specifikacije, uključujući jednostavnu konstrukciju složenih modela (Klügl i sur., 2006).
- *SimSketch* je integrirani alat za crtanje i modeliranje koji studentima omogućuje kreiranje skica i primjenu različitih ponašanja na elemente njihovog crteža (Bollen i van Joolingen,

2013). Mehanizam za simuliranje višeagentskog sustava interpretira i izvršava model, gradeći na taj način intuitivno upotrebljivu i motivirajuću okolinu za učenje i suočavajući učenika s rezultatima i posljedicama njegovog oživotvorenog mentalnog modela (van Joolingen i sur., 2012).

- *SIMUL8* je softverski alat za simuliranje diskretnih događaja koji korisniku omogućuje kreiranje vizualnog modela istraživanog sustava crtanjem objekata izravno na zaslonu (Mc Gregor i sur., 2004). Alat uključuje mogućnosti programskog jezika i vizualizacije modela, što mu omogućuju brže stvaranje točne, fleksibilne i robusne simulacije (Concannon i sur., 2003).
- *StarLogo Nova* je okruženje za programiranje korištenjem blokova instrukcija, koje korisnicima omogućuje stvaranje 3D simulacija i igara (Suzuki, 2017). Ovo okruženje je namijenjeno korisnicima koji istražuju kolektivno ponašanje decentraliziranih agenata. Korisnici mogu po volji slagati definirane i korisničke blokove instrukcija kako bi konstruirali grafičke skripte koje kontroliraju reakcije pojedinačni agenti na podražaje iz okruženja (Pan, 2016)
- *StarLogo TNG* je dizajniran da olakša razvoj i proučavanje simuliranih sustava u učionicama, u okruženju nastavnog procesa. To uključuje grafički programski jezik i trodimenzionalni svijet inspiriran igrom (Klopfer, Scheintaub, Huang, Wendel i Roque, 2009). Grafičko programsko okruženje smanjuje vrijeme potrebno za učenje programiranja u ovom alatu, dok 3D grafika korisnicima pruža upečatljivije i zanimljivije iskustvo (Klopfer, Scheintaub, Huang i Wendel, 2009).
- *Sugarscape* je agentski svijet definiran odozdo prema gore, u kojem su agenti heterogeni s gledišta individualnih sposobnosti (vid) i potreba (metabolizam) (Terna 2001). Alat posjeduje većinu važnih komponenti društvene simulacije. Agenti u *SugarScape*-u imaju niz atributa kao što su vid i metabolizam i sposobni su se prilagoditi različitim okruženjima (D'Souza i sur., 2007).

Pregled ranije istaknutih ključnih značajki onih alata za modeliranje i simuliranja zasnovanih na agentima koji su pogodni za primjenu u nastavnom procesu dan je u Tablici 2.

Tablica 2. Pregled ključnih značajki alata za modeliranje i simuliranje zasnovanih na agentima

ABMS Alat	Tip licence	Programski jezik / API	Operacijski sustav	Složenost razvoja modela
		IDE	Platforma	
AgentScript	LGPL	Biblioteke modela	Windows, Mac OS X, Linux, Unix	Jednostavno / Lako
		CoffeeScripts u okviru pretraživača	Stolno računalo, pametno tablet računalo	
AgentSheets	Komerrijalna	Visual AgenTalk	Windows, Mac OS X, Linux, Unix	Jednostavno / Lako
		Vizualno programsko sučelje	Stolno računalo	
Behaviour Composer (2D/3D)	BSD	Biblioteke modela, Ajax	Windows, Mac OS X, Linux, Unix	Jednostavno / Lako
		Vizualni razvoj modela	Stolno računalo, pametno tablet računalo	
Cellular (2D/3D)	AGPLv3	Grafički blokovi	Windows, Mac OS X	Jednostavno / Lako
		Visual Programming Interface	Stolno računalo	
ExtendSim (2D/3D)	Proprietary	Biblioteke modela	Windows, Mac OS X	Srednje složeno
		Interaktivno grafičko sučelje	Stolno računalo, radna stanica	
Framsticks (2D/3D)	GPL, LGPL	FramScript	Windows, Mac OS X, Linux	Jednostavno / Lako
		Vizualno grafičko sučelje	Stolno računalo, radna stanica	
JAS-mine	LGPL	Ugrađene biblioteke	Svaki OS koji podržava Javu	Jednostavno / Lako
		Eclipse	Stolno računalo	
MIMOSE	GPL	Opis modela	Windows, Linux, Solaris	Srednje složeno
		Grafičko korisničko sučelje	Stolno računalo	
MOBYDIC	GPL	Smalltalk	Windows, Mac OS X, Linux, Unix	Jednostavno / Lako
		Grafičko sučelje zasnovano na predlošcima	Stolno računalo	
NetLogo (2D/3D)	GPL	Biblioteke modela	Windows, Mac OS X, Linux, Unix	Jednostavno / Lako
		N/A	Stolno računalo	

Scratch (2D/3D)	CC-BY-SA 2.0	Grafički blokovi	Svaki pretraživač s Adobe Flash Playerom	Jednostavno / Lako
		Vizualno programsko sučelje	Stolno računalo	
SeSAM	LGPL	Vizualni jezik za modeliranje	Windows, Mac OS X, Linux	Jednostavno / Lako
		N/A	Stolno računalo	
SimSketch	MIT	Java Web Start	Windows, Mac OS X, Linux	Jednostavno / Lako
		N/A	Stolno računalo	
Simul8	Proprietary	Visual Logic	Windows, Mac OS X s VM	Srednje složeno
		Grafičko interaktivno sučelje	Stolno računalo, radna stanica	
StarLogo Nova	CC BY-SA 4.0	Grafički blokovi	Windows, Mac OS X, Linux, Unix, Solaris	Jednostavno / Lako
		Vizualno programsko sučelje	Stolno računalo	
StarLogo TNG	SLTNG v1.0	Grafički blokovi	Windows, Mac OS X	Jednostavno / Lako
		Vizualno programsko sučelje	Stolno računalo	
Sugarscape	GPL	Java	Svaka platforma koja podržava Javu	Jednostavno / Lako
		Vizualni razvoj modela	Stolno računalo	

Prilikom izbora dva alata koja su se koristila u prvom dijelu istraživanja u obzir su uzeti samo oni alati koji su se mogli besplatno koristiti. Također je vođeno računa o korištenoj platformi i potrebnom operacijskom sustavu, kako bi se alati mogli koristiti u okruženju učionica u kojima se nastava odvijala. Zbog vremenskog okvira u kojem se istraživanje provodilo, kao i želje da se većina vremena posveti usvajanju pojmova iz područja agentske paradigme, prednost su imali alati koji su koristili jednostavnije razvojno okruženje i koji su omogućavali lakši razvoj modela. Programski jezik s kojim su studenti od prije bili upoznati ili koji je jednostavan za osnovno korištenje, također je predstavljao prednost prilikom odabira alata.

Vodeći računa o navedenim karakteristikama simulacijskih okruženja, iz skupine ranije navedenih alata odabrani su dva alata različite razine složenosti: jednostavniji *Cellular* i nešto složeniji *NetLogo*.

Cellular je programsko okruženje koje omogućava izradu simulacija utemeljenih na agentima. Upotrebom postojećih blokova ili izradom novih moguće je kreirati različite vrste agenata, nadzirati njihovo kretanje i ponašanje u virtualnom okruženju te promatrati njihovu međusobnu interakciju i interakciju s okruženjem. Izrazito intuitivno vizualno okruženje i način programiranja čini ovaj alat primjeren i za mlađe dobne skupine učenika i studenata. Dodatnu pogodnost pruža i činjenica da je ovaj alat inačica alata, široko korištenog u poučavanju programiranja na svim obrazovnim razinama.

NetLogo predstavlja programski jezik i integrirano programibilno višeagentsko okruženje namijenjeno modeliranju kompleksnih sustava koji se razvijaju tijekom vremena. Moguće je upravljanje i praćenje ponašanja velikog broja neovisnih agenata pod različitim uvjetima. NetLogo je namijenjen studentima i nastavnicima za upotrebu u okviru nastave, ali i istraživačima u mnogim područjima. Sintaksa slična programskom jeziku Logo (Pea, 1987), s kojim se većina studenata već susrela, prilagodljivo interaktivno vizualno okruženje i bogata baza dostupnih modela odigrali su ključnu ulogu kod odabira ovog alata.

2.4. Modeli poučavanja

Umijeće poučavanja često se definira kao oblikovanje i upravljanje obrazovnim okruženjem i iskustvom osobe koja uči sa svrhom poticanja i ostvarenja željenih ishoda učenja. Poučavanje i učenje o poučavanju zahtjevni su zadaci jer se usredotočuju na složene, međusobno povezane skupove misli i djelovanja, kojima se može pristupiti na više načina. Stoga, u poučavanju, ne postoji nužno jedan način da se nešto učini. Kako nastavnik usavršava svoju vještinu poučavanja, potpunije razumijevanje odnosa između poučavanja i učenja može utjecati na njegovu praktičnu izvedbu (Laughran, 1996).

Temeljna pitanja koja zaokupljaju pažnju nastavnika u nastavnom procesu opisana su s tri razine konceptualizacije problema nastavnika (Fuller, 1969., Borich i Fuller, 1974.). Nastavnikova usredotočenost na probleme koji se odnose na njegove sposobnosti rukovođenja skupinom učenika koje poučava, ovladavanja gradivom koje poučava te ocjenom njegovog rada, najčešće se javlja se u početku njegovog nastavnog rada. Stjecanjem nastavnog iskustva i samopouzdanosti, koju to iskustvo nastavniku donosi, opada zabrinutost za pitanja koja se tiču sposobnosti nastavnika da obavlja svoj posao. Zamjenjuju ih problemi vezani za izvođenje nastave, odnosno pitanja izbora nastavnih metoda, materijala koji će se u nastavi koristiti, kao i usklađivanja različitih zahtjeva poučavanja. Nastavnik je na obje razine usmjeren na sebe, svoje sposobnosti i svoj rad. Usmjerenost na učenike, tj. pitanja utjecaja nastave na učenika, emocionalnih potreba učenika i odnos prema učenicima kao pojedincima, predstavljaju treću razinu problema i smatraju se zrelijima od pitanja koja se odnose na nastavnika i izvođenje nastave (Fuller i Bown, 1975.). Ova pitanja dolaze u središte pozornosti kada je nastavnik razriješio dileme koje je imao na prethodnim razinama (Staton, 1992.).

Bez obzira na gradivo koje se poučava, instituciju u kojoj se nastava izvodi i učenike kojima je poučavanje namijenjeno, uspjeh nastavnika u suvremenom dobu, u najvećoj mjeri ovisi o njegovoj sposobnosti da dizajnira učinkovitu nastavu za one koje poučava. Takva nastava mora biti prilagođena individualnim i zajedničkim potrebama šarolike populacije studenata, koji pokušavaju ovladavati nastavnim sadržajem i vještinama potrebnim za život u globalnom tehnološkom društvu. Suvremeni nastavnici okruženi su velikim brojem različitih, poticajnih i moćnih nastavnih sredstava. Međutim, ti resursi, bilo da se radi o nastavnim materijalima, modelima, strategijama ili tehnologijama, djelotvorni su samo ako nastavnici znaju kada i kako ih koristiti (Kilbane i Milman, 2014).

Model poučavanja definiramo kao plan koji se koristi za oblikovanje nastavnog plana i programa predmeta ili cijelog studija, za odabir relevantnih nastavnih materijala te kao vodič kroz aktivnosti nastavnika. Ovi modeli se sastoje od smjernica za oblikovanje obrazovnih aktivnosti i okruženja u kojem se proces učenja i poučavanja odvija. Njima su propisane strategije poučavanja osmišljene kako bi se ostvarili određeni obrazovni ciljevi.

Model poučavanja je način izgradnje ugodnog i poticajnog ekosustava unutar kojeg učenici uče interakcijom s njegovim komponentama (Joyce i sur., 2015). Različiti modeli poučavanja pomažu učenicima u usvajanju znanja i stjecanju vještina i kompetencija te povećavaju njihovu sposobnost osobnog napretka u društvenom i akademskom području.

Ovakav koncept poučavanja mijenja tradicionalnu, ali već pomalo zastarjelu sliku obrazovanja u kojoj je nastavnik osoba koja prenosi znanja i vještine pričajući, potičući i trenirajući učenike određenim tempom, na iscrpljujući način, pri čemu se učenici zatrpavaju informacijama na isti način, bez obzira na njihova predznanja, sklonosti i mogućnosti.

Srećom, kao i ostalim područjima ljudskog djelovanja, tako i u poučavanju dolazi do razvoja i primjene novih saznanja i spoznaja što ovu aktivnost osuvremenjava i čini ju učinkovitijom i primjerenijom modernom dobu.

Danas postoje brojni modeli za dizajniranje i isporuku dobrih predavanja, motiviranje učenika i provođenje učinkovite obuke. Naravno da postoje i situacije kada je potrebno i primjereno koristiti tradicionalne načine isporuke nastavnih sadržaja, ali bismo pri tome trebali koristiti najbolje dostupne načine za dizajniranje iskustava učenja, biti svjesni svrhe nastavnog sata te koristiti pristup koji tu svrhu može ispuniti.

Razvoj modela poučavanja nije ništa drugo nego instrukcijski dizajn. Instrukcijski dizajn strukturira tijek obrazovnog procesa, pri čemu tehnologija nije u prvom planu, već su to rezultati učenja i poučavanja, kao i zadovoljstvo učenika. Instrukcijski dizajn se može promatrati kao primijenjena didaktika i metodika u oblikovanju obrazovnog procesa.

U ovom procesu primjenjuju se pedagoške teorije i teorije učenja kao i različiti principi, tehnike i metode izvođenja nastave. Značaj ovog procesa je u planiranju i izradi nastavnih materijala, te oblikovanju procesa učenja i poučavanja kako bi se u nastavnom okruženju potakla interakcija učenika koja će dovesti do specifične promjene u njegovom ponašanju.

Razvoj i oblikovanje modela poučavanja ima za cilj pomoći nastavniku da dobije širok raspon mogućih pristupa za stvaranje odgovarajućeg interaktivnog okruženja pogodnog za učenje. Inteligentno korištenje ovih pristupa omogućava učitelju da ih prilagodi potrebama učenika.

Modeli poučavanja posebno dobivaju na važnosti zbog sve većeg nastojanja da se u obrazovnom procesu postigne visoka razina kvalitete.

Dobro definiran model poučavanja treba sadržavati (Aggerwal, 2014):

- specifikaciju ishoda učenja kojom će se taksativno navesti što sve učenici trebaju moći izvesti nakon završenog nastavnog ciklusa;
- specifikaciju okruženja, tj. precizno određenje karakteristika okruženja u kojem se učenikova reakcija treba promatrati;
- specifikaciju kriterija izvedbe koja se očekuje od učenika;
- specifikaciju operacija, tj. mehanizama koji osiguravaju reakciju učenika i njegovu interakciju s okruženjem;
- znanstveni postupak koji predstavlja sustavno osmišljenu proceduru kojom se modificira ponašanje učenika, a ne slučajnu kombinaciju činjenica.

Postupak provedbe bilo kojeg nastavnog modela poučavanja sastoji se od definiranja šest osnovnih elemenata (Joyce i sur., 2015).

- **Fokus** je središnji aspekt modela poučavanja koji obuhvaća ciljeve nastavnog procesa i aspekte okruženja u kojem se on odvija. Fokus predstavlja jezgru nastavnog modela, a njegovi ciljevi ujedno određuju i način njegovog razvoja. Modeli mogu imati različite faze, te razvijati neke specifične kompetencije kod učenika.
- **Sintaksa** modela opisuje model u akciji, tj. predstavlja plan djelovanja na opisan način. Ona uključuje sustavan slijed aktivnosti u modelu. Svaki model ima različit tijek faza koje uključuju nastavne aktivnosti i interakciju između učenika i nastavnika. Sintaksa nastavnog modela određuje one točke koje proizvode aktivnosti usmjerene k obrazovnim ciljevima u različitim fazama. Elementi sintakse modela poučavanja čine slijed koji omogućavaju lakše postizanje nastavnih ciljeva pružanjem poželjnog okruženja.
- **Društveni sustav** opisuje uloge nastavnika i učenika, kao i njihove međusobne odnose. Međusobni odnosi nastavnika i učenika mogu biti izrazito strukturirani (uloga nastavnika je dominantna), umjereno strukturirani (uloge nastavnika i učenika su ravnopravne) ili slabo strukturirani (učenik dominira u aktivnostima učenja). Osim toga društvenim sustavom je određen i tip okruženja u kojem se učenje odvija. Ovaj element se zasniva na pretpostavci da je svaki razred minijaturno društvo. U njemu se raspravlja i o odabiru motivirajuće strategije i taktike za učenike. Društveni sustav zauzima središnje mjesto u uspješnoj realizaciji prethodno odabranih ciljeva.

- **Načela reakcije** odnose se na ponašanje nastavnika u razredu, tj. njegove reakcije na aktivnosti učenika. Nastavnik treba znati kako reagirati na učeničke odgovore i utvrditi jesu li učenici aktivno uključeni u nastavni proces. U nekim modelima poučavanja nastavnici otvoreno pokušavaju oblikovati ponašanje učenika nagrađivanjem određenih aktivnosti i izražavanjem neutralnog stava prema drugim. Nasuprot tome, u drugim modelima, posebno onima namijenjenim razvoju kreativnosti, nastavnik pokušava izbjeći procjenjivački stav, i potaknuti učenika na samostalno i samousmjerenno učenje.
- **Sustav za podršku** opisuje uvjete potrebne za provedbu modela poučavanja. Podrška se odnosi na dodatne zahtjeve izvan uobičajenih ljudskih vještina, sposobnosti i tehničkih uređaja. To uključuje neke specifične vještine i znanja nastavnika, ali i posebne audio-vizualne materijale, knjige, filmove, laboratorijske setove, referentne materijale, studijske posjete i sl. Dakle, sustav za podršku predstavlja nastavne materijale i resurse koje nastavnik koristi za ispunjenje postavljenih ciljeva. Procjena o tome jesu li nastavni ciljevi postignuti ili ne obavlja se usmenim ili pismenim ispitom. Na osnovi tog uspjeha ili neuspjeha, dobiva se jasan pokazatelj o učinkovitosti strategija, taktika i tehnika koje se koriste u nastavi.
- **Primjena** je važan element nastavnog modela. Ona označava mogućnost korištenja naučenog gradiva u drugim situacijama. Svaki model poučavanja pokušava postići primjenjivost u različitim kontekstima povezanim s postignutim ciljevima u smislu kognitivne i afektivne promjene ponašanja učenika.

Modeli poučavanja mogu se grupirati u četiri skupine, na osnovi različitih stavova o načinima na koji ljudi uče (Joyce i sur, 2015):

- **Modeli obrade informacija** naglašavaju načine poboljšanja urođenog napora čovjeka da spozna smisao svijeta stjecanjem i organiziranjem podataka, prepoznavanjem problema i stvaranjem rješenja za njih, te razvojem koncepata i jezika za njihovo prenošenje.
- **Društveni modeli** su konstruirani kako bi kolektivnu energiju, tj. sinergiju, nastalu pri zajedničkom radu u učionici iskoristili za rješavanje nastalih problema. Na upravljanje razredom gledaju kao na razvoja suradničkih odnosa u učionici.
- **Osobni modeli** zasnivaju se na stavu da se ljudska stvarnost nalazi u individualnim svijestima. Ljudi razvijaju jedinstvene osobnosti i vide svijet iz perspektive koja je proizvod iskustava i stavova pojedinca. Zajednička razumijevanja proizvod su pregovaranja pojedinaca koji moraju živjeti, raditi i zajedno stvarati obitelji.

- ***Bihevioralni modeli*** prate teoriju društvenog učenja, prema kojoj su ljudska bića samoupravljajući komunikacijski sustavi koji mijenjaju ponašanje kao odgovor na informacije o tome koliko uspješno obavljaju zadatke. Ponašanje se prilagođava u skladu s povratnim informacijama.

Na osnovu rezultata istraživanja opisanog u nastavku, bit će predložen model poučavanja agentske paradigme korištenjem simulacijskih okruženja realiziranih pomoću alata za modeliranje i simuliranje zasnovanih na agentima.

3. METODOLOGIJA ISTRAŽIVANJA

3.1. Predmet i cilj istraživanja

Učenje i poučavanje u području računarstva zahtijeva složena konceptualna znanja i razumijevanja te je istovremeno tehničke i praktične prirode. Usvajanje i razumijevanje složenih i apstraktnih koncepata u relativno kratkom vremenu predstavlja izazov koji se stavlja pred nastavnike i studente. Problem nemogućnosti vizualizacije apstraktnih koncepata i njihove praktične primjene dovodi do slabijeg razumijevanja nastavnih sadržaja. S ciljem prevladavanja ovih poteškoća osmišljeno je istraživanje u kojem će se ispitati primjerenost i učinkovitost poučavanja agentske paradigme korištenjem simulacijskih okruženja. Simulacijska okruženja predstavljaju bogati kontekst koji se može koristiti za poučavanje koncepata računarstva, uključujući i one složene i apstraktne. Provedena istraživanja potvrdila su pozitivan učinak njihovog korištenja u nastavnom procesu na usvajanje novih znanja i pokazala da se ona mogu iskoristiti kao učinkovita i motivacijska okruženja za učenje. Praktični primjeri i zadaci, realizirani u ovim okruženjima, potaknut će studente na aktivno sudjelovanje u nastavi, a pristup učenja kroz rad olakšat će im usvajanje nastavnih sadržaja. Prema dobivenim rezultatima, predložit će se model poučavanja agentske paradigme u visokom obrazovanju kroz integrirano korištenje simulacijskih okruženja.

Cilj istraživanja je utvrditi kako primjena postojećih simulacijskih okruženja utječe na usvajanje i razumijevanje koncepata agentske paradigme kod studenata u visokom obrazovanju. Prema rezultatima istraživanja oblikovat će se model poučavanja agentske paradigme koji će sadržavati karakteristične primjere i zadatke iz ovog područja prikazane korištenjem simulacijskih okruženja. Oblikovani model poučavanja agentske paradigme na visokoškolskoj razini učiniti javno dostupnim.

3.2. Istraživačka pitanja

Na osnovu navedenog cilja istraživanja proizašla su sljedeća istraživačka pitanja:

1. Utječe li način poučavanja agentske paradigme na uspješnost studenata u usvajanju i razumijevanju nastavnih sadržaja ovog područja?
2. Postoje li razlike u uspješnosti usvajanja i razumijevanja nastavnih sadržaja iz područja agentske paradigme koje su posljedica odabranog simulacijskog okruženja?

Postavljena je polazna hipoteza prema kojoj će studenti informatike i računarstva postići bolje rezultate u usvajanju i razumijevanju osnovnih koncepata agentske paradigme uz poučavanje

temeljeno na korištenju simulacijskih okruženja u odnosu na klasičan način poučavanja *ex cathedra*.

3.3. Paradigma i metodologijski pristup

Kod Istraživanja u obrazovanju ne postoje laboratorijski uvjeti kao u nekim drugim područjima istraživanja: medicini, kemiji, fizici, i sl. Zbog tog nije moguć utjecaj na neke od faktora kao što su izbor uzorka populacije. Selektivni uzorak populacije učenika narušio bi se prirodni okoliš, odnosno ponašanje učenika, čime bi se u pitanje mogla dovesti i valjanost istraživanja. Opseg istraživanja ograničen je nastavnim planom i programom koji diktira vrijeme i trajanje provođenja ispitivanja, opseg pojmova čija se usvojenost provjerava, kao i dozvoljene instrumente. Zbog navedenih ograničenja istraživanje opisano u ovom radu provedeno je u tri dijela, odnosno kroz tri istraživanja, što je omogućilo triangulacija istraživanja. U svakoj od faza istraživanja eliminirano je neko od ograničenja iz ostalih istraživanja čime je dobiven potpuniji uvid u istraživački problem.

Svi dijelovi istraživanje provedeni su u prirodnom okruženju učionice, bez prisustva trećih osoba. Ovakav pristup eliminirao je neprirodno ponašanje sudionika, tj. studenata, tijekom istraživanja. Svjestan pristanak studenata nije bio neophodan jer će se nastava odvijati prema postojećem planu i programu. Uz ovakav pristup istraživanje spada u naturalističku paradigmu jer se pojava i sudionici promatraju u prirodnom okruženju. Time se eliminira utjecaj vanjskih čimbenika na dobivene rezultate (Cohen i sur., 2013). Osim rezultata koji će ukazati na razinu usvojenosti osnovnih koncepata agentske paradigme, u prvom dijelu istraživanja prikupljena su i iskustva, stavovi i dojmovi sudionika o održanoj nastavi te je, fenomenološkom analizom utvrđena razina usvojenosti koncepte agentske paradigme.

Ovo istraživanje se ima značajke empirijskog istraživanja jer su podaci prikupljeni tijekom nastavnog procesa. Ono je istovremeno i transverzalno jer se provodilo tijekom tri akademske godine na različitim studijskim godinama. Istraživanje je i razvojno jer njegovi rezultati mogu dovesti do primjene novih metoda poučavanja agentske paradigme na visokoškolskoj razini. Na kraju, s obzirom na činjenicu sa je autor sudjelovao u svim pripremnim i izvedbenim fazama istraživanja, ono se može okarakterizirati i kao akcijsko.

Prirodno okruženje sudionika, u kojem je istraživač/nastavnik dio istraživanog svijeta, osigurala je, prema naturalističkim istraživanjima, valjanost provjere znanja. Dodatno je unutar njega valjanost osigurana triangulacijom istraživača (kroz aktivno sudjelovanje više nastavnika u prvom i drugom dijelu istraživanja), prostornom triangulacijom (koja je postignuta

uključivanjem dvije institucije u drugom dijelu istraživanja), triangulacijom unutar metoda (kroz ponavljanje istraživanja učinka simulacije u različitim vremenskim točkama) kao i triangulacijom između metoda (jer su se, osim kvantitativnih podataka o nastavnim metodama, prikupili i stavovi i utisci studenata o održanoj nastavi). Konstruktna valjanost je osigurana opsežnim pretraživanjem literature i uzimanjem u razmatranje različitih značenja pojedinih konstrukata agentske paradigme. Sadržajna valjanost provjere znanja nakon svake nastavne jedinice osigurana je činjenicom da su se provjere odnosile na ograničeni opseg sadržaja kao i da su provjere znanja prethodno sadržajno usuglašene s predmetnim nastavnicima s dugogodišnjim iskustvom u nastavi iz ovog područja.

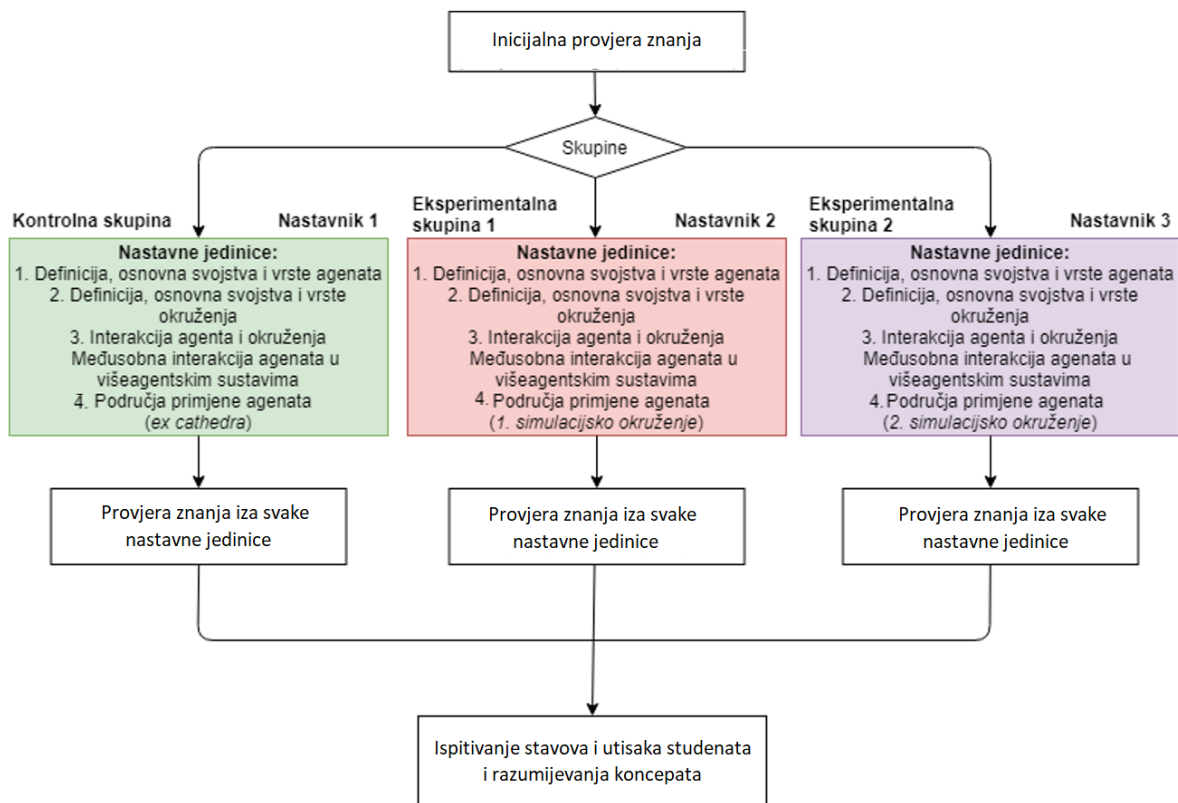
3.4. Nacrt istraživanja

Istraživanje je provedeno u tri dijela, u akademskim godinama 2017./2018., 2018./2019. i 2019./2020. Nastavne jedinice iz područja agentske paradigme su se za vrijeme istraživanja obrađivale na dva načina, klasičnim *ex cathedra* pristupom i korištenjem simulacijskih okruženja realiziranih alatima za modeliranje i simuliranje zasnovanim na agentima.

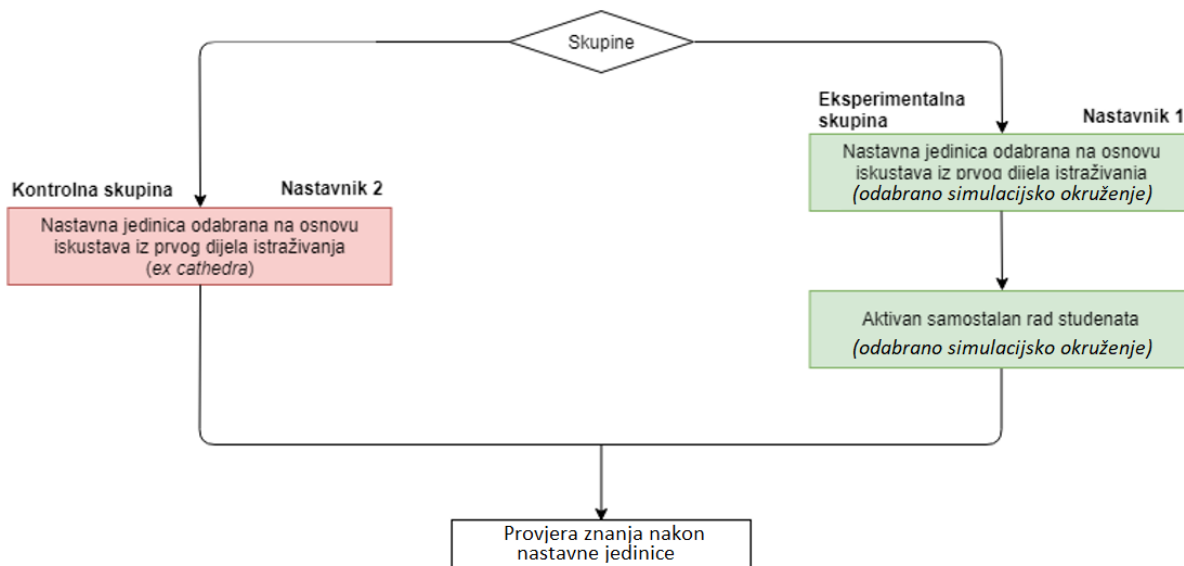
Prvi dio istraživanja proveden je u ljetnom semestru akademske 2017./2018. godine, nad 27 studenata četvrtog semestra diplomskog studija informatike, tijekom pet tjedana nastave, odnosno 10 školskih sati, u okviru predmeta Metodičko-informatički seminar s praksom na Fakultetu prirodoslovno-matematičkih i odgojnih područja Sveučilišta u Mostaru. U ovom dijelu istraživanja kontrolna skupina je prisustvovala klasičnoj *ex cathedra* nastavi, a dvije eksperimentalne skupine su prisustvovala nastavi uz korištenje simulacijskih okruženja realiziranih pomoću dva odabrana alata za modeliranje i simuliranje zasnovana na agentima. Shematski prikaz prvog dijela istraživanja prikazan je na slici 13.

Drugi dio istraživanja proveden je u zimskom semestru akademske 2018./2019. godine nad 80 studenata prvog semestra preddiplomskog studija informatike u okviru predmeta Uvod u računarstvo na Fakultetu prirodoslovno-matematičkih i odgojnih područja Sveučilišta u Mostaru i nad 72 studenta prvog semestra diplomskih studija strojarstva i računarstva u okviru predmeta Umjetna inteligencija na Fakultetu strojarstva, računarstva i elektrotehnike Sveučilišta u Mostaru. Drugi dio istraživanja provodio se tijekom jednog tjedna nastave, odnosno 2 školska sata, prema nastavnom planu i programu. U ovom dijelu istraživanja kontrolna skupina je prisustvovala klasičnoj *ex cathedra* nastavi, a eksperimentalne skupine su prisustvovala nastavi uz korištenje simulacijskih okruženja realiziranih pomoću simulacijskih

okruženja realiziranih odabranim alatom za modeliranje i simuliranje zasnovanom na agentima. Shematski prikaz prvog dijela istraživanja prikazan je na slici 14.

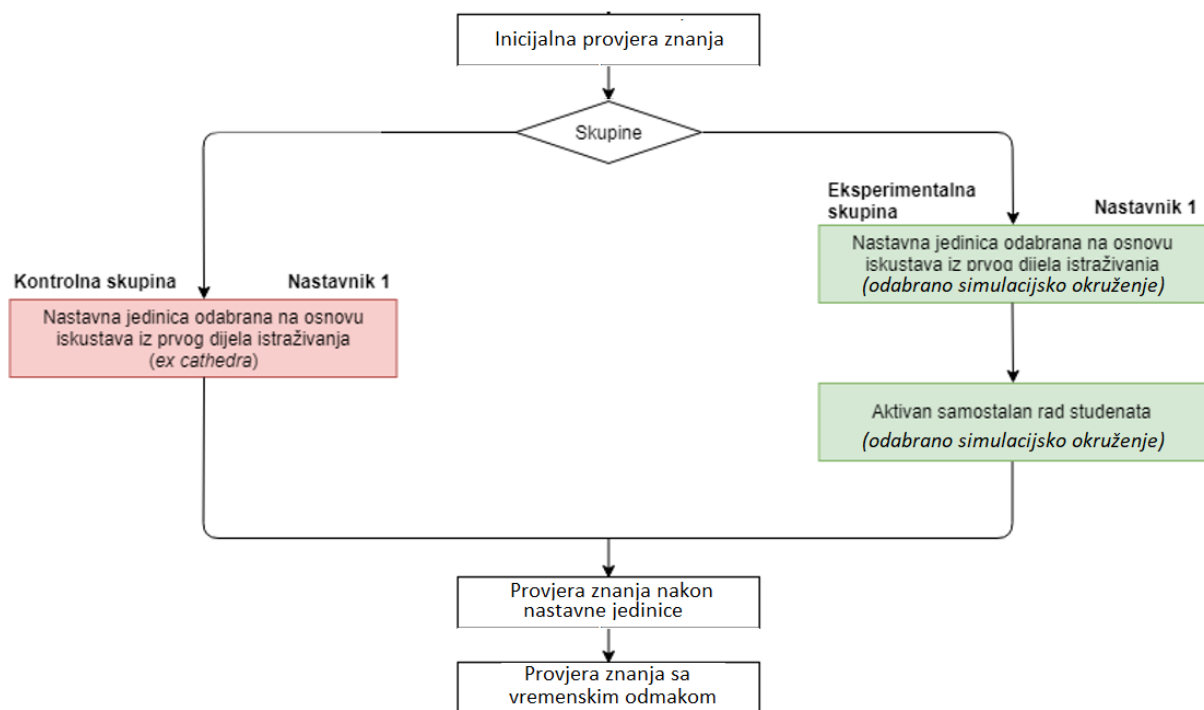


Slika 13. Shematski prikaz prvog dijela istraživanja



Slika 14. Shematski prikaz drugog dijela istraživanja

Zbog nedostataka uočenih analizom provedenog drugog dijela istraživanja, koje su mogle negativno utjecati na kvalitetu prikupljenih podataka, a time i na rezultate dobivene analizom tih podataka. proveden je treći dio istraživanja u zimskom semestru akademske 2019./2020. godine, nad 48 studenata prvog semestra diplomskog studij računarstva u okviru predmeta Umjetna inteligencija na Fakultetu strojarstva, računarstva i elektrotehnike Sveučilišta u Mostaru. Treći dio istraživanja provodio se tijekom jednog tjedna nastave, odnosno 4 školska sata, prema nastavnom planu i programu. U ovom dijelu istraživanja kontrolna skupina je prisustvovala klasičnoj *ex cathedra* nastavi, a eksperimentalne skupina je prisustvovala nastavi uz korištenje simulacijskih okruženja realiziranih odabranim alatom za modeliranje i simuliranje zasnovanom na agentima. Shematski prikaz prvog dijela istraživanja prikazan je na slici 15.



Slika 15. Shematski prikaz trećeg dijela istraživanja

Svi dijelovi istraživanja bit će detaljnije opisani u nastavku.

3.5. Mjerni instrumenti i analiza podataka

Mjerni instrumenti koji se korišteni u prvom dijelu istraživanja su:

- a) inicijalna provjera znanja za utvrđivanje ujednačenosti grupa,
- b) provjere znanja nakon svake nastavne jedinice, za utvrđivanje usvojenosti i razumijevanja osnovnih koncepata agentske paradigme,

c) anketni upitnik za ispitivanje stavova i utisaka studenata o održanoj nastavi, eventualnih primjedbi i preporuka.

d) standardizirani intervju sa pitanjima otvorenog tipa

Inicijalna provjera znanja zasnivala se na zadacima kojima se provjeravalo poznavanje koncepata agentske paradigme. Koristili su se zadaci otvorenog tipa i zadaci višestrukog izbora.

Za provjere znanja nakon nastavne jedinice i sa vremenskim odmakom koristili su se zadaci otvorenog tipa za provjeru razina usvajanja i razumijevanja testiranih koncepata

Anketni upitnik s nizom pitanja procjena u obliku Likertovih ljestvica s 5 ponuđenih kategorija poslužio je za ispitivanje stavova i utisaka studenata o održanoj nastavi

Unaprijed definirana pitanja imala su za cilj utvrđivanje razine usvajanja i razumijevanja koncepata agentske paradigme

Instrument koji se koristio u drugom dijelu istraživanja je:

e) provjera znanja za utvrđivanje usvojenosti i razumijevanja koncepata odabrane nastavne jedinice.

Za provjeru znanja nakon nastavne jedinice i sa vremenskim odmakom koristili su se zadaci otvorenog tipa za provjeru razina usvajanja i razumijevanja testiranih koncepata

Instrumenti koji su se koristili u trećem dijelu istraživanja su:

f) inicijalna provjera znanja za utvrđivanje ujednačenosti grupa,

g) provjera znanja nakon nastavne jedinice, za utvrđivanje usvojenosti i razumijevanja osnovnih koncepata agentske paradigme,

h) provjera znanja sa vremenskim odmakom, za utvrđivanje razine zadržavanja usvojenog znanja iz obrađenih nastavnih sadržaja.

Inicijalna provjera znanja zasnivala se na zadacima kojima se provjeravalo poznavanje koncepata agentske paradigme. Koristili su se zadaci otvorenog tipa i zadaci višestrukog izbora.

Za provjere znanja nakon nastavne jedinice i sa vremenskim odmakom koristili su se zadaci otvorenog tipa za provjeru razina usvajanja i razumijevanja testiranih koncepata.

Za analizu kvantitativnih podataka koristio se programski paket za statističku obradu podataka IBM SPSS Statistics 20.

Za provjeru pouzdanosti provedenih provjera znanja koristio se Cronbach α koeficijent za procjenu pouzdanosti u pisanim testovima. Tumačenje Cronbach α koeficijenta je sljedeće (Cohen i sur., 2013):

- veće od 0,90 – vrlo visoko pouzdan
- 0,80–0,90 – visoko pouzdan
- 0,70–0,79 – pouzdan
- 0,60–0,69 – granično / minimalno pouzdane
- manje od 0,60 – neprihvatljivo nisko pouzdane.

4. MODEL POUČAVANJA AGENTSKE PARADIGME KORIŠTENJEM SIMULACIJSKIH OKRUŽENJA

Istraživanje je provedeno u prirodnom okruženju učionice, bez prisustva trećih osoba. Ovakvim pristupom eliminirano je neprirodno ponašanje sudionika, tj. studenata, tijekom istraživanja. Svjestan pristanak studenata nije bio neophodan jer se nastava odvijala prema postojećem planu i programu.

Uz ovakav pristup istraživanje spada u naturalističku paradigmu jer su pojava i sudionici promatrani u prirodnom okruženju. Time je eliminiran utjecaj vanjskih čimbenika na dobivene rezultate (Cohen i sur., 2013).

Osim rezultata koji iskazuju razinu usvojenosti osnovnih koncepata agentske paradigme, prikupljena su i iskustva i stavovi sudionika o održanoj nastavi, kao i njihove primjedbe i prijedlozi. Također su, nizom standardiziranih pojedinačnih intervjua, utvrđeni načini na koje studenti doživljavaju i shvaćaju koncepte agentske paradigme, kao i najčešće miskonceptije prilikom usvajanja nastavnih sadržaja.

Istraživanje je imalo i praktičnu primjenu jer su njegovi rezultati doveli do prijedloga modela poučavanja agentske paradigme koji će se moći primijeniti u redovitoj nastavi.

4.1. Prvi dio istraživanja

Prvi dio istraživanja proveden je tijekom pet tjedana nastave ljetnog semestra akademske 2017./2018. godine, u okviru predmeta Metodičko informatički seminar s praksom, u četvrtom semestru diplomskog studija informatike na Fakultetu prirodoslovno-matematičkih i odgojnih područja Sveučilišta u Mostaru.

Planirana nastava odnosila se na odabrane teme iz područja agentske paradigme koje bi upoznale studente s pojmovima (inteligentnog) agenta, okruženja u kojem agent djeluje, interakcijom agenta i okruženja, odnosno više agenata u višeagentskom sustavu te potencijalnim područjima primjene agenata.

Aktualni nastavni plan i program uključuje nastavnu cjelinu definiranja općeg modela nastave izvedenog iz didaktičkih teorija i modela poučavanja, što je omogućilo da se za potrebe istraživanja provede nastava prema redovnom programu, te osmisle i izrade instrumenti provjere znanja.

Tijekom prvog dijela istraživanja studenti su bili podijeljeni u tri skupine, dvije eksperimentalne (ES1 i ES2) i jednu kontrolnu skupinu (KS). Nastavi je prisustvovalo ukupno 27 studenata koji čine neslučajni prirodni uzorak.

Za potrebe istraživanja korištena su dva ranije spomenuta simulacijska okruženja različite razine složenosti, tj. *Cellular* i *NetLogo*.

Nastava za kontrolnu skupinu izvodila se na "klasičan" način, tj. *ex cathedra*, uz korištenje pripremljenih prezentacija i tradicionalnih nastavnih pomagala krede i ploče, dok se za dvije eksperimentalne skupine nastava izvodila uz korištenje simulacijskih okruženja i pripremljenih karakterističnih primjera i zadataka za opisivanje koncepata agentske paradigme. Primjeri su odabrani i razrađeni u dogovoru s nastavnicima s višegodišnjim iskustvom u provedbi nastave koja uključuje koncepte agentske paradigme obrađene planiranom nastavom.

Nastavu su izvodila tri nastavnika s višegodišnjim iskustvom u provedbi nastave iz područja računarstva i informatike. Svaki od nastavnika je izvodio je svu predviđenu nastavu za jednu od skupina. Svaku skupinu činilo je 9 studenata.

4.1.1. Inicijalna provjera znanja

Na početku istraživanja provedena je inicijalna provjera znanja svih ispitanika iz područja agentske paradigme.

Inicijalna provjera znanja, u trajanju od 30 minuta, uključivala je zadatke otvorenog tipa i zadatke višestrukog izbora, a studenti su mogli ostvariti najviše 60 bodova (Prilog 17).

Studenti su prije početka provjere dobili jasne upute o načinu rješavanja, te su upoznati s razrađenim kriterijem bodovanja, usklađenim s razinom znanja potrebnim za rješavanje pojedinog zadatka. Mogućnost pogađanja umanjena je konstrukcijom pitanja.

Cilj ove provjere bio je ustanoviti predznanja ispitanika i međusobno usporediti ispitivane skupine.

Za provjeru pouzdanosti ispita korišten je Cronbach α koeficijent. Inicijalna provjera znanja spada u kategoriju visoko pouzdanih (Cronbach $\alpha = 0.812$).

Na osnovu postignutih rezultata izračunati su aritmetička sredina (AS), medijan (MED) i aritmetička sredina rangova (ASR) kao tri mjere srednje vrijednosti te standardna devijacija (SD) i raspon (R) kao dvije mjere disperzije. Analiza rezultata inicijalne provjere znanja prikazana je u Tablici 3.

Tablica 3. Analiza rezultata inicijalne provjere znanja

Statistička mjera	Kontrolna skupina (KS)	Ekperimentalna skupina 1 (ES1)	Ekperimentalna skupina 2 (ES2)
Aritmetička sredina (AS)	37,20	35,33	35,56
Medijan (MED)	35,00	35,00	35,00
Aritmetička sredina rangova (ASR)	14,39	13,67	13,94
Standardna devijacija (SD)	8,70	11,50	11,30
Raspon (R)	30,00	40,00	35,00

Postavljena je nulta hipoteza H_0 da ne postoji značajna razlika u postignućima tri skupine, odnosno da je raspodjela rezultata jednaka u svim skupinama.

Kako se radi o malom uzorku odlučili smo se za neparametrijski Kruskal-Wallisov test za tri nezavisne skupine kojim je testirana nulta hipoteza. U tu svrhu svi rezultati su rangirani po veličini. Rezultat analize, provedene uz pomoć statističkog paketa programske podrške SPSS, prikazan je u Tablici 4.

Tablica 4. Kruskal-Wallisov test za tri nezavisne skupine

Nulta hipoteza H_0	Provedeni test	Značajnost	Odluka
Raspodjela rezultata jednaka je u svim skupinama	Kruskal-Wallisov test za nezavisne uzorke	0,981	Zadržati nultu hipotezu H_0

Prikazana je asimptotska značajnost. Razina značajnosti je 0,05.

Statistička analiza rezultata inicijalne provjere znanja potvrdila je nultu hipotezu H_0 . Time je pokazano je da ne postoji statistički značajna razlika u predznanju studenata, odnosno da je raspodjela postignutih rezultata jednaka u sve tri skupine.

Nastava provedena u okviru prvog dijela istraživanja uključivala je sljedeće nastavne jedinice:

- definicija, osnovna svojstva i vrste agenata (NJ1),
- definicija, osnovna svojstva i vrste okruženja (NJ2),
- interakcija agenata i okruženja i međusobna interakcija agenata u višeagentskim sustavima (NJ3),
- područja primjene agenata (NJ4).

Nastava je izvedena u četiri uzastopna tjedna tijekom kojih je za svaku nastavnu jedinicu utrošeno 90 minuta, tj. dva školska sata.

Neposredno nakon svake nastavne jedinice provedeno je ispitivanje znanja za utvrđivanje razine usvojenosti obrađenih nastavnih sadržaja.

Ispitivanja znanja, u trajanju od 30 minuta, uključivala su zadatke otvorenog tipa čije je rješavanje zahtijevalo različite razine znanja definirane Bloomovom taksonomijom u kognitivnom području (Bloom, 1956), a studenti su na svakom od njih mogli ostvariti najviše 30 bodova.

Na početku provjere studenti su dobili upute o načinu rješavanja i upoznati su s kriterijem bodovanja pojedinih zadataka. Jedinstvenim ispitivanjem bile su obuhvaćene sve tri skupine ispitanika, kako bi se mogla usporediti postignuća studenata pojedinih skupina.

4.1.2. Definicija, osnovna svojstva i vrste agenata (NJ1)

Prva nastavna jedinica (NJ1) poslužila je za upoznavanje studenata s različitim definicijama pojma agenta (Coen, 1995; Maes, 1995; Steels, 1995; Wooldridge i Jennings, 1995; Nwana, 1996; Russell i Norvig, 2016).

Također su pobrojana i osnovna svojstva agenata, odnosno osobine ovih entiteta programske podrške koji ih čine agentima (Rosenschein i Genesereth, 1985; Galliers, 1988; Shoham, 1993; Castelfranchi, 1994; Genesereth i Ketchpel, 1994; Maes, 1994; White, 1994; Wooldridge i Jennings 1995; Franklin i Graesser, 1996; Florian, 2000).

Kao što je ranije navedeno, u ovom smo se istraživanju odlučili za klasifikaciju agenata koju predlažu Russell i Norvig (2016), odnosno podjelu prema stupnju izraženog inteligentnog ponašanja i sposobnosti. Vrste agenata, prema ovoj podjeli, su:

- jednostavni reaktivni (refleksivni) agenti;
- reaktivni agenti sa stanjima;
- agenti usmjereni prema cilju;
- agenti usmjereni prema uspješnosti (korisnosti);
- agenti koji uče.

Zbog vremenskog ograničenja, u ovoj nastavnoj jedinici je najsloženija vrsta agenata, tj. agenti koji uče, samo definirana. Ostale vrste agenata su definirane i opisane, a njihove karakteristike dodatno obrazložene kroz pripremljene prigodne primjere. Za kontrolnu skupinu primjeri su prikazani korištenjem prezentacije i projektora, dok su obje eksperimentalne skupine samostalno pokrenule, koristile i u manjoj mjeri manipulirale primjerima realiziranim u simulacijskim okruženjima alata *Cellular* i *NetLogo*. Pri tome su primjeri za sve tri skupine

opisivali agente istovjetnih mogućnosti i ponašanja. Detaljan opis primjera, kojima su se studenti upoznali s različitim vrstama agenata, njihovim karakteristikama i mogućnostima dat je u nastavku.

4.1.2.1. Jednostavni reaktivni agenti

Primjer 1.1: U ovom primjeru je opisan rad jednostavnog reaktivnog agenta koji pokreće autonomno vozilo. Podsjetit ćemo se ranije navedenog pseudokoda jednostavnog reaktivnog agenta (Slika 16.)

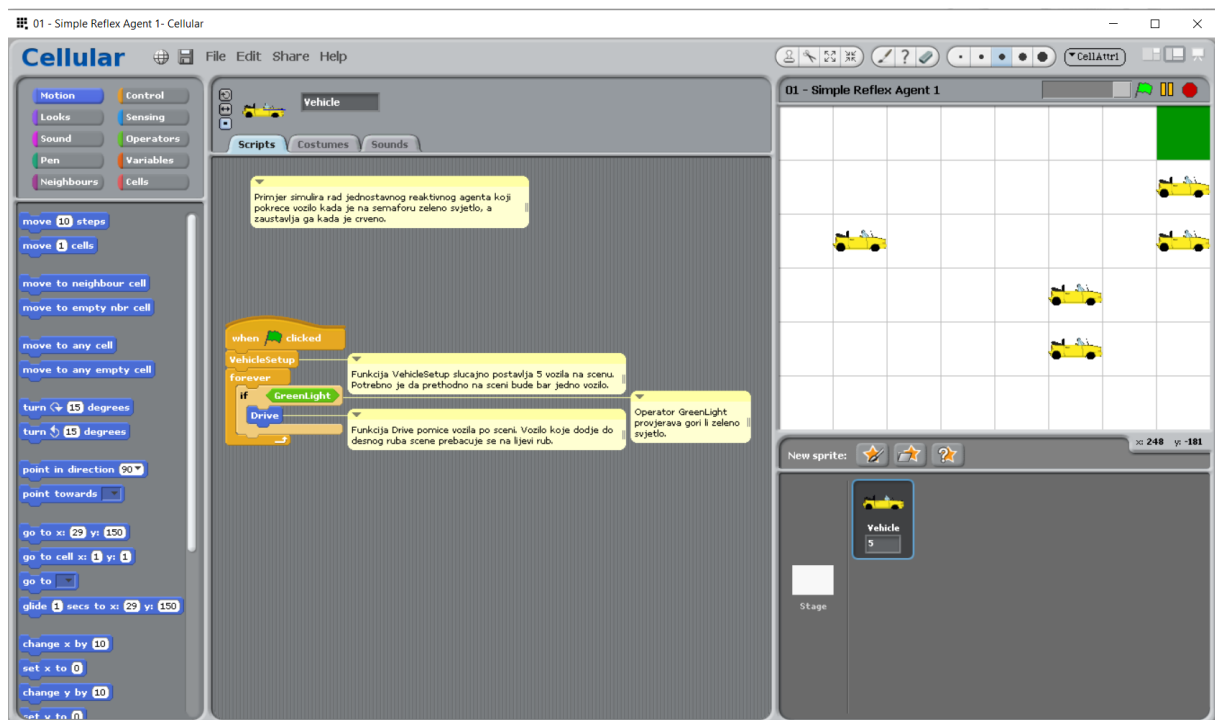
```

function JEDNOSTAVNI-REAKTIVNI-AGENT (opažaj)
  static: pravila /* skup unaprijed definiranih pravila stanje-akcija */

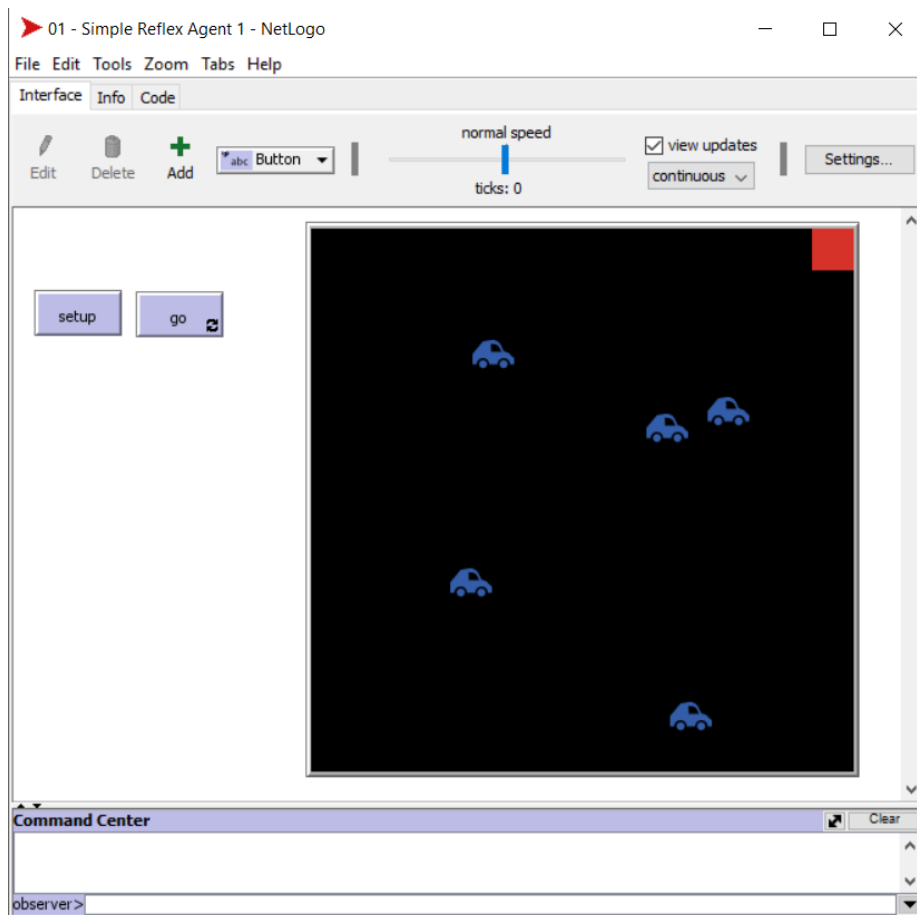
  stanje ← INTERPRET-INPUT (opažaj)
  pravilo ← RULE-MATCH (stanje, pravila)
  akcija ← RULE-ACTION (pravilo)
  return akcija
  
```

Slika 16. Pseudokod agentske funkcije jednostavnog reaktivnog agenta

Kada senzori agenta opaze da je na semaforu zeleno svjetlo, vozilo se pokreće i vozi sve dok senzori agenta ne opaze crveno svjetlo na semaforu. U tom trenutku vozilo se zaustavlja do ponovnog opažanja zelenog svjetla (Prilog 1).



Slika 17. Jednostavni reaktivni agent koji pokreće autonomno vozilo u Cellularu

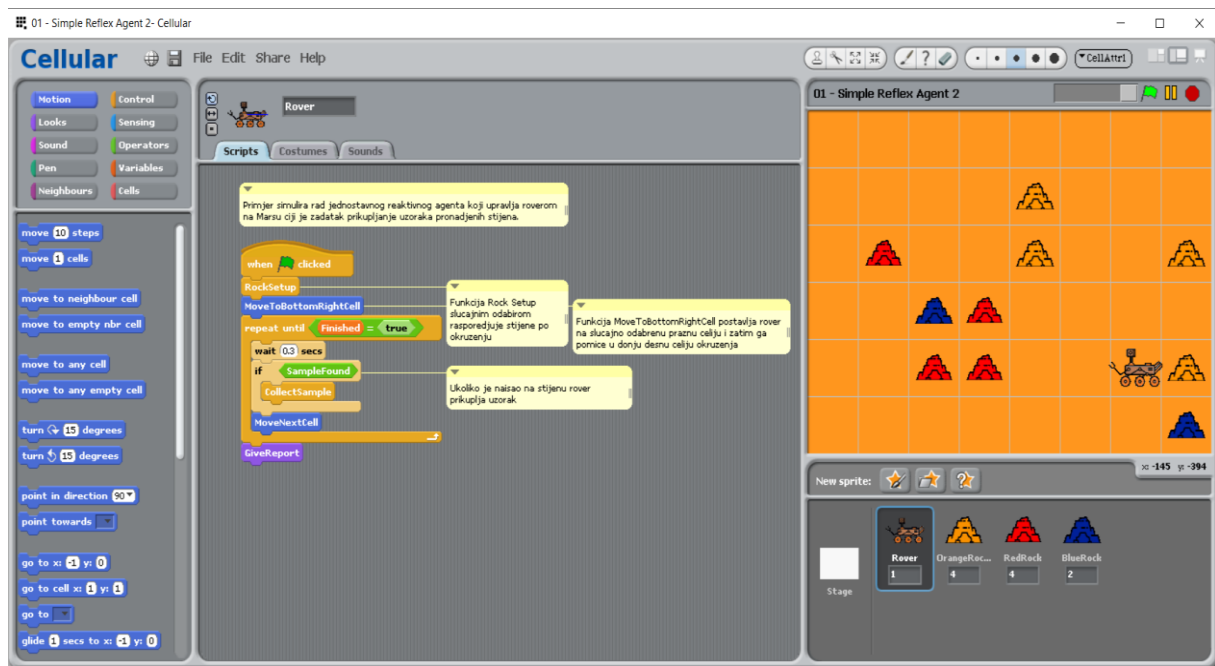


Slika 18. Jednostavni reaktivni agent koji pokreće autonomno vozilo u *NetLogo*

Agent koji upravlja autonomnim vozilom reagira samo na trenutni podražaj, tj. svjetlo koje je trenutno upaljeno na semaforu, ne vodeći računa o prethodnim opažanjima. Njegova funkcionalnost je zasnovana na dva definirana pravila stanje-akcija; opaženo zeleno svjetlo inicira pokretanje automobila, a opaženo crveno svjetlo njegovo zaustavljanje.

Studenti eksperimentalnih skupina su u simulaciji mogli mijenjati broj automobila, kao i interval trajanja crvenog, odnosno zelenog svjetla na semaforu.

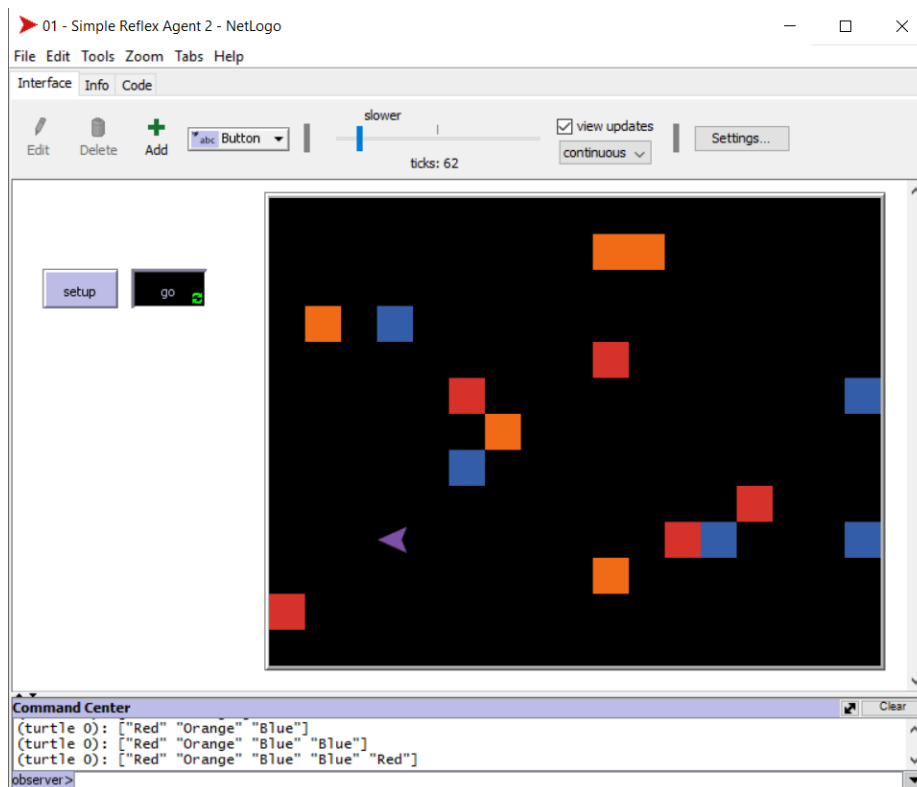
Primjer 1.2: U ovom primjeru je opisan rad jednostavnog reaktivnog agenta koji upravlja roverom na Marsu čiji je zadatak prikupljanje uzoraka pronađenih stijena. Kada senzori agenta opaze da je na rover naišao na stijenu prikuplja se uzorak stijene. Ukoliko na poziciji rovera nema stijene, rover nastavlja kretanje (Prilog 2).



Slika 19. Jednostavni reaktivni agent koji pokreće rover na Marsu u *Cellularu*

Agent koji upravlja roverom na Marsu reagira samo na trenutni podražaj, tj. postojanje stijene na trenutnoj poziciji rovera, ne vodeći računa o prethodnim opažanjima. Njegova funkcionalnost je zasnovana na jednom definiranom pravilu stanje-akcija; opažena stijena inicira uzimanje uzorka.

Studenti eksperimentalnih skupina su u simulaciji mogli mijenjati broj stijena inicijalno postavljenih u okruženju.



Slika 20. Jednostavni reaktivni agent koji pokreće rover na Marsu u *NetLogo*

4.1.2.2. Reaktivni agenti sa stanjima

Primjer 1.3: U ovom primjeru je opisan rad reaktivnog agenta sa stanjima. Podsjetit ćemo se ranije navedenog pseudokoda reaktivnog agenta sa stanjima (Slika 21.)

```

function REAKTIVNI-AGENT-SA-STANJIMA (opažaj)
  static: stanje /* opis trenutnog stanja okruženja */
           pravila /* skup unaprijed definiranih pravila stanje-akcija */

  stanje ← UPDATE-STATE (stanje, opažaj)
  pravilo ← RULE-MATCH (stanje, pravila)
  akcija ← RULE-ACTION (pravilo)
  stanje ← UPDATE-STATE (stanje, akcija)
  return akcija

```

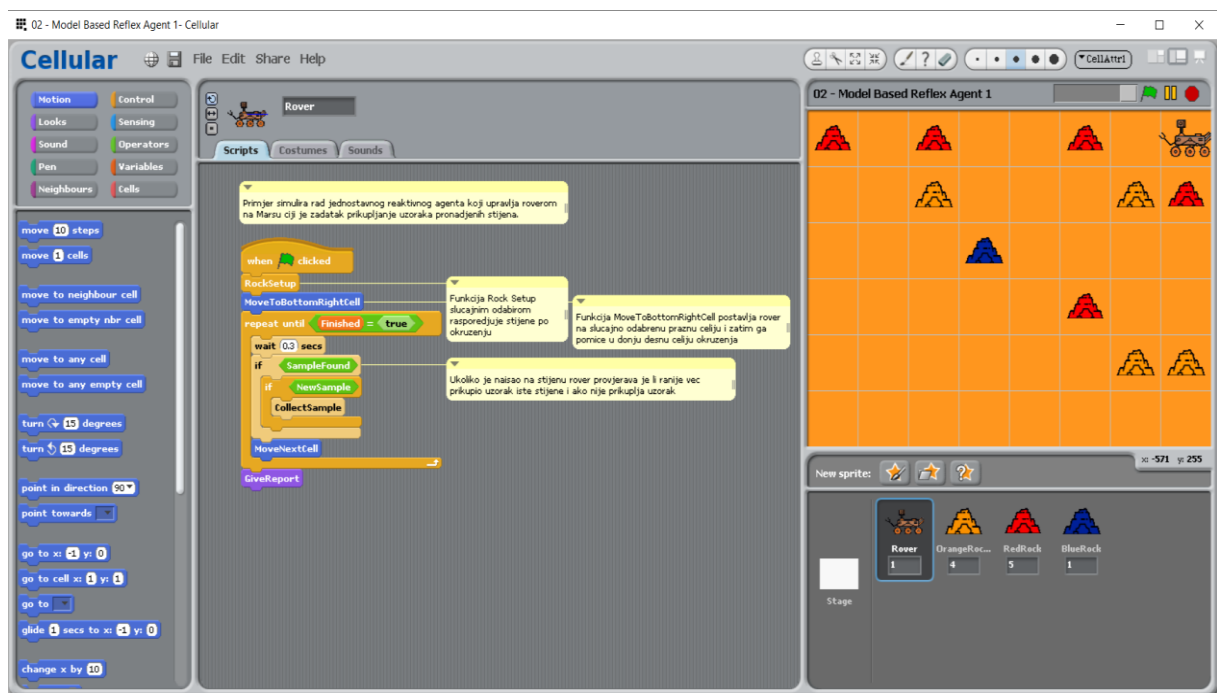
Slika 21. Pseudokod agentske funkcije reaktivnog agenta sa stanjima

Agent upravlja roverom na Marsu čiji je zadatak prikupljanje uzoraka pronađenih stijena, vodeći računa o ranije prikupljenim uzorcima. Kada senzori agenta opaze da je na rover naišao na stijenu čiji uzorak do tada nije prikupio, uzima se uzorak. Ukoliko na poziciji rovera nema stijene ili se nalazi stijena čiji je uzorak ranije prikupljen, rover nastavlja kretanje (Prilog 3).

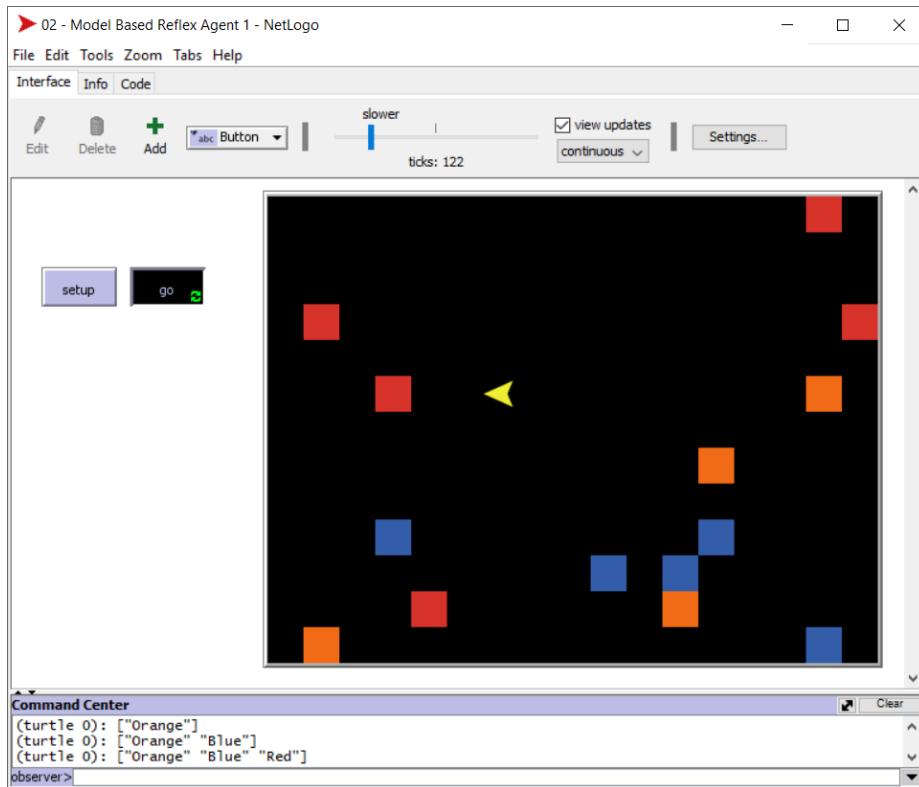
Agent koji upravlja roverom na Marsu reagira na trenutni podražaj, tj. postojanje stijene na trenutnoj poziciji rovera, ali vodi računa i o prethodnim opažanjima, tj. provjerava je li uzorak stijene na koju je naišao već prikupljen.

Trenutni opažaj, u kombinaciji s prethodno pohranjenim modelom stanja, daje ažurirani opis stanja. Korištenjem ovog internog modela, odnosno povijesti opažanja agent odabire akciju koju će izvesti.

Studenti eksperimentalnih skupina su u simulaciji mogli mijenjati broj stijena inicijalno postavljanih u okruženju.

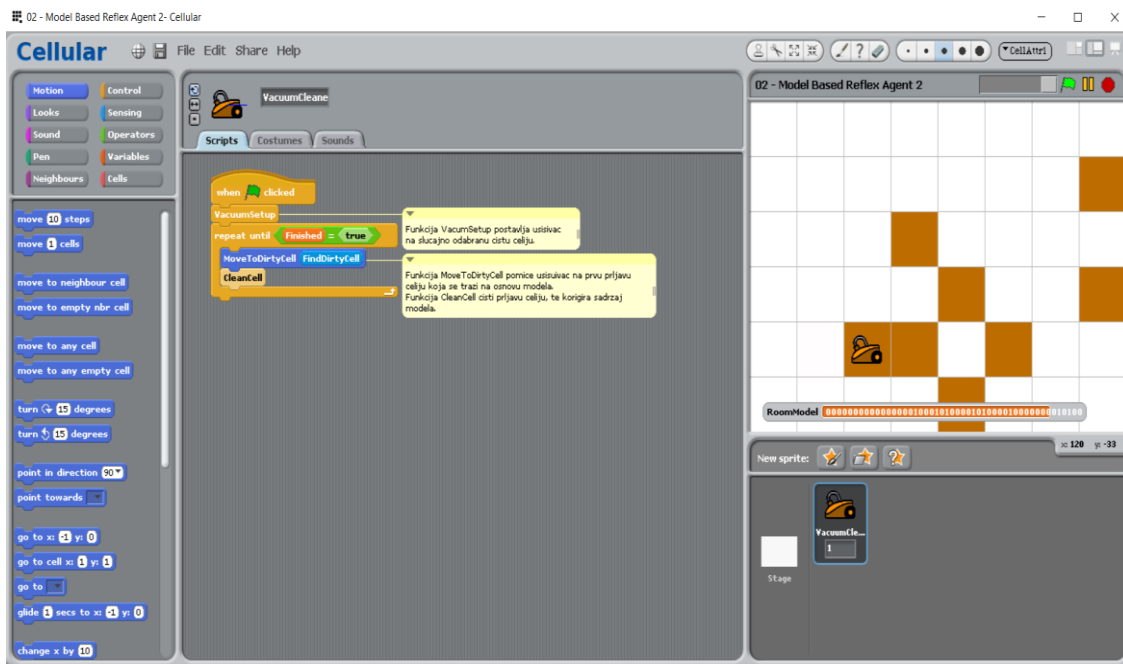


Slika 22. Reaktivni agent sa stanjima koji pokreće rover na Marsu u *Cellularu*

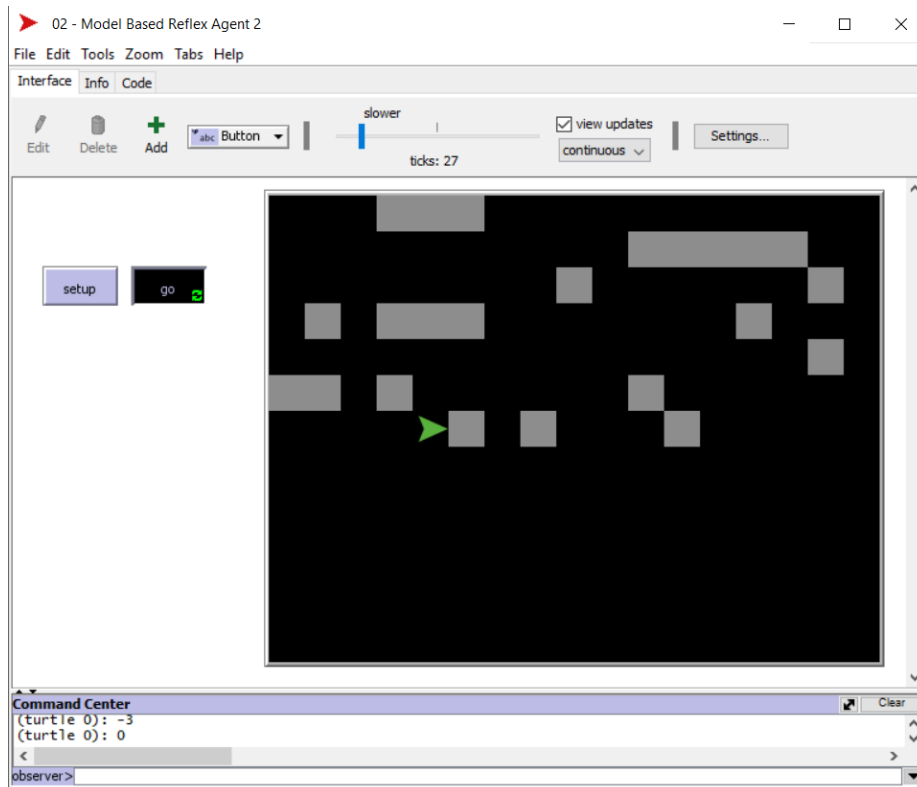


Slika 23. Reaktivni agent sa stanjima koji pokreće rover na Marsu u *NetLogo*

Primjer 1.4: U ovom primjeru je opisan rad reaktivnog agenta sa stanjima koji upravlja radom pametnog usisavača. Na osnovi modela prostorije, odnosno rasporeda prljavih područja, agent se pomiče na najbliže prljavo područje i čisti ga, pri tome ažurirajući model prostorije (Prilog 4).



Slika 24. Reaktivni agent sa stanjima koji pokreće pametni usisavač u *Cellularu*



Slika 25. Reaktivni agent sa stanjima koji pokreće pametni usisavač u *NetLogo*

Agent koji upravlja pametnim usisivačem djeluje na osnovi pohranjenog modela stanja okruženja. Nakon poduzete akcije, tj. čišćenja prljavog dijela u prostoriji, model stanja se ažurira. Korištenjem ovog internog modela agent odabire akciju koju će izvesti.

Studenti eksperimentalnih skupina su u simulaciji mogli mijenjati broj inicijalno postavljanih prljavih dijelova prostorije.

4.1.2.3. Agenti usmjereni prema cilju

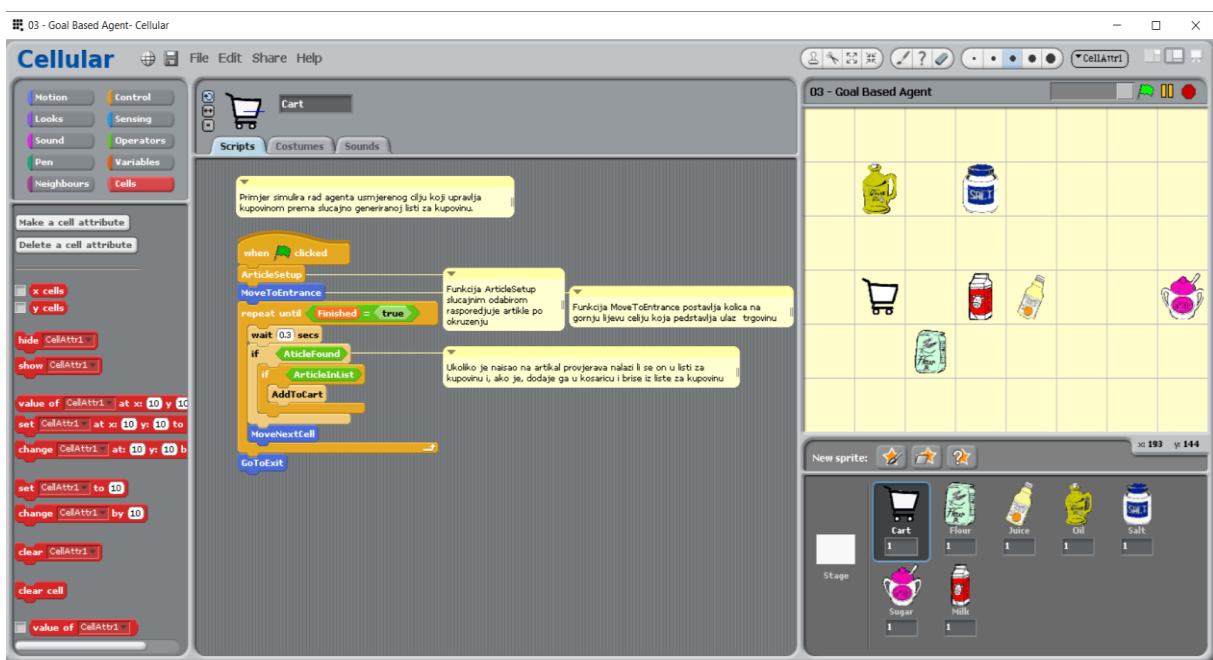
Primjer 1.5: U ovom primjeru je opisan rad agenta usmjerenog prema cilju. Podsjetit ćemo se ranije navedenog pseudokoda agenta usmjerenog prema cilju (Slika 23.)

```
function AGENT-USMJEREN-PREMA-CILJU (opažaj)
  static: stanje /* opis trenutnog stanja okruženja */
           pravila /* skup unaprijed definiranih pravila stanje-akcija */
           cilj /* skup definiranih ciljanih stanja */

  stanje ← UPDATE-STATE(stanje, opažaj)
  pravilo ← RULE-MATCH(stanje, pravila)
  akcija ← RULE-ACTION(pravilo)
  stanje ← UPDATE-STATE(stanje, akcija)
  if (stanje in cilj) then
    return akcija
  else
    opažaj ← OBTAIN-PERCEPT(stanje, cilj)
    return AGENT-USMJEREN-PREMA-CILJU (opažaj)
```

Slika 26. Pseudokod agentske funkcije agenta usmjerenog prema cilju

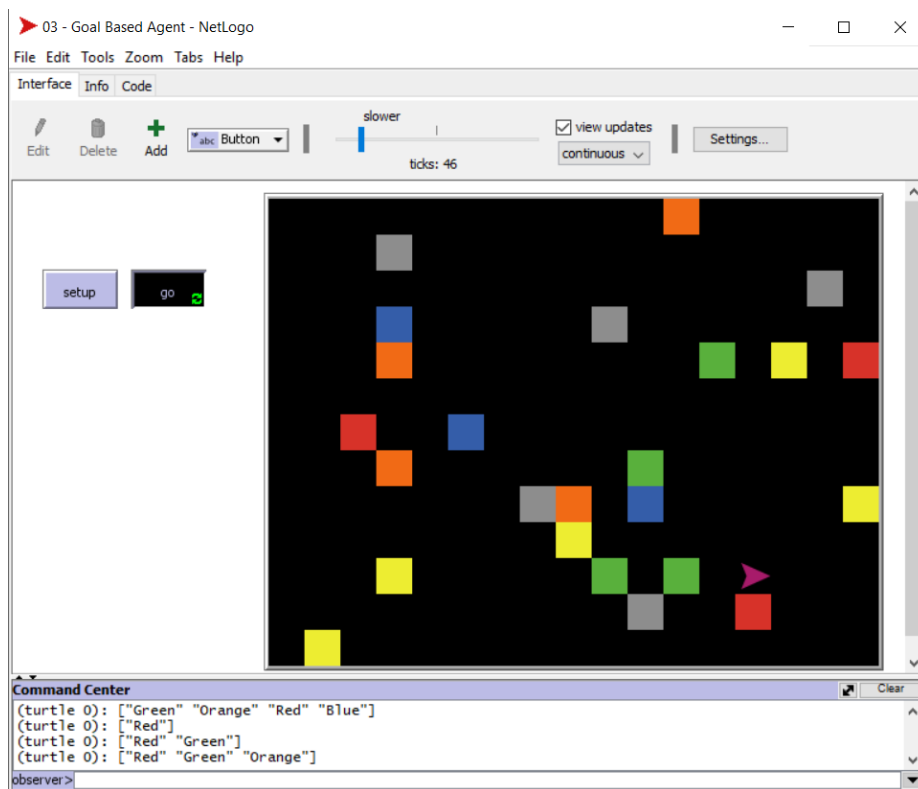
Agent upravlja pametnim kolicima u trgovini i obavlja kupovinu u skladu sa slučajno generiranim popisom za kupovinu. Kada senzori agenta opaze da su kolica naišla na artikl u trgovini, agent provjerava nalazi li se taj artikl na popisu i u tom slučaju stavlja ga u kolica. Ukoliko na poziciji kolica nema artikla, ili se nalazi artikl koji nije na popisu za kupovinu, kolica nastavljaju kretanje (Prilog 5).



Slika 27. Agent usmjeren prema cilju koji pokreće pametna kolica u trgovini u *Cellularu*

Agent koji upravlja pametnim kolicima u trgovini ima zadani cilj, tj. inicijalizirani popis za kupovinu. Informacija o cilju djelovanja opisuje poželjne situacije, u ovom slučaju artikle s popisa, i na taj način omogućava agentu da između više mogućnosti odabere onu koja vodi ka ciljanom stanju.

Studenti eksperimentalnih skupina su u simulaciji mogli mijenjati broj artikala koji će se naći na popisu.



Slika 28. Agent usmjeren prema cilju koji pokreće pametna kolica u trgovini u *NetLogo*

4.1.2.4. Agenti usmjereni prema uspješnosti

Primjer 1.6: U ovom primjeru je opisan rad agenta usmjerenog prema uspješnosti Podsjetit ćemo se ranije navedenog pseudokoda agenta usmjerenog prema uspješnosti (Slika 26.)

```
function AGENT-USMJEREN-PREMA-USPJEŠNOSTI (opažaj)
  static: stanje /* opis trenutnog stanja okruženja */
           pravila /* skup unaprijed definiranih pravila stanje-akcija */
           cilj /* skup definiranih ciljanih stanja */

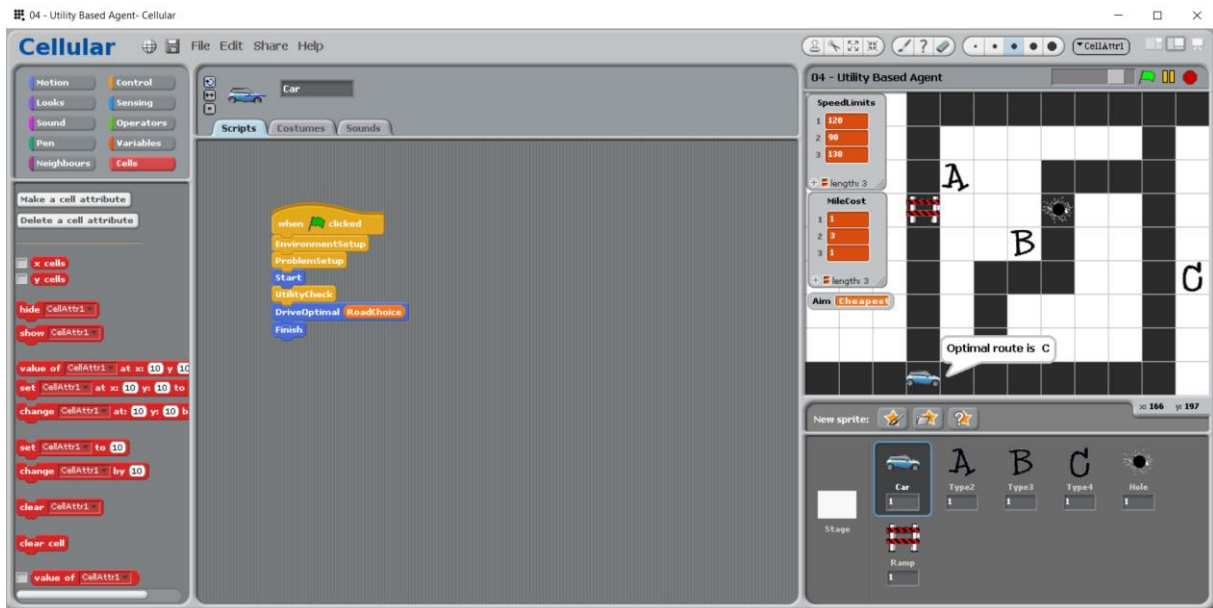
  stanje ← UPDATE-STATE(stanje, opažaj)
  pravilo ← RULE-MATCH(stanje, pravila)
  akcija ← RULE-ACTION(pravilo)
  stanje ← UPDATE-STATE(stanje, akcija)
  rezultat ← OBTAIN-SCORE(stanje)
  if (stanje in cilj) and BEST-SCORE(rezultat) then
    return akcija
  else
    opažaj ← OBTAIN-PERCEPT(stanje, cilj)
    return AGENT-USMJEREN-PREMA-USPJEŠNOSTI (opažaj)
```

Slika 29. Pseudokod agentske funkcije agenta usmjerenog prema uspješnosti

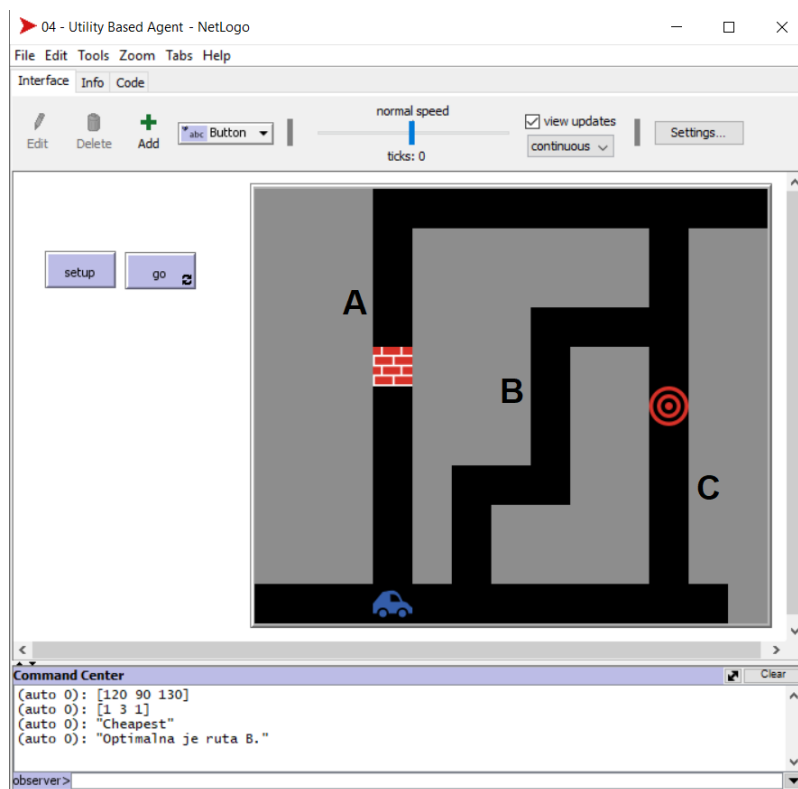
Agent određuje optimalnu rutu vozila. Na osnovu informacija o eventualnim preprekama na putu koje primaju senzori agenta, te najveće dozvoljene brzine na svakom od mogućih pravaca i cestarine koji se određuju slučajnim odabirom prilikom pokretanja simulacije, agent određuje optimalnu rutu vozila u skladu s primarnim ciljem. Primarni cilj može biti najbrži dolazak ili najjeftinija vožnja, što se također određuje slučajnim odabirom na početku simulacije (Prilog 6).

Agent koji određuje optimalnu rutu vozila razlikuje ciljane stanja od onih koja to nisu, ali posjeduje i mjerilo koliko je neko stanje poželjno. Ovo mjerilo je realizirano putem funkcije uspješnosti, koja preslikava pojedino stanje u realan rezultat, tj. mjeru njegove uspješnosti.

Studenti eksperimentalnih skupina su u simulaciji mogli mijenjati najveći broj prepreka koje se mogu pojaviti, kao i raspone najvećih dozvoljenih brzina i cestarina.



Slika 30. Agent usmjeren prema uspješnosti koji određuje optimalnu rutu vozila u *Cellularu*



Slika 31. Agent usmjeren prema uspješnosti koji određuje optimalnu rutu vozila u *NetLogu*

4.1.2.5. Ispitivanje znanja nakon NJ1 i analiza rezultata

Neposredno nakon obrađene prve nastavne jedinice sve tri skupine su istovremeno pristupile ispitivanju znanja.

Ispitivanja znanja, u trajanju od 30 minuta, uključivalo je zadatke otvorenog tipa čije je rješavanje zahtijevalo različite razine znanja definirane Bloomovom taksonomijom u kognitivnom području (Bloom, 1956), a studenti su na svakom od njih mogli ostvariti najviše 30 bodova. Studenti su upućeni u način rješavanja zadataka i upoznati s kriterijem bodovanja usklađenim s razinom potrebnog znanja za njihovo rješavanje (Prilog 18).

Za provjeru pouzdanosti ispita korišten je Cronbach α koeficijent. Ispitivanje znanja nakon obrađene prve nastaven jedinice spada u kategoriju visoko pouzdanih (Cronbach $\alpha = 0.847$).

Na osnovu postignutih rezultata izračunati su aritmetička sredina (AS), medijan (MED) i aritmetička sredina rangova (ASR) kao tri mjere srednje vrijednosti te standardna devijacija (SD) i raspon (R) kao dvije mjere disperzije. Analiza rezultata ispitivanja provjere znanja prikazana je u Tablici 5.

Tablica 5. Analiza rezultata ispitivanja znanja nakon NJ1

Statistička mjera	Kontrolna skupina (KS)	Eksperimentalna skupina 1 (ES1)	Eksperimentalna skupina 2 (ES2)
Aritmetička sredina (AS)	11,11	16,44	17,44
Medijan (MED)	11,00	15,00	15,00
Aritmetička sredina rangova (ASR)	7,72	16,22	18,06
Standardna devijacija (SD)	4,88	5,05	4,79
Raspon (R)	16,00	15,00	15,00

Postavljena je nulta hipoteza H_0 da ne postoji značajna razlika u postignućima tri skupine, odnosno da je raspodjela rezultata jednaka u svim skupinama.

Proveden je neparametrijski Kruskal-Wallisov test za tri nezavisne skupine kojim je testirana nulta hipoteza. U tu svrhu svi rezultati su rangirani po veličini. Rezultat analize, provedene uz pomoć statističkog paketa programske podrške SPSS, prikazan je u Tablici 6.

Tablica 6. Kruskal-Wallisov test rezultata ispitivanja znanja nakon NJ1 za tri nezavisne skupine

Nulta hipoteza H0	Provedeni test	Značajnost	Odluka
Raspodjela rezultata jednaka je u svim skupinama	Kruskal-Wallisov test za nezavisne uzorke	0,013	Odbaciti nultu hipotezu H0

Prikazana je asimptotska značajnost. Razina značajnosti je 0,05.

Statistička analiza rezultata ispitivanja znanja nakon prve nastavne jedinice odbacila je nultu hipotezu H0. Time je pokazano da postoji statistički značajna razlika među skupinama u poznavanju gradiva prve nastavne jedinice, odnosno da raspodjela postignutih rezultata nije jednaka u sve tri skupine.

Rezultati su dodatno testirani kako bi se preciznije utvrdilo između kojih skupina studenata postoji utvrđena razlika. U tu svrhu su postavljene nulte hipoteze H0 da nema značajnije razlike u postignućima za svaki par skupina (KS i ES1, KS i ES2, ES1 i ES2). Ove hipoteze su testirane neparametrijskim Mann-Whitneyevim U testom za dvije nezavisne skupine. Rezultati analiza provedenih uz pomoć statističkog softverskog paketa SPSS prikazani su u Tablici 7.

Tablica 7. Mann-Whitneyev U test rezultata ispitivanja znanja nakon NJ1 za parove nezavisnih skupina

Nulta hipoteza H0	Provedeni test	Značajnost	Odluka
Raspodjela rezultata jednaka je u KS i ES1	Mann-Whitneyev U test za dvije nezavisne skupine	0,019	Odbaciti nultu hipotezu H0
Raspodjela rezultata jednaka je u KS i ES2	Mann-Whitneyev U test za dvije nezavisne skupine	0,006	Odbaciti nultu hipotezu H0
Raspodjela rezultata jednaka je u ES1 i ES2	Mann-Whitneyev U test za dvije nezavisne skupine	0,605	Zadržati nultu hipotezu H0

Prikazana je asimptotska značajnost. Razina značajnosti je 0,05.

Statistička analiza rezultata ispitivanja znanja nakon obrađene prve nastavne jedinice pokazala je da postoji značajna razlika u postignućima studenata dvije eksperimentalne skupine u odnosu na kontrolnu skupinu studenata. Također je vidljivo da ne postoji značajnija razlika u postignućima studenata dvije eksperimentalne skupine. Prikazane mjere srednje vrijednosti postignutih rezultata (AS, MED i ASR) ukazuju na činjenicu da su bolje rezultate za gradivo prve nastavne jedinice ostvarili studenti eksperimentalnih skupina, koji su tijekom nastave koristili jedno od odabranih simulacijskih okruženja.

4.1.3. Definicija, osnovna svojstva i vrste okruženja (NJ2)

U drugoj nastavnoj jedinici (NJ2) studenti su se upoznali s okruženjima u kojima agenti djeluju. Iako se prilikom dizajniranja rješenja umjetne inteligencije zasnovanim na agentima naglasak

stavlja na osobine realiziranih agenata ili prirodu podataka kojima oni manipuliraju, karakteristike okruženja agenta često jedan su od ključnih elemenata za određivanje pravog modela rješenja.

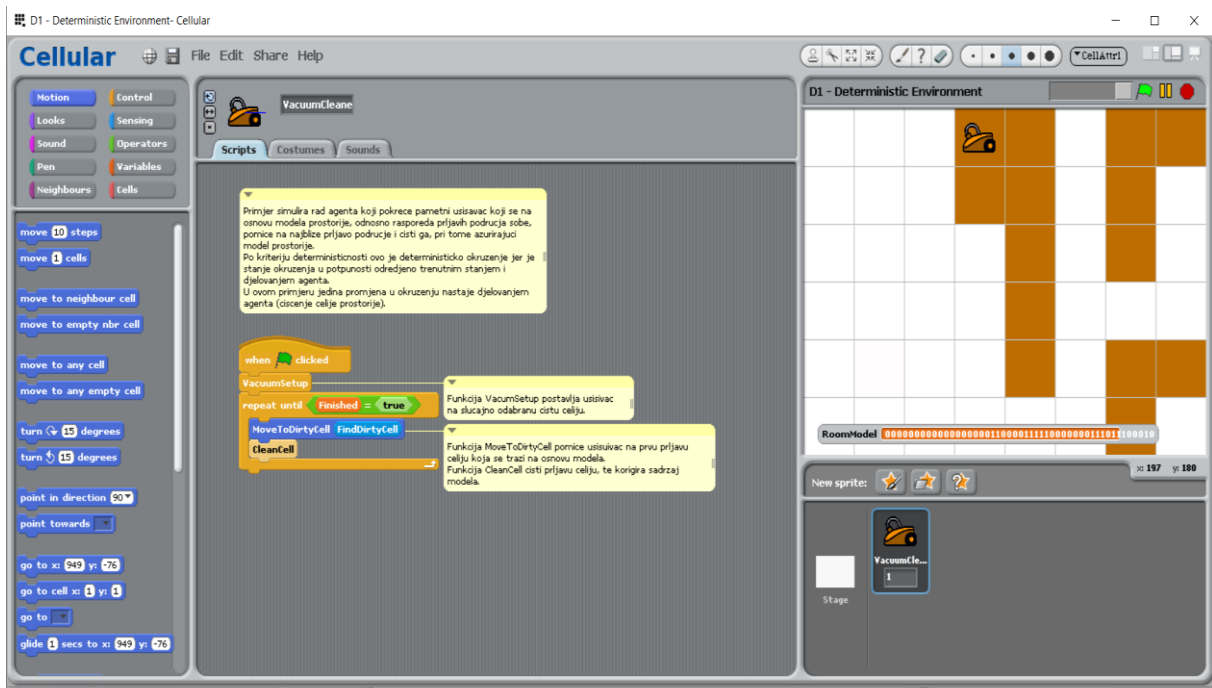
Vodeći se klasifikacijom okruženja agenata koju predlažu Russell i Norvig (2016), određenom osnovnim karakteristikama okruženja koje, između ostalog, nameću priroda problema, opseg dostupnih informacija i znanja o okruženju i promjene koje u okruženju nastaju djelovanjem agenta, studente smo upoznali sa sedam kriterija za klasificiranje okruženja umjetne inteligencije (*D-SOAKED*), od kojih svaki dijeli okruženja u dva disjunktna podskupa:

- determinističnost;
- statičnost;
- dostupnost informacija;
- broj agenata;
- znanje o okruženju;
- epizodičnost;
- diskretnost.

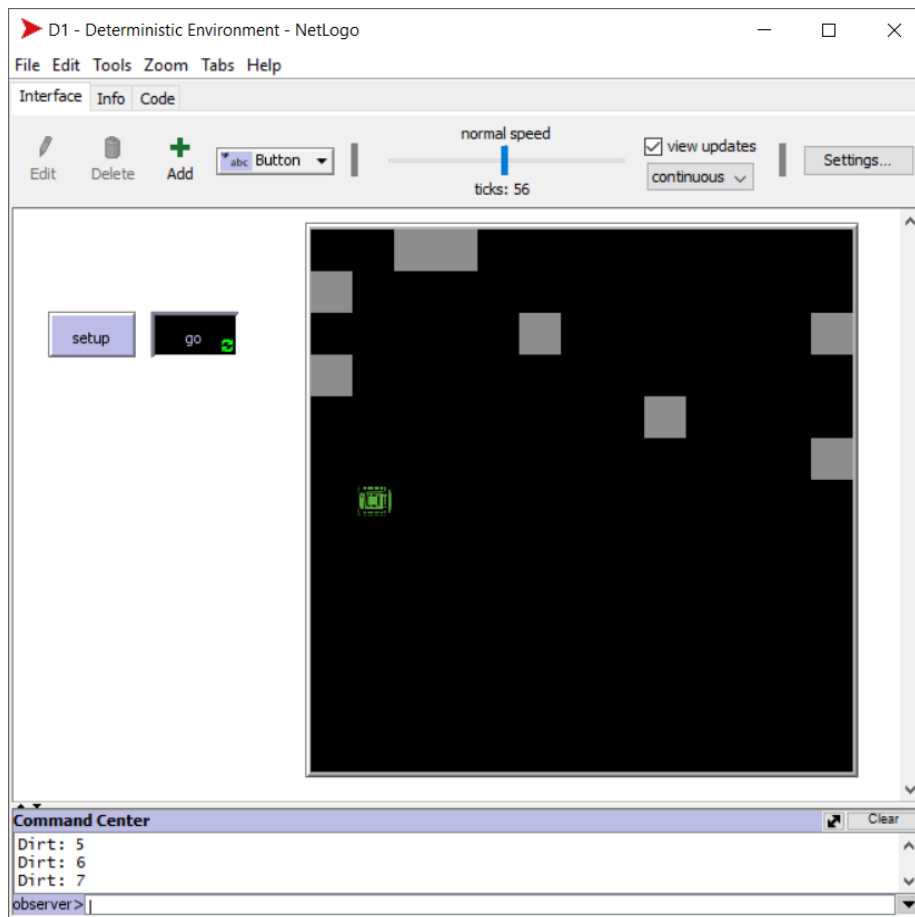
Zbog vremenskog ograničenja, u ovoj nastavnoj jedinici definirani su svi kriteriji i dani primjeri okruženja za svaki od njih. Posebna pažnja posvećena je kriterijima determinističnosti, dostupnosti informacija i epizodičnosti, odnosno njihove karakteristike su dodatno obrazložene kroz pripremljene prigodne primjere okruženja koja po ovim kriterijima klasifikacije pripadaju različitim klasama okruženja. Za kontrolnu skupinu primjeri su prikazani korištenjem prezentacije i projektora, dok su obje eksperimentalne skupine samostalno pokrenule, koristile i u manjoj mjeri manipulirale primjerima realiziranim u simulacijskim okruženjima alata *Cellular* i *NetLogo*. Pri tome su primjeri za sve tri skupine opisivali okruženja istovjetnih karakteristika. Detaljan opis primjera, kojima su se studenti upoznali s različitim vrstama okruženja, njihovim karakteristikama i specifičnostima dat je u nastavku.

4.1.3.1. Kriterij determinističnosti

Primjer 2.1: U ovom primjeru je opisan rad agenta koji upravlja radom pametnog usisavača. Na osnovu modela prostorije, odnosno rasporeda prljavih područja prostorije, agent se pomiče na najbliže prljavo područje i čisti ga, pri tome ažurirajući model prostorije (Prilog 7).



Slika 32. Determinističko okruženje po kriteriju determinističnosti u *Cellularu*



Slika 33. Determinističko okruženje po kriteriju determinističnosti u *NetLogo*

Po kriteriju determinističnosti ovo je determinističko okruženje jer je stanje okruženja u potpunosti određeno trenutnim stanjem i djelovanjem agenta koji se nalazi u okruženju.

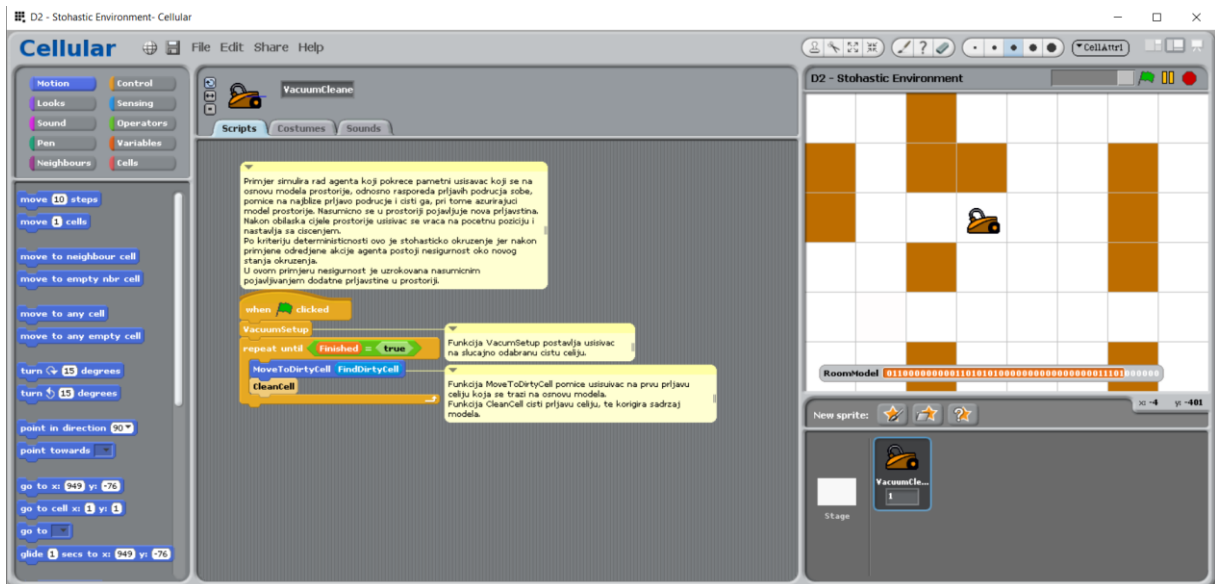
Jedina promjena u okruženju nastaje djelovanjem agenta (čišćenje ćelije prostorije). U ovakvom okruženju agent ne treba voditi računa o neizvjesnosti koja je posljedica prirode okruženja.

Studenti eksperimentalnih skupina su u simulaciji mogli mijenjati broj inicijalno zadanih prljavih dijelova prostorije.

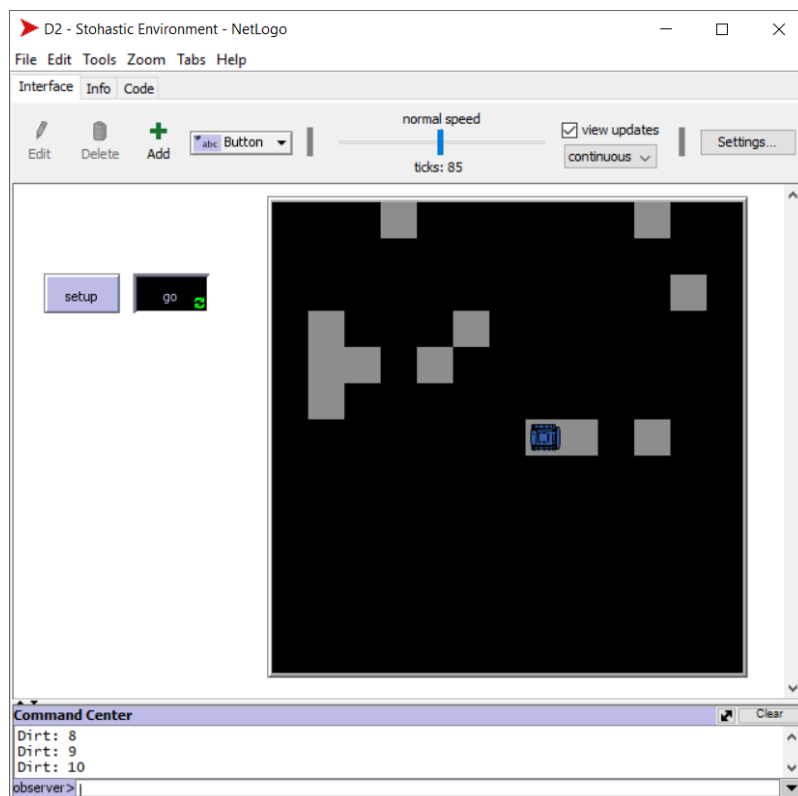
Primjer 2.2: U ovom primjeru je, također, opisan rad agenta koji upravlja radom pametnog usisavača. Razlika u odnosu na prethodni primjer je ta što se tijekom čišćenja u prostoriji nasumično pojavljuje nova prljavština. Zbog toga nakon obilaska cijele prostorije usisivač se vraća na početnu poziciju i nastavlja s čišćenjem sve dok ne očisti cijelu prostoriju. I ovdje se agent kreće na osnovi modela prostorije, odnosno rasporeda prljavih područja prostorije. Nakon što se pomakne na najbliže prljavo područje, agent ga očisti i ažurira model prostorije, u skladu sa svojom akcijom (Prilog 8).

Po kriteriju determinističnosti ovo je stohastičko okruženje jer nakon primjene određene akcije agenta postoji nesigurnost oko novog stanja okruženja. U ovom primjeru nesigurnost je uzrokovana nasumičnim pojavljivanjem dodatne prljavštine u prostoriji.

Studenti eksperimentalnih skupina su u simulaciji mogli mijenjati broj inicijalno zadanih prljavih dijelova prostorije, kao i vjerojatnost pojavljivanja novih prljavih dijelova.



Slika 34. Stohastičko okruženje prema kriteriju determinističnosti u *Cellularu*



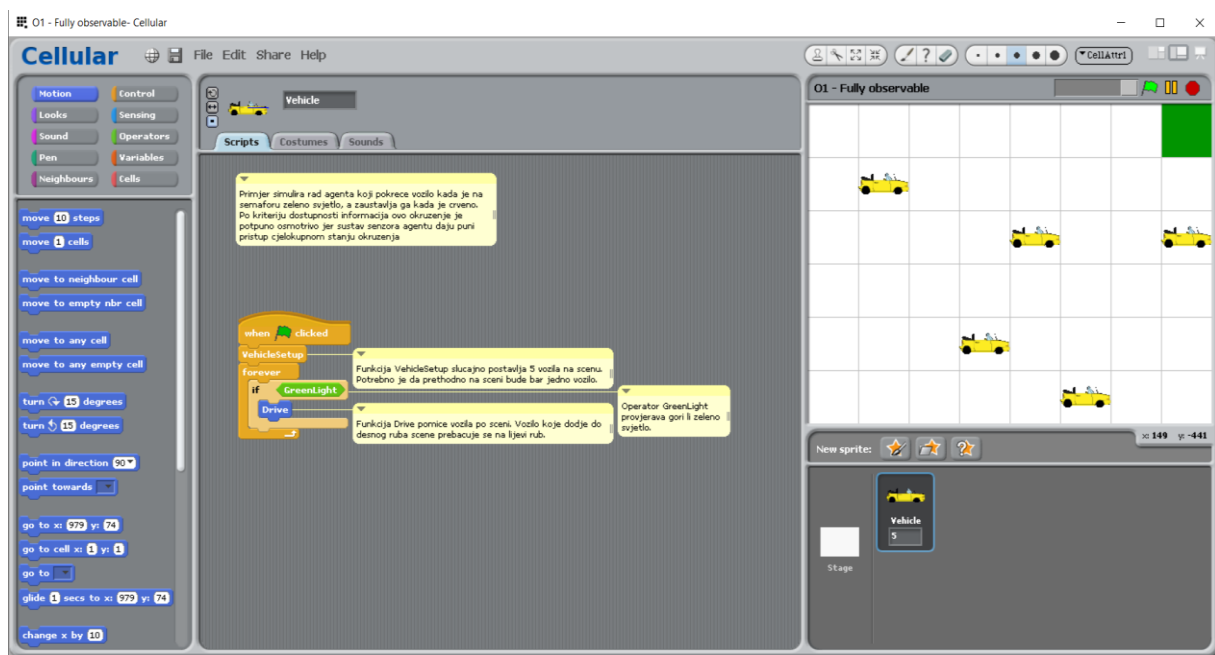
Slika 35. Stohastičko okruženje prema kriteriju determinističnosti u *NetLogu*

4.1.3.2. Kriterij dostupnosti informacija

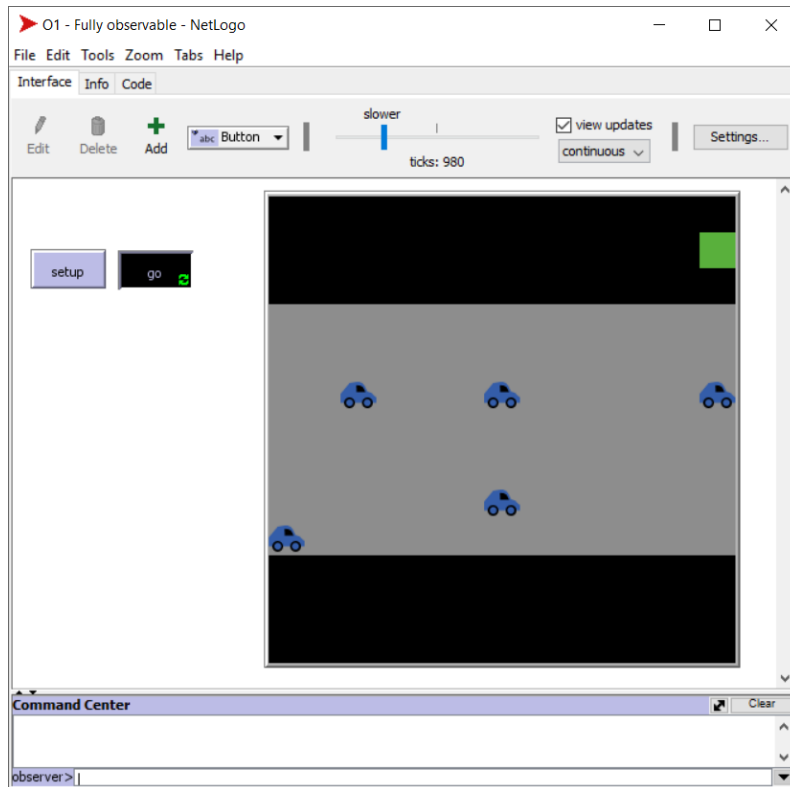
Primjer 2.3: U ovom primjeru je opisan rad agenta koji pokreće autonomno vozilo. Kada senzori agenta opaze da je na semaforu zeleno svjetlo, vozilo se pokreće i vozi sve dok senzori agenta ne opaze crveno svjetlo na semaforu. U tom trenutku vozilo se zaustavlja do ponovnog opažanja zelenog svjetla (Prilog 9).

Po kriteriju dostupnosti informacija ovo okruženje je potpuno vidljivo jer sustav senzora agentu daje puni pristup cjelokupnom stanju okruženja. Senzorski aparat agenta može detektirati sve aspekte okruženja relevantne za njegov izbor akcije. U ovom primjeru radi se samo o svjetlu koje je trenutno upaljeno na semaforu. Potpuno vidljiva okruženja su pogodna za implementaciju rješenja zasnovanih na agentima, jer agent ne treba održavati bilo kakvo unutrašnje stanje kojim bi pratio svijet oko sebe.

Studenti eksperimentalnih skupina su u simulaciji mogli mijenjati broj vozila u okruženju, kao i interval trajanja crvenog odnosno zelenog svjetla na semaforu.

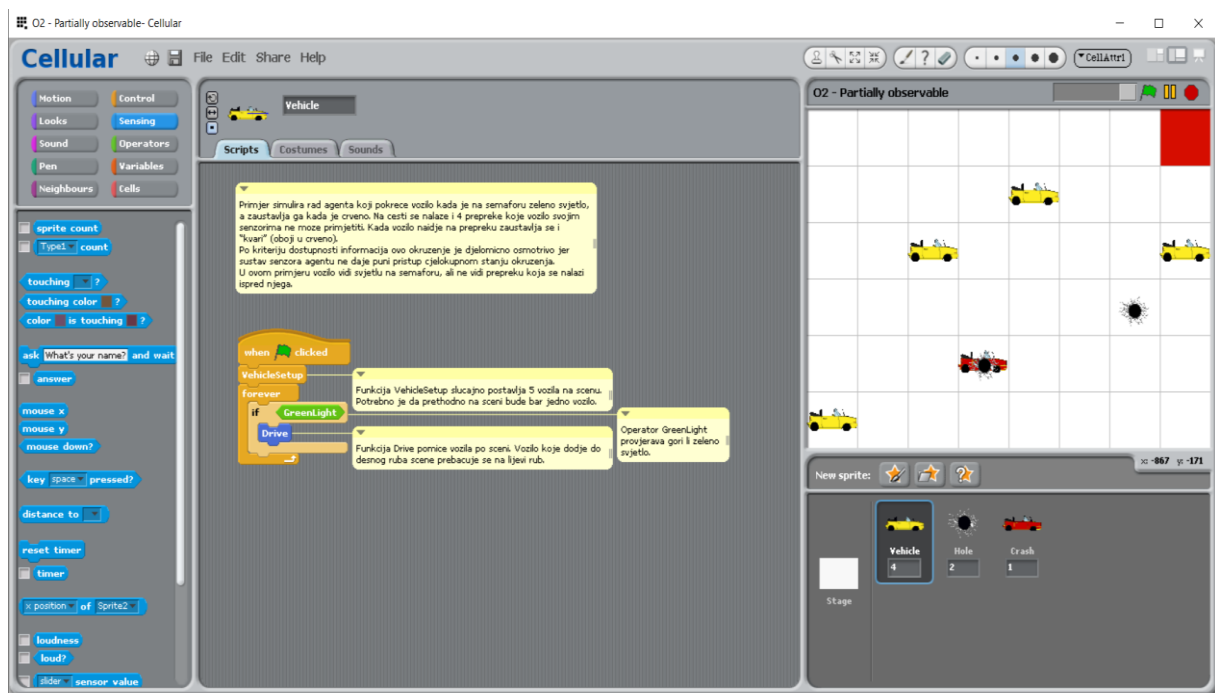


Slika 36. Potpuno vidljivo okruženje po kriteriju dostupnosti informacija u *Cellularu*



Slika 37. Potpuno vidljivo okruženje po kriteriju dostupnosti informacija u *NetLogo*

Primjer 2.4: U ovom primjeru je, ponovno, opisan rad agenta koji pokreće autonomno vozilo i reagira na promjenu svjetla na semaforu.

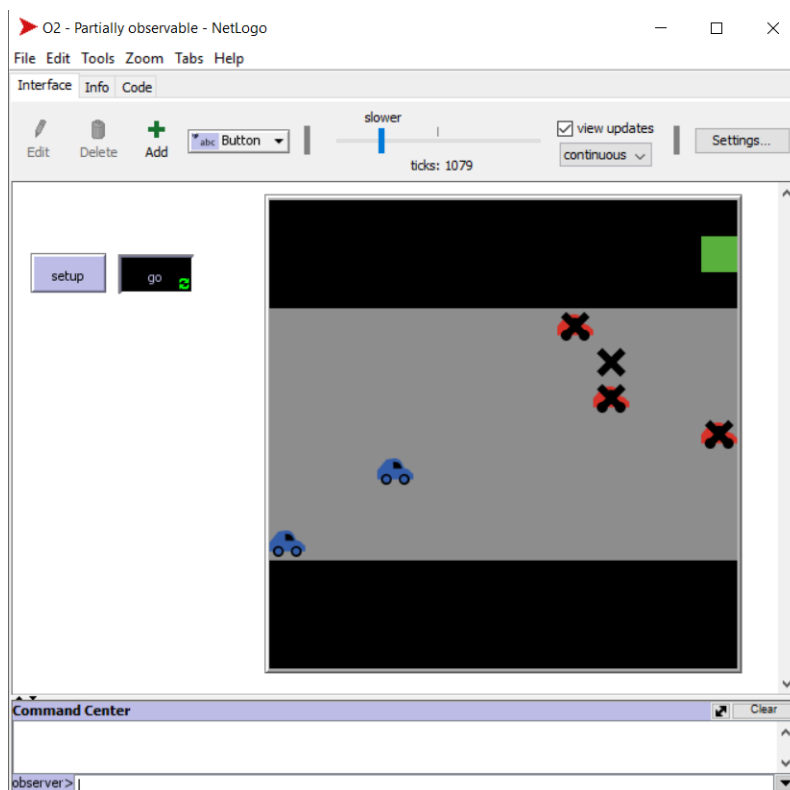


Slika 38. Djelomično vidljivo okruženje po kriteriju dostupnosti informacija u *Cellularu*

Razlika u odnosu na prethodni primjer je postojanje prepreka ili oštećenja na putu koje senzorski aparat agenta ne može opaziti. Kada vozilo naiđe na prepreku, kvari se, tj. zaustavlja (Prilog 10).

Po kriteriju dostupnosti informacija ovo okruženje je djelomično vidljivo jer sustav senzora agentu ne daje puni pristup cjelokupnom stanju okruženja. U ovom primjeru vozilo vidi svjetlo na semaforu, ali ne vidi prepreku koja se nalazi ispred njega.

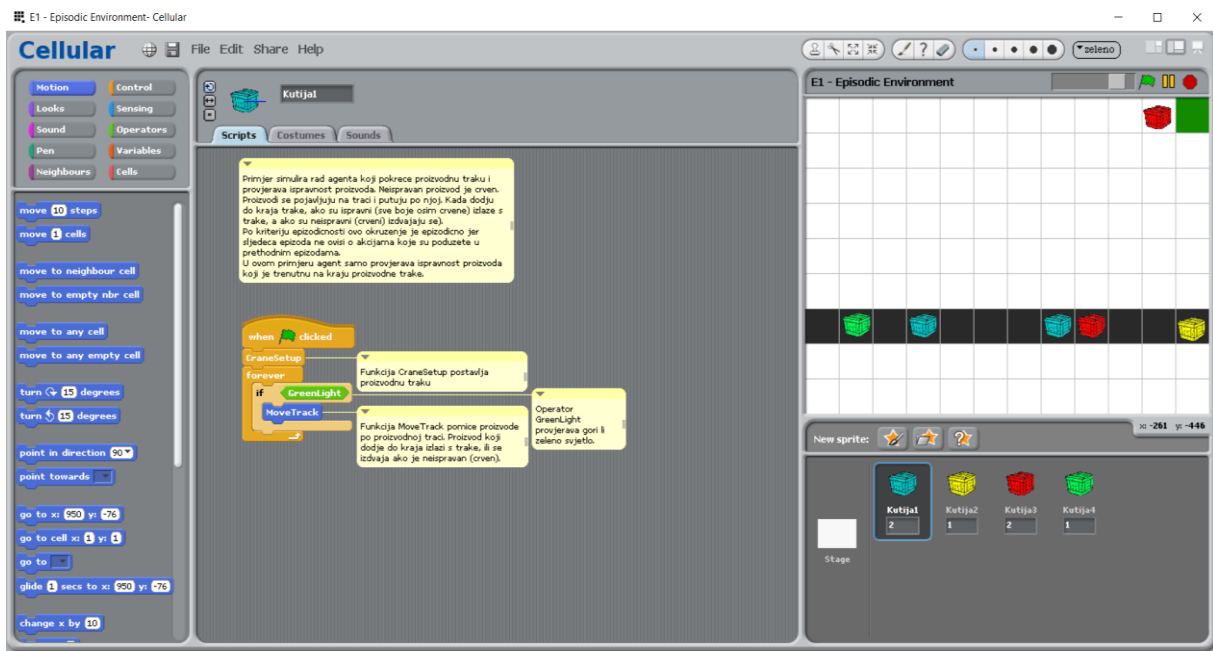
Studenti eksperimentalnih skupina su u simulaciji mogli mijenjati broj vozila u okruženju, broj prepreka koje se pojavljuju na putu, kao i interval trajanja crvenog odnosno zelenog svjetla na semaforu.



Slika 39. Djelomično vidljivo okruženje po kriteriju dostupnosti informacija u *NetLogo*

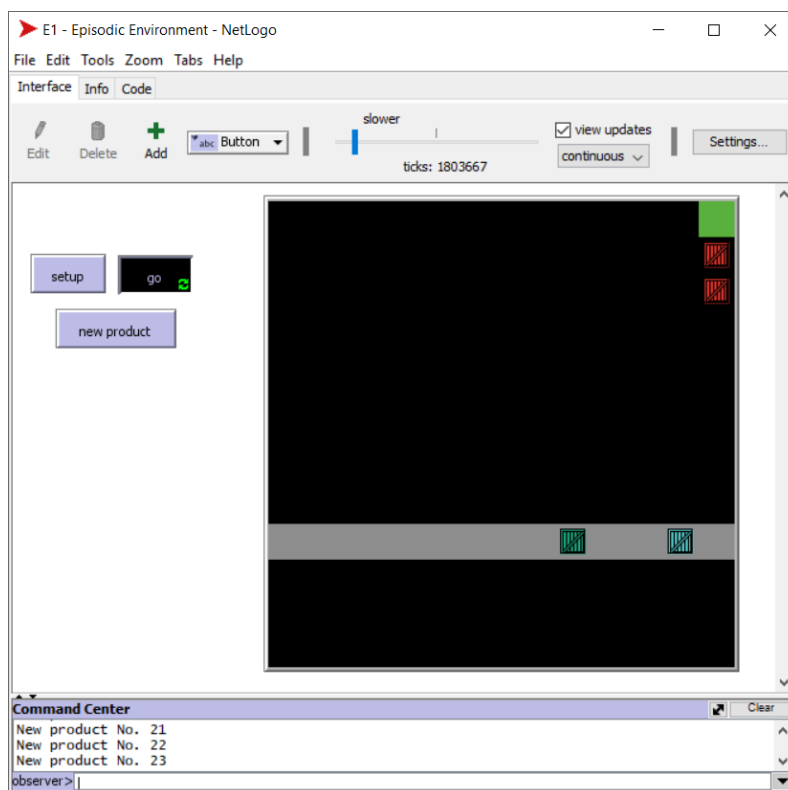
4.1.3.3. Kriterij epizodičnosti

Primjer 2.5: U ovom primjeru opisan je rad agenta koji nadgleda proizvodnu traku i proizvode na njoj. Agent registrira neispravne proizvode (obojene crvenom bojom) i ispravne proizvode (obojene nekom drugom bojom) kada dođu do kraja proizvodne trake. Ispravni proizvodi silaze s trake (uklanjaju se s ekrana), a neispravni se izdvajaju (zadržavaju na ekranu). Proizvodnu seriju čini 30 proizvoda i simulacija se prekida kada do kraja trake dođe 30 proizvoda, bez obzira koliko je bilo neispravnih proizvoda (Prilog 11).



Slika 40. Epizodično okruženje po kriteriju epizodičnosti u *Cellularu*

Po kriteriju epizodičnosti ovo okruženje je epizodično jer buduće odluke agenta ne ovise o akcijama koje je agent prethodno poduzeo, nego samo o informacijama dobivenim putem senzora o stanju u kojem se okruženje nalazi. U ovom primjeru registrira se ispravnost ili neispravnost proizvoda, ne vodeći računa kakvi su rezultati bili prije toga.



Slika 41. Epizodično okruženje po kriteriju epizodičnosti u *NetLogu*

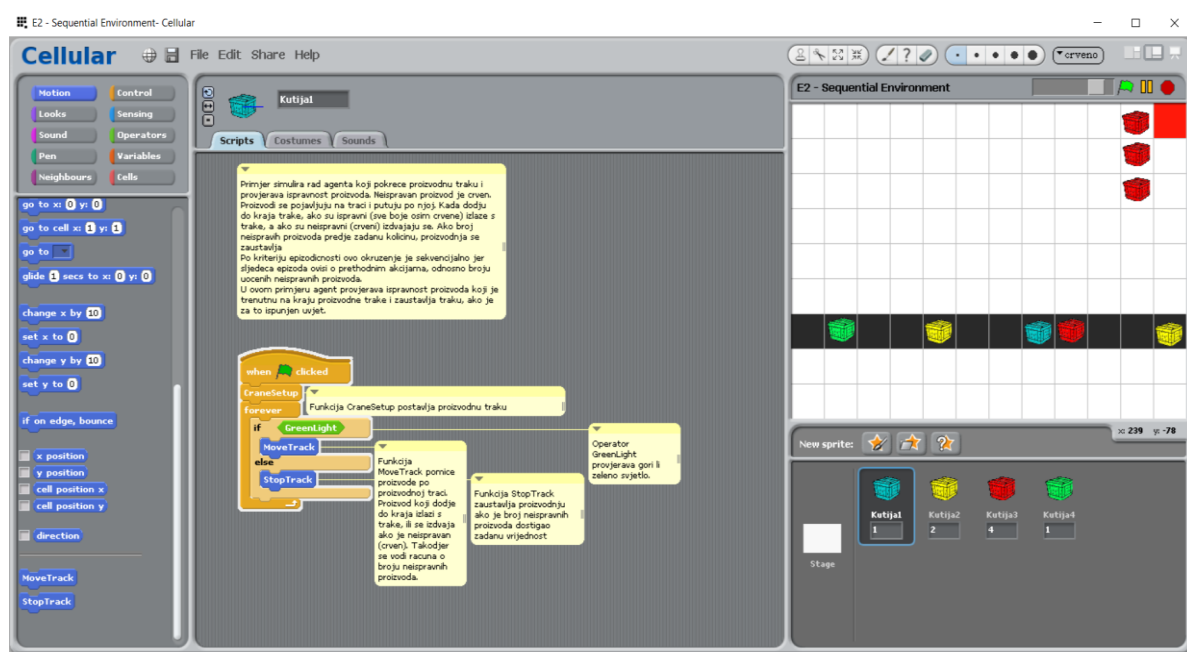
Studenti eksperimentalnih skupina su u simulaciji upravljali dodavanjem novog proizvoda na proizvodnu traku, te su mogli mijenjati veličinu proizvodne serije.

Primjer 2.6: U ovom primjeru je ponovno opisan rad agenta koji nadgleda proizvodnu traku i proizvode na njoj, odnosno registrira ih kao neispravne (obojene crvenom bojom) ili ispravne proizvode (obojene nekom drugom bojom) kada dođu do kraja proizvodne trake. Ispravni proizvodi se uklanjaju s ekrana, a neispravni zadržavaju na ekranu (Prilog 12).

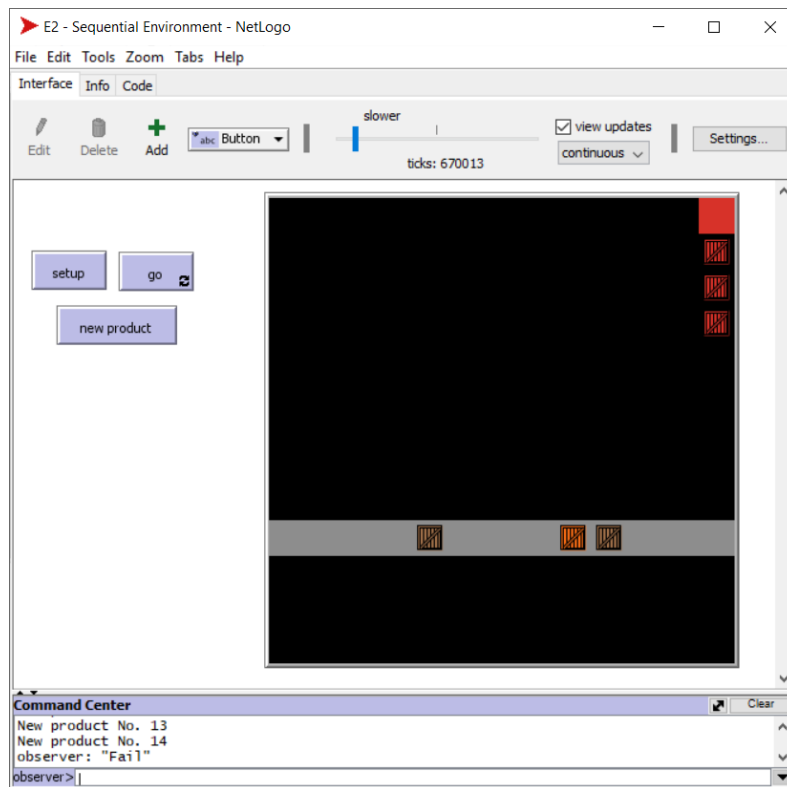
Seriju proizvoda čini 30 proizvoda i ako je manje od 10% proizvoda neispravno serija se smatra uspješnom, a ako je 10% ili više neispravnih proizvoda serija se smatra neuspješnom.

Po kriteriju epizodičnosti ovo okruženje je sekvencijalno jer buduće odluke agenta ovise o akcijama koje je agent prethodno poduzeo, odnosno trenutna akcija ima dugoročne posljedice. U ovom primjeru zaustavlja se proizvodnja u trenutku kad je serija označena kao neispravna (10% neispravnih proizvoda).

Studenti eksperimentalnih skupina su u simulaciji upravljali dodavanjem novog proizvoda na proizvodnu traku, te se mogli mijenjati veličinu proizvodne serije i granicu koja određuje hoće li se serija smatrati ispravnom ili neispravnom.



Slika 42. Sekvencijalno okruženje po kriteriju epizodičnosti u u *Cellularu*



Slika 43. Sekvencijalno okruženje po kriteriju epizodičnosti u *NetLogo*

4.1.3.4. Ispitivanje znanja nakon NJ2 i analiza rezultata

Neposredno nakon obrađene druge nastavne jedinice sve tri skupine su istovremeno pristupile ispitivanju znanja.

Ispitivanja znanja, u trajanju od 30 minuta, uključivalo je zadatke otvorenog tipa čije je rješavanje zahtijevalo različite razine znanja definirane Bloomovom taksonomijom u kognitivnom području (Bloom, 1956), a studenti su na svakom od njih mogli ostvariti najviše 30 bodova. Studenti su upućeni u način rješavanja zadataka i upoznati s kriterijem bodovanja usklađenim s razinom potrebnog znanja za njihovo rješavanje (Prilog 19).

Za provjeru pouzdanosti ispita korišten je Cronbach α koeficijent. Ispitivanje znanja nakon obrađene prve nastaven jedinice spada u kategoriju pouzdanih (Cronbach $\alpha = 0.764$).

Na osnovi postignutih rezultata izračunati su aritmetička sredina (AS), medijan (MED) i aritmetička sredina rangova (ASR) kao tri mjere srednje vrijednosti te standardna devijacija (SD) i raspon (R) kao dvije mjere disperzije. Analiza rezultata ispitivanja provjere znanja prikazana je u Tablici 8.

Tablica 8. Analiza rezultata ispitivanja znanja nakon NJ2

Statistička mjera	Kontrolna skupina (KS)	Ekperimentalna skupina 1 (ES1)	Ekperimentalna skupina 2 (ES2)
Aritmetička sredina (AS)	10,44	15,89	15,39
Medijan (MED)	9,00	16,50	16,50
Aritmetička sredina rangova (ASR)	8,39	17,28	16,33
Standardna devijacija (SD)	4,72	5,69	5,34
Raspon (R)	13,00	17,50	16,50

Postavljena je nulta hipoteza H_0 da ne postoji značajna razlika u postignućima tri skupine, odnosno da je raspodjela rezultata jednaka u svim skupinama.

Proveden je neparametrijski Kruskal-Wallisov test za tri nezavisne skupine kojim je testirana nulta hipoteza. U tu svrhu svi rezultati su rangirani po veličini. Rezultat analize, provedene uz pomoć statističkog paketa programske podrške SPSS, prikazan je u Tablici 9.

Tablica 9. Kruskal-Wallisov test rezultata ispitivanja znanja nakon NJ2 za tri nezavisne skupine

Nulta hipoteza H_0	Provedeni test	Značajnost	Odluka
Raspodjela rezultata jednaka je u svim skupinama	Kruskal-Wallisov test za nezavisne uzorke	0,033	Odbaciti nultu hipotezu H_0

Prikazana je asimptotska značajnost. Razina značajnosti je 0,05.

Statistička analiza rezultata ispitivanja znanja nakon druge nastavne jedinice odbacila je nultu hipotezu H_0 . Time je pokazano je da postoji statistički značajna razlika među skupinama u poznavanju gradiva druge nastavne jedinice, odnosno da raspodjela postignutih rezultata nije jednaka u sve tri skupine.

Rezultati su dodatno testirani kako bi se preciznije utvrdilo između kojih skupina studenata postoji utvrđena razlika. U tu svrhu su postavljene nulte hipoteze H_0 da nema značajnije razlike u postignućima za svaki par skupina (KS i ES1, KS i ES2, ES1 i ES2). Ove hipoteze su testirane neparametrijskim Mann-Whitneyevim U testom za dvije nezavisne skupine. Rezultati analiza provedenih uz pomoć statističkog softverskog paketa SPSS prikazani su u Tablici 10.

Tablica 10. Mann-Whitneyev U test rezultata ispitivanja znanja nakon NJ2 za parove nezavisnih skupine

Nulta hipoteza H0	Provedeni test	Značajnost	Odluka
Raspodjela rezultata jednaka je u KS i ES1	Mann-Whitneyev U test za dvije nezavisne skupine	0,019	Odbaciti nultu hipotezu H0
Raspodjela rezultata jednaka je u KS i ES2	Mann-Whitneyev U test za dvije nezavisne skupine	0,031	Odbaciti nultu hipotezu H0
Raspodjela rezultata jednaka je u ES1 i ES2	Mann-Whitneyev U test za dvije nezavisne skupine	0,796	Zadržati nultu hipotezu H0

Prikazana je asimptotska značajnost. Razina značajnosti je 0,05.

Statistička analiza rezultata ispitivanja znanja nakon obrađene druge nastavne jedinice pokazala je da postoji značajna razlika u postignućima studenata dvije eksperimentalne skupine u odnosu na kontrolnu skupinu studenata. Također je vidljivo da ne postoji značajnija razlika u postignućima studenata dvije eksperimentalne skupine. Prikazane mjere srednje vrijednosti postignutih rezultata (AS, MED i ASR) ukazuju na činjenicu da su bolje rezultate za gradivo druge nastavne jedinice ostvarili studenti eksperimentalnih skupina, koji su tijekom nastave koristili jedno od odabranih simulacijskih okruženja.

4.1.4. Interakcija agenata i okruženja i međusobna interakcija agenata u višeagentskim sustavima (NJ3)

U trećoj nastavnoj jedinici (NJ3) posvećena je pažnja interakcijama koje se događaju u agentskim i višeagentskim sustavima.

Interakcija agenata i okruženja je relativno jednostavna i jednoobrazna. Sastoji se od djelovanja okruženja na senzorni aparat agenta kroz podražaje i djelovanja agenata na okruženje putem aktuatora, a na osnovu primljenog podražaja iz okruženja i realizirane agentske funkcije. Kako je ovaj oblik interakcije detaljnije objašnjen u prve dvije nastavne jedinice, u trećoj nastavnoj jedinici pažnja je posvećena međusobnoj interakciji agenata u višeagentskim sustavima.

Interakcija među agentima u višeagentskom sustavu se razvijaju iz niza akcija pojedinih agenata, čije posljedice utječu na buduće ponašanje agenata. Njihov oblik je određen, u najvećoj mjeri, ciljevima i sposobnostima pojedinačnih agenata te raspoloživim resursima u okruženju višeagentskog sustava (Ferber, 1999).

Interakciju kvalificiramo kao pozitivnu ili negativnu, pri čemu kod pozitivne interakcije djelovanje jednog agenta (ili skupine agenata) ne utječe negativno na djelovanje drugog agenta (ili skupine agenata), dok kod negativne interakcija uspješno djelovanje jednog agenta (ili skupine agenata) uzrokuje neuspješno djelovanje drugog agenta (ili skupine agenata).

Razlikujemo sljedeće vrste interakcije u višeagentskim sustavima, zasnovane na ciljevima i sposobnostima agenata te resursima dostupnim u okruženju (Ferber, 1999):

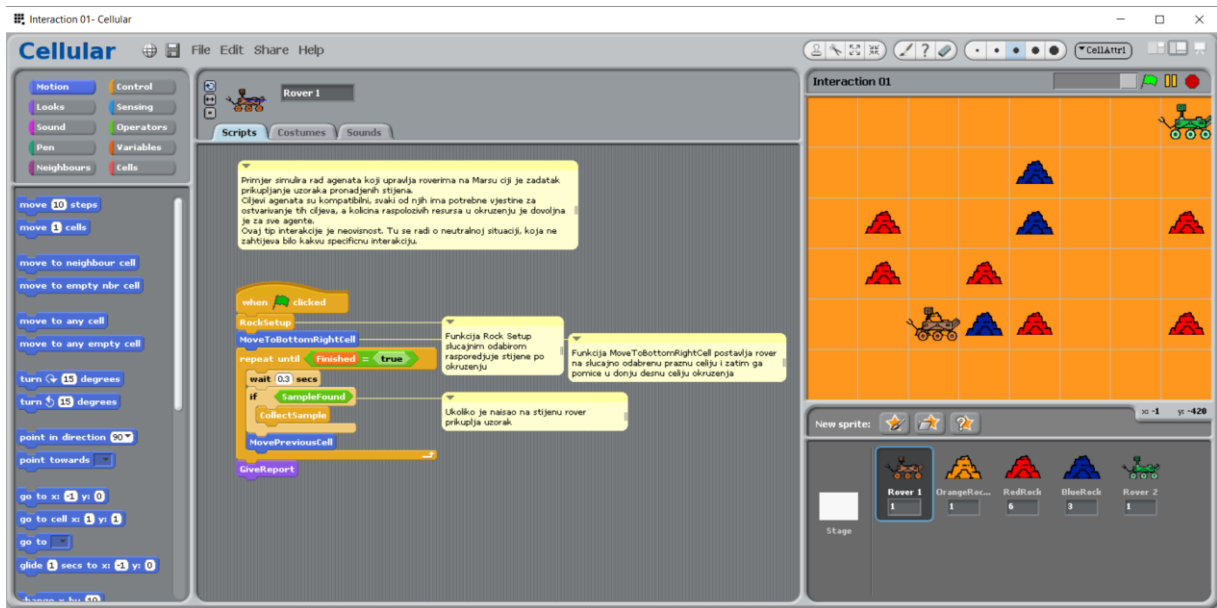
- neovisnost;
- jednostavna suradnja;
- opstrukcija;
- koordinirana suradnja;
- čisto pojedinačno natjecanje;
- čisto kolektivno natjecanje;
- pojedinačni sukob oko resursa;
- kolektivni sukob oko resursa.

Uslijed vremenskog ograničenja, u ovoj nastavnoj jedinici definirani su svi oblici interakcija i dani primjeri višeagentskih sustava i situacija u kojima se pojedine interakcije javljaju. Posebna pažnja posvećena je neovisnosti, jednostavnoj suradnji, opstrukciji i čistom pojedinačnom natjecanju, a njihove karakteristike i posebnosti su dodatno obrazložene kroz pripremljene prigodne primjere višeagentskih sustava u kojima se javljaju ovi oblici interakcija. Za kontrolnu skupinu primjeri su prikazani korištenjem prezentacije i projektoru, dok su obje eksperimentalne skupine samostalno pokrenule, koristile i u manjoj mjeri manipulirale primjerima realiziranim u simulacijskim okruženjima alata *Cellular* i *NetLogo*. Pri tome su primjeri za sve tri skupine opisivali višeagentske sustave istovjetnih karakteristika. Detaljan opis primjera, kojima su se studenti upoznali s različitim vrstama interakcija u višeagentskim sustavima, njihovim karakteristikama i specifičnostima dat je u nastavku.

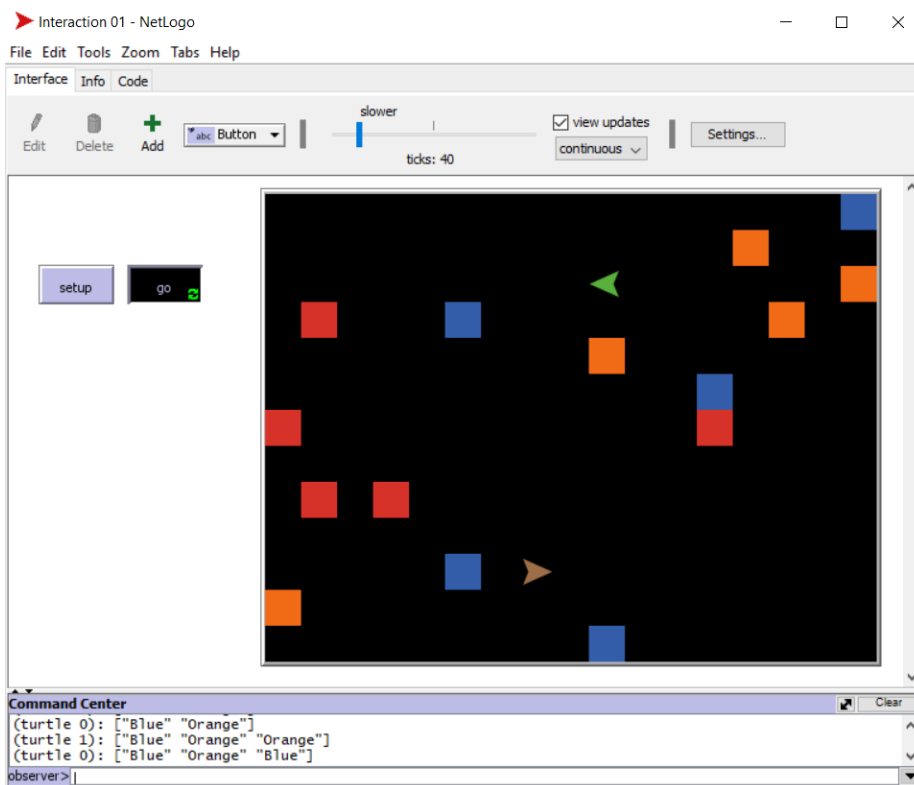
4.1.4.1. Neovisnost

Primjer 3.1: U ovom primjeru je opisan rad agenata koji upravljaju roverima na Marsu čiji je zadatak prikupljanje uzoraka pronađenih stijena. Roveri se kreću unaprijed definiranim putanjama i ukoliko naiđu na stijenu uzimaju njen uzorak (Prilog 13).

Ciljevi agenata su kompatibilni, svaki od njih ima potrebne vještine za ostvarivanje tih ciljeva, a količina raspoloživih resursa u okruženju je dovoljna za sve agente.



Slika 44. Neovisnost u višeagentskom sustavu u *Cellularu*



Slika 45. Neovisnost u višeagentskom sustavu u *NetLogu*

Ovaj tip interakcije je neovisnost. Tu se radi o neutralnoj situaciji koja ne zahtijeva bilo kakvu specifičnu interakciju.

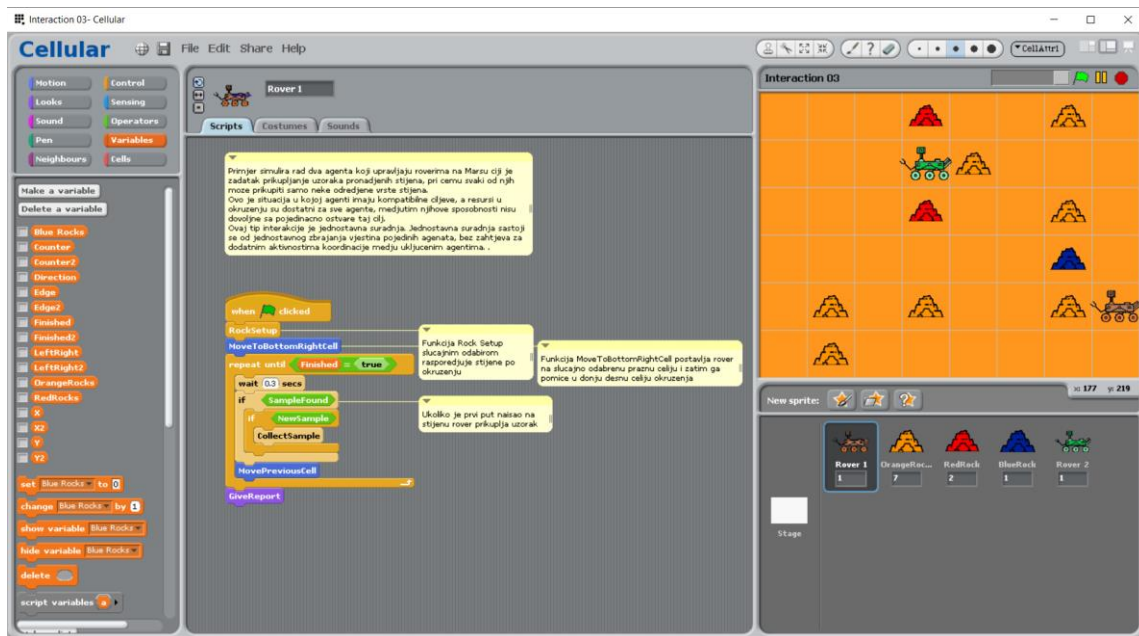
Studenti eksperimentalnih skupina su u simulaciji mogli mijenjati broj primjeraka svake vrste stijena koje se inicijalno javljaju u okruženju.

4.1.4.2. Jednostavna suradnja

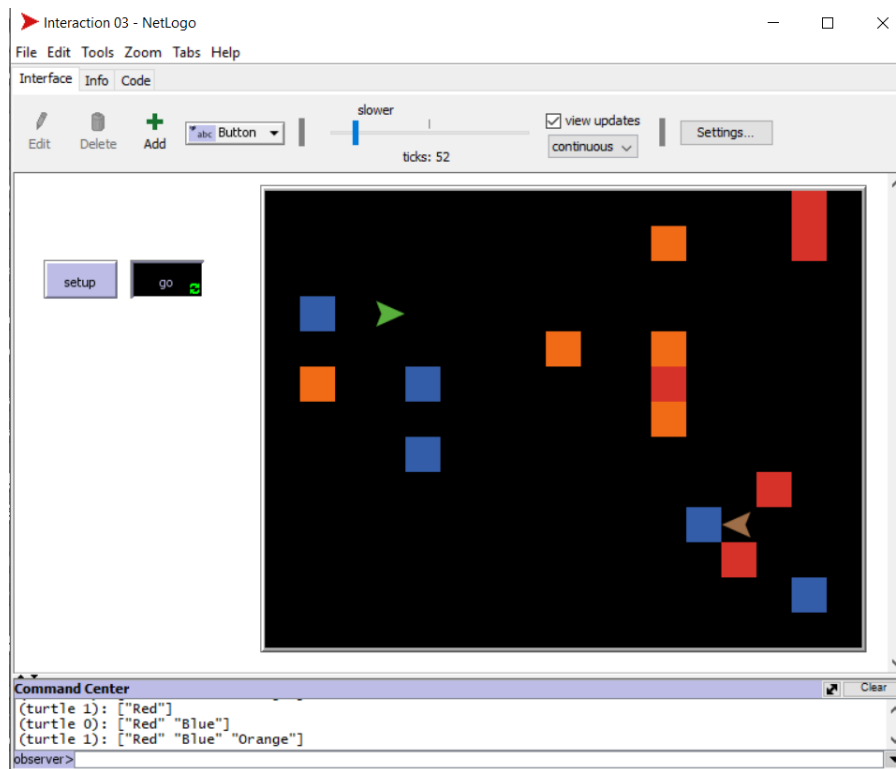
Primjer 3.2: U ovom primjeru je ponovno opisan rad agenata koji upravljaju roverima na Marsu, a svaki od njih ima zadatak prikupljanje uzoraka određenih stijena. Roveri se kreću unaprijed definiranim putanjama i ukoliko naiđu na stijenu odgovarajuće vrste, uzimaju njen uzorak (Prilog 14).

Ciljevi agenata su kompatibilni a količina raspoloživih resursa u okruženju je dovoljna za sve agente. Za razliku od prethodnog primjera, sposobnosti pojedinih agenata nisu dovoljne da pojedinačno ostvare svoj cilj. Mogu ga ostvariti samo suradnjom.

Ovaj tip interakcije je jednostavna suradnja. Jednostavna suradnja sastoji se od jednostavnog zbrajanja vještina pojedinih agenata, bez zahtjeva za dodatnim aktivnostima koordinacije među uključenim agentima.



Slika 46. Jednostavna suradnja u višeagentskom sustavu u *Cellularu*

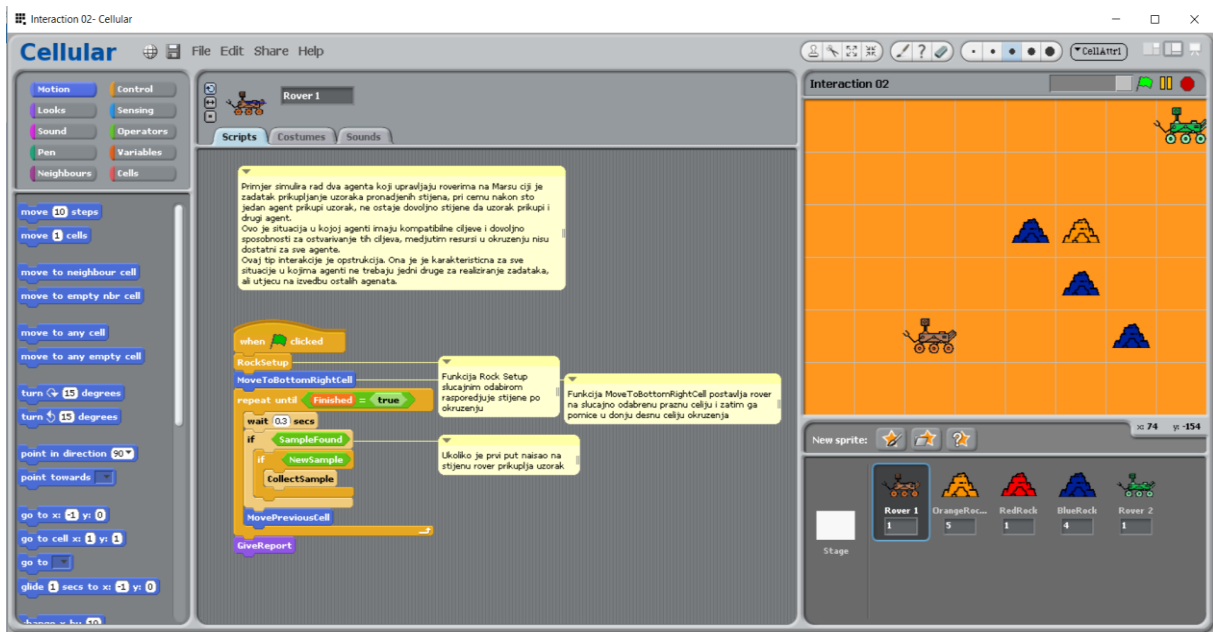


Slika 47. Jednostavna suradnja u višeagentskom sustavu u *NetLogo*

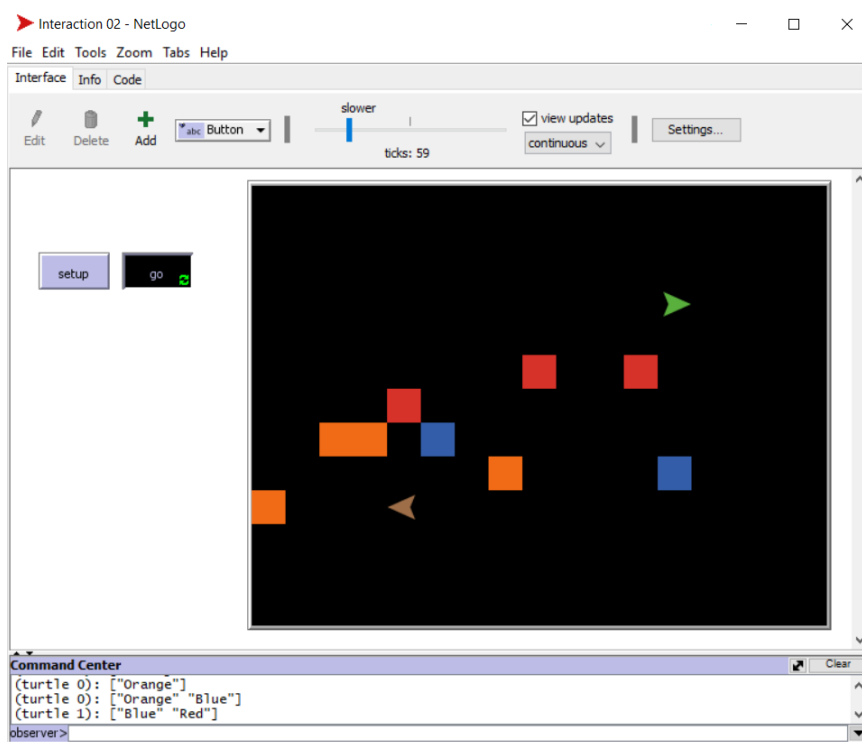
Studenti eksperimentalnih skupina su u simulaciji mogli mijenjati broj primjeraka svake vrste stijena koje se inicijalno javljaju u okruženju.

4.1.4.3. Opstrukcija

Primjer 3.3: Ponovno je i u ovom primjeru opisan rad dva agenta koji upravljaju roverima na Marsu čiji je zadatak prikupljanje uzoraka pronađenih stijena. Pri tome, nakon što jedan agent prikupi uzorak, ne ostaje dovoljno stijene da uzorak prikupi i drugi agent (Prilog 15).



Slika 48. Opstrukcija u višeagentskom sustavu u *Cellularu*



Slika 49. Opstrukcija u višeagentskom sustavu u *NetLogu*

Ovo je situacija u kojoj agenti imaju kompatibilne ciljeve i dovoljno sposobnosti za ostvarivanje tih ciljeva, međutim resursi u okruženju nisu dostatni za sve agente.

Ovaj tip interakcije je opstrukcija. Ona je karakteristična za sve situacije u kojima agenti ne trebaju jedni druge za realiziranje zadataka, ali utječu na izvedbu ostalih agenata.

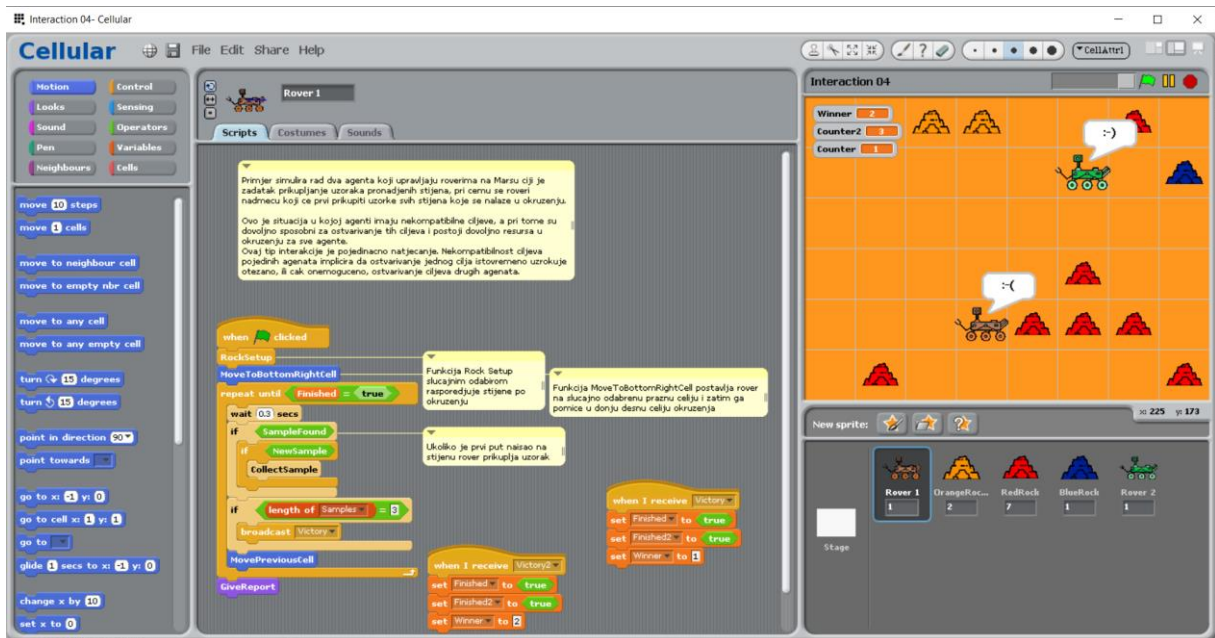
Studenti eksperimentalnih skupina su, i u ovom primjeru, mogli mijenjati broj primjeraka svake vrste stijena koje se inicijalno javljaju u okruženju.

4.1.4.4. Čisto pojedinačno natjecanje

Primjer 3.4: I u ovom primjeru opisan je rad dva agenta koji upravljaju roverima na Marsu, s identičnim zadatkom kao i u prethodnim primjerima. U ovoj simulaciji roveri se nadmeću koji od njih će prije prikupiti uzorke svih stijena koje se nalaze u okruženju (Prilog 16).

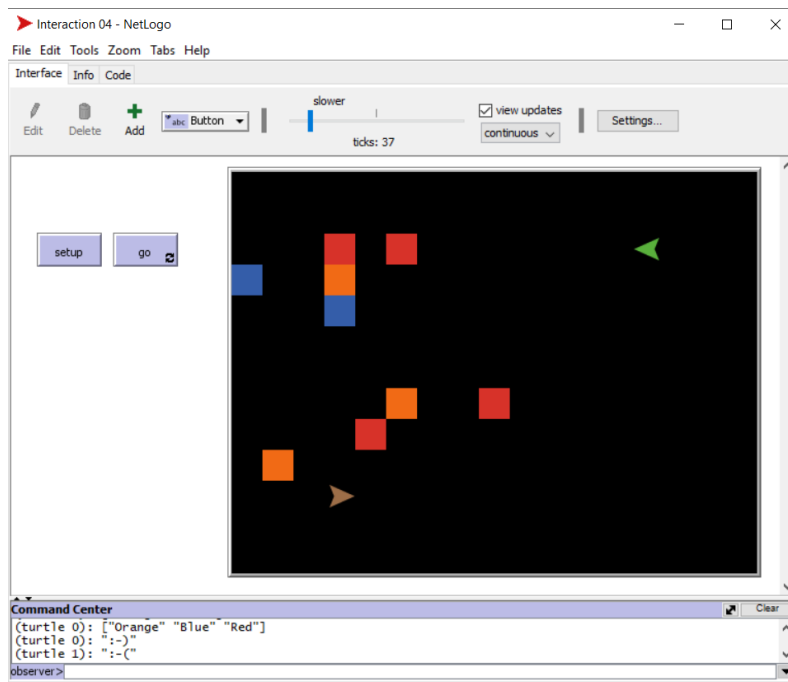
Ovo je situacija u kojoj agenti imaju nekompatibilne ciljeve, a pri tome su dovoljno sposobni za ostvarivanje tih ciljeva i postoji dovoljno resursa u okruženju za sve agente.

Ovaj tip interakcije je čisto pojedinačno natjecanje. Nekompatibilnost ciljeva pojedinih agenata implicira da ostvarivanje jednog cilja istovremeno uzrokuje otežano, ili čak onemogućeno, ostvarivanje ciljeva drugih agenata.



Slika 50. Čisto pojedinačno natjecanje u višeagentskom sustavu u *Cellularu*

Studenti eksperimentalnih skupina su u simulaciji mogli mijenjati broj primjeraka svake vrste stijena koje se inicijalno javljaju u okruženju.



Slika 51. Čisto pojedinačno natjecanje u višeagentskom sustavu u *NetLogo*

4.1.4.5. Ispitivanje znanja nakon NJ3 i analiza rezultata

Neposredno nakon obrađene treće nastavne jedinice sve tri skupine su istovremeno pristupile ispitivanju znanja.

Ispitivanja znanja, u trajanju od 30 minuta, uključivalo je zadatke otvorenog tipa čije je rješavanje zahtijevalo različite razine znanja definirane Bloomovom taksonomijom u kognitivnom području (Bloom, 1956), a studenti su na svakom od njih mogli ostvariti najviše 30 bodova. Studenti su upućeni u način rješavanja zadataka i upoznati s kriterijem bodovanja usklađenim s razinom potrebnog znanja za njihovo rješavanje (Prilog 20).

Za provjeru pouzdanosti ispita korišten je Cronbach α koeficijent. Ispitivanje znanja nakon obrađene prve nastaven jedinice spada u kategoriju visoko pouzdanih (Cronbach $\alpha = 0.852$).

Na osnovi postignutih rezultata izračunati su aritmetička sredina (AS), medijan (MED) i aritmetička sredina rangova (ASR) kao tri mjere srednje vrijednosti te standardna devijacija (SD) i raspon (R) kao dvije mjere disperzije. Analiza rezultata ispitivanja provjere znanja prikazana je u Tablici 11.

Tablica 11. Analiza rezultata ispitivanja znanja nakon NJ3

Statistička mjera	Kontrolna skupina (KS)	Ekperimentalna skupina 1 (ES1)	Ekperimentalna skupina 2 (ES2)
Aritmetička sredina (AS)	10,33	16,33	16,00
Medijan (MED)	9,00	12,00	13,00
Aritmetička sredina rangova (ASR)	8,00	16,50	17,50
Standardna devijacija (SD)	4,77	8,06	7,87
Raspon (R)	17,00	20,00	21,00

Postavljena je nulta hipoteza H_0 da ne postoji značajna razlika u postignućima tri skupine, odnosno da je raspodjela rezultata jednaka u svim skupinama.

Proveden je neparametrijski Kruskal-Wallisov test za tri nezavisne skupine kojim je testirana nulta hipoteza. U tu svrhu svi rezultati su rangirani po veličini. Rezultat analize, provedene uz pomoć statističkog paketa programske podrške SPSS, prikazan je u Tablici 12.

Tablica 12. Kruskal-Wallisov test rezultata ispitivanja znanja nakon NJ3 za tri nezavisne skupine

Nulta hipoteza H_0	Provedeni test	Značajnost	Odluka
Raspodjela rezultata jednaka je u svim skupinama	Kruskal-Wallisov test za nezavisne uzorke	0,019	Odbaciti nultu hipotezu H_0

Prikazana je asimptotska značajnost. Razina značajnosti je 0,05.

Statistička analiza rezultata ispitivanja znanja nakon treće nastavne jedinice odbacila je nultu hipotezu H_0 . Time je pokazano je da postoji statistički značajna razlika među skupinama u poznavanju gradiva treće nastavne jedinice, odnosno da raspodjela postignutih rezultata nije jednaka u sve tri skupine.

Rezultati su dodatno testirani kako bi se preciznije utvrdilo između kojih skupina studenata postoji utvrđena razlika. U tu svrhu su postavljene nulte hipoteze H_0 da nema značajnije razlike u postignućima za svaki par skupina (KS i ES1, KS i ES2, ES1 i ES2). Ove hipoteze su testirane neparametrijskim Mann-Whitneyevim U testom za dvije nezavisne skupine. Rezultati analiza provedenih uz pomoć statističkog softverskog paketa SPSS prikazani su u Tablici 13.

Tablica 13. Mann-Whitneyev U test rezultata ispitivanja znanja nakon NJ2 za parove nezavisnih skupine

Nulta hipoteza H0	Provedeni test	Značajnost	Odluka
Raspodjela rezultata jednaka je u KS i ES1	Mann-Whitneyev U test za dvije nezavisne skupine	0,014	Odbaciti nultu hipotezu H0
Raspodjela rezultata jednaka je u KS i ES2	Mann-Whitneyev U test za dvije nezavisne skupine	0,019	Odbaciti nultu hipotezu H0
Raspodjela rezultata jednaka je u ES1 i ES2	Mann-Whitneyev U test za dvije nezavisne skupine	0,666	Zadržati nultu hipotezu H0

Prikazana je asimptotska značajnost. Razina značajnosti je 0,05.

Statistička analiza rezultata ispitivanja znanja nakon obrađene treće nastavne jedinice pokazala je da postoji značajna razlika u postignućima studenata dvije eksperimentalne skupine u odnosu na kontrolnu skupinu studenata. Također je vidljivo da ne postoji značajnija razlika u postignućima studenata dvije eksperimentalne skupine. Prikazane mjere srednje vrijednosti postignutih rezultata (AS, MED i ASR) ukazuju na činjenicu da su bolje rezultate za gradivo treće nastavne jedinice ostvarili studenti eksperimentalnih skupina, koji su tijekom nastave koristili jedno od odabranih simulacijskih okruženja.

4.1.5. Područja primjene agenata (NJ4)

Na kraju prvog dijela istraživanja, u četvrtoj nastavnoj jedinici (NJ4), studentima su predstavljena različita područja primjene agenata.

Razvoj tehnologija računalnih agenata, odnosno tehnologije uopće, donosi nova i poboljšana rješenja problema u različitim aspektima ljudske djelatnosti: u svakodnevnom životu, na poslu, u komunikaciji i zabavi.

Primjene agentske paradigme nalazimo u:

- malim sustavima;
- srednje složenim sustavima;
- složenim otvorenim sustavima.

Računalni sustavi različite razine složenosti, u kojima agenti mogu imati ključnu ulogu, nalaze se svuda oko nas: kao mehanizam za filtriranje elektroničke pošte ili pretraživanje interneta, za personalizaciju prikaza novosti ili Internet trgovine, kao sigurnosno važan alat u kontroli cestovnog željezničkog ili zračnog prometa.

U kombinaciji s Internetom stvari, primjena agentske paradigma omogućava dodavanje pametnog i inteligentnog ponašanja fizičkim objektima u našem okruženju, čime koncept Interneta stvar dobiva višu dimenziju i postaje Internet agenata.

Uključivanje Interneta stvari i tehnologije računalnih agenata u proizvodna postrojenja dovodi da četvrte industrijske revolucije, tj. Industrije 4.0. Nadogradnja proizvodnih strojeva suvremenim informacijskim i komunikacijskim tehnologijama čini se neizbježnim u stvaranju globalno konkurentne proizvodnje, a samim tim i bitnim dijelom obrazovanja budućih inženjera strojarstva, računarstva i elektrotehnike.

Drugu skupinu agentskih sustava čine simulacijski sustavi, modelirani i programirani za oponašanje realnih sustava s naglašenim bitnim aspektima. Ovi sustavi omogućavaju testiranje rada realnih sustava u kritičnim situacijama bez stvarne opasnosti, mijenjanje ulaznih parametara realnih procesa i promatranje učinka tih promjena, uvid u oku nevidljive procese ili vizualizaciju apstraktnih sustava. Simulacije realnih sustava danas koriste stručnjaci i znanstvenici u cijelom nizu djelatnosti i područja istraživanja kao što su: umjetnost, zabava, prirodne, tehničke i društvene znanosti.

U četvrtoj nastavnoj jedinici studentima su opisani i predstavljeni primjeri primjene agentskih i višeagentskih sustava u različitim područjima istraživanja. Posebna pažnja posvećena je simulacijama zasnovanim na agentima, čijim promatranjem dolazimo da važnih spoznaja o funkcioniranju modeliranih prirodnih, društvenih i tehničkih sustava. Kroz pripremljene primjere primjene agenata studentima je također ukazano na mogućnost praćenja rezultata promjene različitih ulaznih parametara sustava, što je u praksi dovodilo i do različitih rezultata promatranih procesa. Za kontrolnu skupinu primjeri su prikazani korištenjem prezentacije i projektor, dok su obje eksperimentalne skupine samostalno pokrenule, koristile i u manjoj mjeri manipulirale primjerima realiziranim u simulacijskim okruženjima alata *Cellular* i *NetLogo*. Pri tome su primjeri za sve tri skupine opisivali primjene agenata u istovjetnim sustavima. Detaljan opis ovih primjera dat je u nastavku.

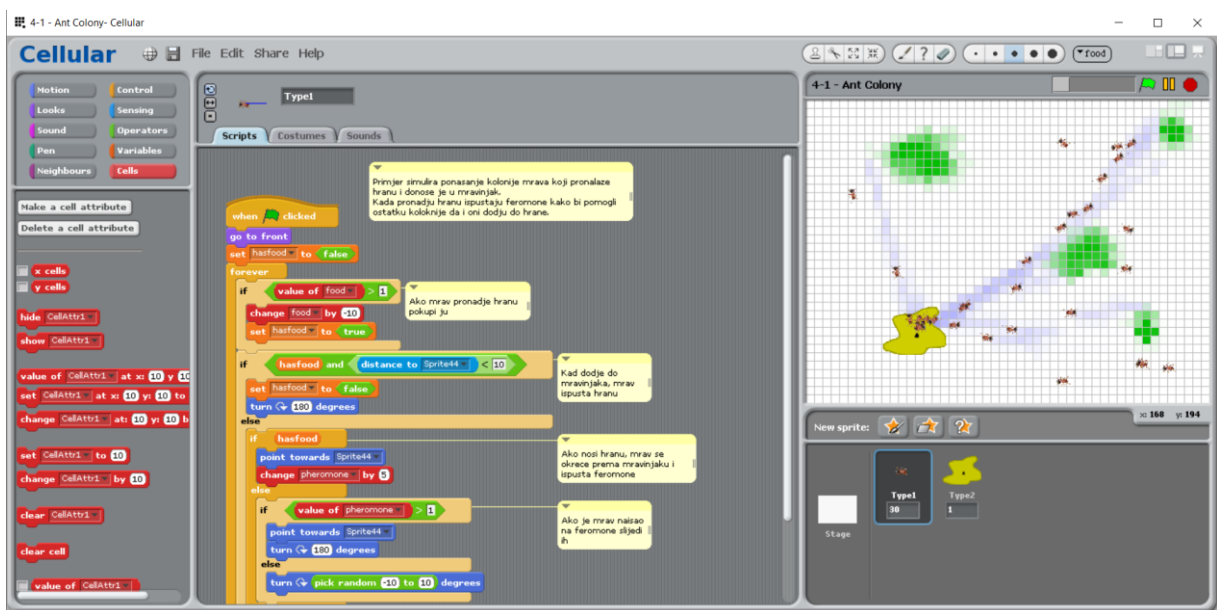
Primjer 4.1: U ovom primjeru simuliran je prirodni, biološki sustav kolonije mrava, odnosno ponašanje kolonije mrava u potrazi za hranom. Iako svaki mrav slijedi niz jednostavnih pravila, kolonija kao cjelina djeluje na sofisticiran način (Wilenski, 1997a).

Kada mrav naiđe komadić hrane, nosi ga natrag u mravinjak, ispuštajući pri kretanju kemijsku tvar, feromone. Kada drugi mravi osjete feromone, prate njihov trag do hrane. S vremenom feromoni isparavaju i trag se gubi. Što više mrava nosi hranu u mravinjak, pojačava se trag feromona i ostatak kolonije lakše pronalazi hranu dok god je na tom mjestu ima.

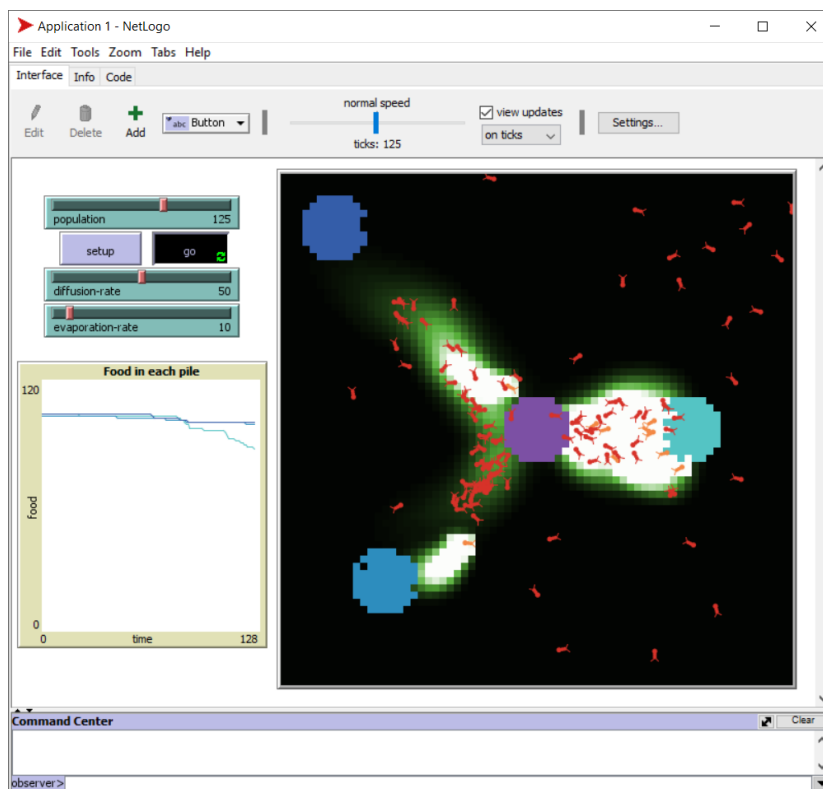
Kolonija mrava općenito iskorištava izvore hrane po redu, počevši od hrane koja je najbliža gnijezdu, a završava s hranom koja je najudaljenija od gnijezda. Mravima je teže stvoriti

stabilnu stazu za udaljeniju hranu, budući da feromonski trag ima više vremena za širenje i isparavanje prije nego se pojača prolaskom novih jedinki kolonije. Nakon što kolonija završi s prikupljanjem najbliže hrane, feromonski trag do te hrane prirodno nestaje, omogućavajući mravima da pomognu u prikupljanju hrane s drugih izvora. Udaljeniji izvori hrane zahtijevaju veći broj mrava kako bi se stvorio stabilan trag.

Studenti eksperimentalnih skupina su u ovoj simulaciji mogli manipulirati brojem mrava u koloniji, brojem izvora hrane u okruženju, brzinom širenja feromona i brzinom njihovog isparavanja.



Slika 52. Simulacija kolonije mrava u prikupljanju hrane u *Cellularu*



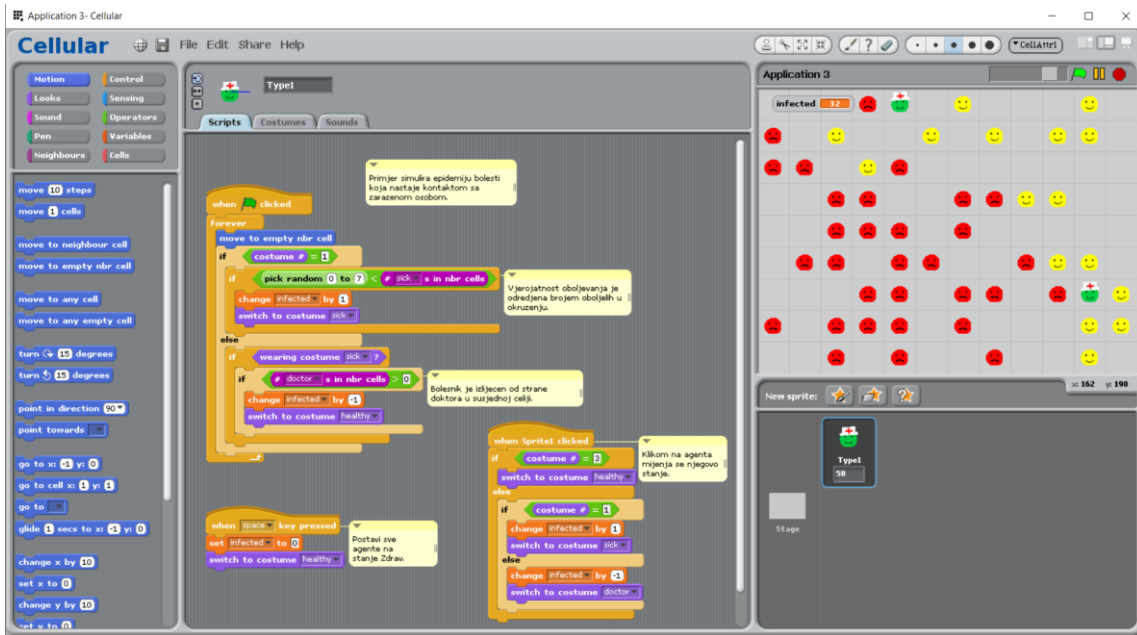
Slika 53. Simulacija kolonije mrava u prikupljanju hrane u *NetLogo*

Primjer 4.2: U ovom primjeru dat je model prirodnog sustava koji simulira prijenos i vijek trajanja virusa u ljudskoj populaciji. Pri tome se vodi računa o nizu čimbenika koji, prema struci, mogu utjecati na preživljavanje virusa koji se izravno prenosi unutar populacije (Wilensky, 1998).

Model je inicijaliziran s 150 osoba, od kojih je 10 zaraženo. Ljudi se nasumice kreću po okruženju i mogu se nalaziti u jednom od tri stanja: zdravi i podložni zarazi, zdravi i imuni te bolesni i zarazni.

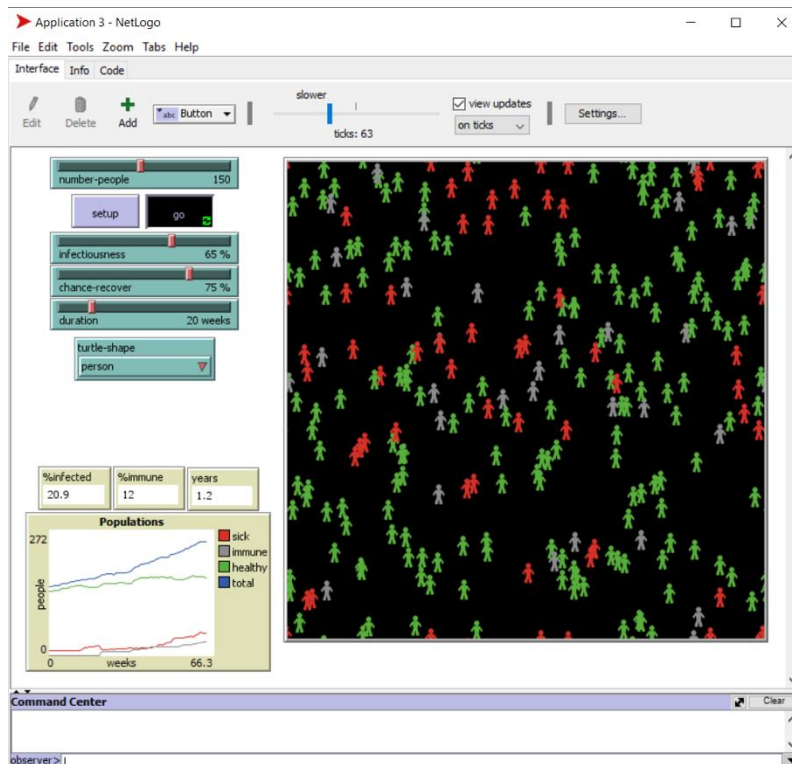
Zaraženi ljudi mogu ozdraviti ili umrijeti od bolesti, u skladu s vjerojatnošću oporavka. Vrijeme nakon kojeg zaražena osoba koja nije ozdravila umire određeno je parametrom simulacijskom modela. Zaražena osoba koja ozdravi postaje imuna na bolest određeno vrijeme. Svi ljudi mogu umrijeti od starosti, bilo da su zaraženi, podložni zarazi ili imuni, dok se novi pripadnici populacije, kao potomstvo zdravih jedinki u situaciji kada je veličina populacije manja od kapaciteta okruženja, rađaju zdravi i podložni zarazi.

Bitan čimbenik ovog modela je gustoća populacije jer ona utječe na to koliko često zaraženi, imuni i pojedinci podložni zarazi dolaze u kontakt. Vjerojatnost zaraze prilikom kontakta s bolesnom osobom je još jedan bitan čimbenik modela.



Slika 54. Simulacija epidemije zarazne bolesti u *Cellularu*

Studenti eksperimentalnih skupina su u ovoj simulaciji mogli mijenjati veličinu inicijalne populacije, vjerojatnost ozdravljanja i zaraze, kao i maksimalno vrijeme trajanja zaraze.



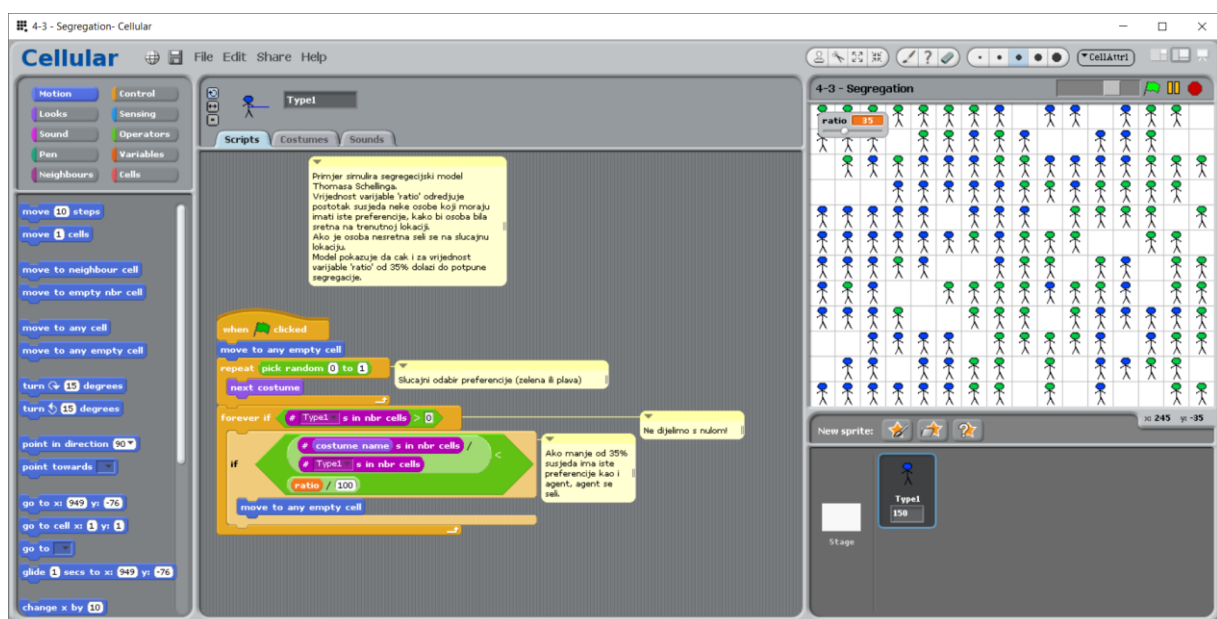
Slika 55. Simulacija epidemije zarazne bolesti u *NetLogo*

Primjer 4.3: U ovom primjeru prikazan je model društvenog sustava koji simulira ponašanje dva tipa agenata u susjedstvu. Boja agenta simbolizira njegove stavove ili preferencije. Svaki agent želi osigurati život u blizini jedinki istih stavova ili preferencija, odnosno u blizini agenata iste boje. Simulacija prikazuje kako se pojedinačne preferencije pronose kroz susjedstvo, što u konačnici dovodi do segregacije skupina agenata istih preferencija (Wilensky, 1997c). Primjer je inspiriran je radom Thomasa Schellinga o društvenim sustavima.

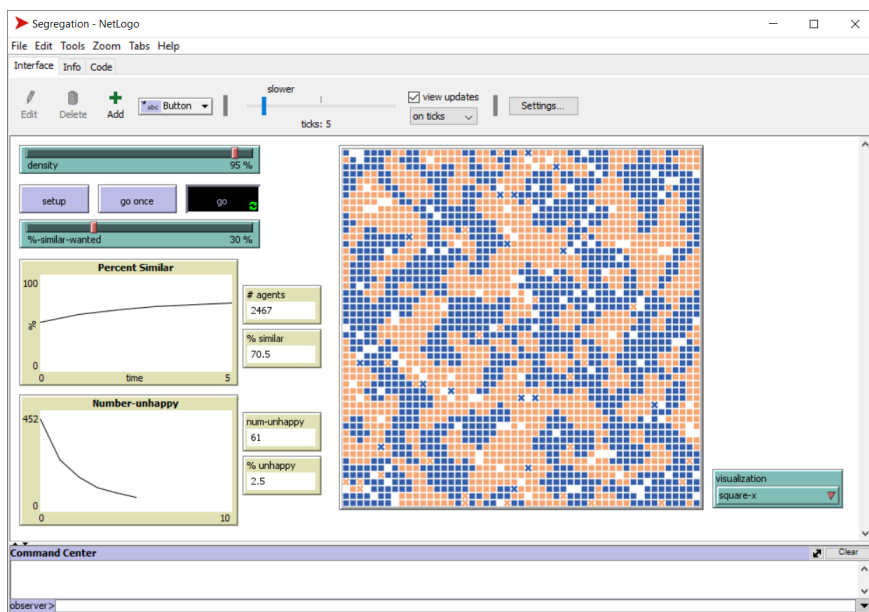
Inicijalno se u okruženju javlja podjednak broj agenata dvije skupine. Ako agenti nemaju dovoljno susjeda iste vrste, presele se na obližnje, prazno područje okruženja.

Nakon preseljenja na novu lokaciju, agenti mogu ugroziti ravnotežu lokalnog stanovništva, što može potaći druge agente da presele. Ako se nekoliko agenata jednog tipa doseli u neko područje, to može navesti agente drugog tipa da se odsele. Tijekom vremena, broj nesretnih agenata se smanjuje, a okruženje postaje sve veše segregirano, s grupacijama agenata iste boje na jedom području. Model pokazuje da kada svaki agent želi barem 30% susjeda istih preferencija, agenti (u prosjeku) završavaju sa 70% susjeda istih preferencija. Tako relativno male individualne preferencije mogu dovesti do značajne ukupne segregacije.

Studenti eksperimentalnih skupina mogu u simulacijama mijenjati gustoću naseljenosti susjedstva, odnosno ukupan broj agenata u okruženju, postotak agenata istog tipa u neposrednom susjedstvu koji trenutnu lokaciju čini zadovoljavajućom. Također mogu pratiti prosječan postotak susjeda iste vrste za svakog agenta i postotak agenata koji imaju manje susjeda iste boje nego što to žele te se žele preseliti.

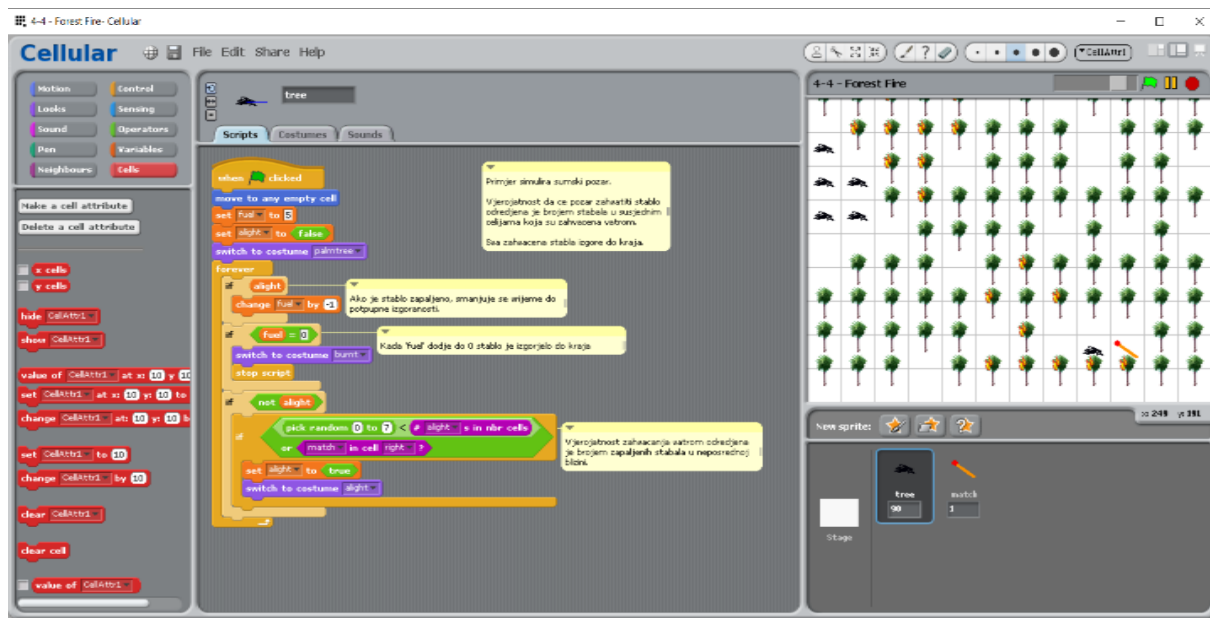


Slika 56. Simulacija segregacijskog modela Thomasa Schellinga u *Cellularu*

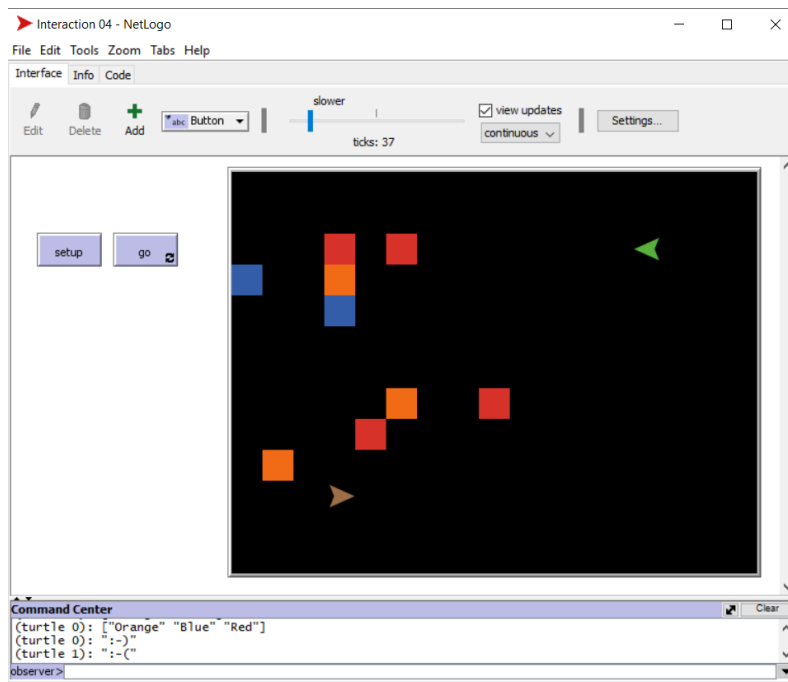


Slika 57. Simulacija segregacijskog modela Thomasa Schellinga u *NetLogo*

Primjer 4.4: U ovom primjeru simuliran je model prirodnog, ekološkog sustava koji prikazuje širenje požara kroz šumu. U ovisnosti o gustoći stabala, model provjerava hoće li vatra progutati cijelu šumu (Wilensky, 1997b). Ovim primjerom prikazane su zajedničke značajke mnogih složenih sustava, tj. postojanje nelinearnog kritičnog praga vjerojatnosti promatranog događaja.



Slika 58. Simulacija širenja šumskog požara u *Cellularu*



Slika 59. Simulacija širenja šumskog požara u *NetLogo*

Požar počinje na lijevom rubu šume i širi se na susjedna stabla. Vatra se prostire u četiri smjera: sjever, istok, jug i zapad. Model pretpostavlja da nema vjetra. Dakle, vatra mora imati stabla duž puta kako bi napredovala. Ona ne može preskočiti ogoljeno područje, tako da svako takvo područje blokira kretanje vatre u tom smjeru.

Stablo koje se zapali i gori se ne pomiče i svako zapaljeno stablo na kraju u potpunosti izgori. Prema tome, vatra se kreće prenošenjem na susjedna stabla. Simulacija pokazuje da za gustoću stabala od 50% praktički nema šanse da vatra dosegne desni rub šume, a da je za gustoću stabala od 70% gotovo sigurno da će požar doći do desnog ruba šume. Kritični prag je, prema ovoj simulaciji, 59% gustoće pri čemu su vjerojatnosti da vatra dosegne desni rub šume ili se zaustavi prije toga gotovo identične.

Studenti eksperimentalnih skupina su u ovoj simulaciji mogli mijenjati gustoću stabala u šumi i način propagiranja požara (četiri osnovne strane svijeta ili svih osam susjednih polja).

4.1.5.1. Ispitivanje znanja nakon NJ2 i analiza rezultata

Neposredno nakon obrađene četvrte nastavne jedinice sve tri skupine su istovremeno pristupile ispitivanju znanja.

Ispitivanja znanja, u trajanju od 30 minuta, uključivalo je zadatke otvorenog tipa čije je rješavanje zahtijevalo različite razine znanja definirane Bloomovom taksonomijom u kognitivnom području (Bloom, 1956), a studenti su na svakom od njih mogli ostvariti najviše

30 bodova. Studenti su upućeni u način rješavanja zadataka i upoznati s kriterijem bodovanja usklađenim s razinom potrebnog znanja za njihovo rješavanje (Prilog 21).

Za provjeru pouzdanosti ispita korišten je Cronbach α koeficijent. Ispitivanje znanja nakon obrađene prve nastaven jedinice spada u kategoriju pouzdanih (Cronbach $\alpha = 0.728$).

Na osnovi postignutih rezultata izračunati su aritmetička sredina (AS), medijan (MED) i aritmetička sredina rangova (ASR) kao tri mjere srednje vrijednosti te standardna devijacija (SD) i raspon (R) kao dvije mjere disperzije. Analiza rezultata ispitivanja provjere znanja prikazana je u Tablici 14.

Tablica 14. Analiza rezultata ispitivanja znanja nakon NJ4

Statistička mjera	Kontrolna skupina (KS)	Eksperimentalna skupina 1 (ES1)	Eksperimentalna skupina 2 (ES2)
Aritmetička sredina (AS)	14,67	9,67	8,61
Medijan (MED)	14,00	9,50	9,00
Aritmetička sredina rangova (ASR)	20,56	11,50	9,94
Standardna devijacija (SD)	4,21	3,61	3,65
Raspon (R)	14,00	13,50	11,50

Postavljena je nulta hipoteza H_0 da ne postoji značajna razlika u postignućima tri skupine, odnosno da je raspodjela rezultata jednaka u svim skupinama.

Proveden je neparametrijski Kruskal-Wallisov test za tri nezavisne skupine kojim je testirana nulta hipoteza. U tu svrhu svi rezultati su rangirani po veličini. Rezultat analize, provedene uz pomoć statističkog paketa programske podrške SPSS, prikazan je u Tablici 15.

Tablica 15. Kruskal-Wallisov test rezultata ispitivanja znanja nakon NJ4 za tri nezavisne skupine

Nulta hipoteza H_0	Provedeni test	Značajnost	Odluka
Raspodjela rezultata jednaka je u svim skupinama	Kruskal-Wallisov test za nezavisne uzorke	0,009	Odbaciti nultu hipotezu H_0

Prikazana je asimptotska značajnost. Razina značajnosti je 0,05.

Statistička analiza rezultata ispitivanja znanja nakon treće nastavne jedinice odbacila je nultu hipotezu H_0 . Time je pokazano je da postoji statistički značajna razlika među skupinama u poznavanju gradiva četvrte nastavne jedinice, odnosno da raspodjela postignutih rezultata nije jednaka u sve tri skupine.

Rezultati su dodatno testirani kako bi se preciznije utvrdilo između kojih skupina studenata postoji utvrđena razlika. U tu svrhu su postavljene nulte hipoteze H_0 da nema značajnije razlike u postignućima za svaki par skupina (KS i ES1, KS i ES2, ES1 i ES2). Ove hipoteze su testirane neparametrijskim Mann-Whitneyevim U testom za dvije nezavisne skupine. Rezultati analiza provedenih uz pomoć statističkog softverskog paketa SPSS prikazani su u Tablici 16.

Tablica 16. Mann-Whitneyev U test rezultata ispitivanja znanja nakon NJ4 za parove nezavisnih skupine

Nulta hipoteza H_0	Provedeni test	Značajnost	Odluka
Raspodjela rezultata jednaka je u KS i ES1	Mann-Whitneyev U test za dvije nezavisne skupine	0,014	Odbaciti nultu hipotezu H_0
Raspodjela rezultata jednaka je u KS i ES2	Mann-Whitneyev U test za dvije nezavisne skupine	0,004	Odbaciti nultu hipotezu H_0
Raspodjela rezultata jednaka je u ES1 i ES2	Mann-Whitneyev U test za dvije nezavisne skupine	0,666	Zadržati nultu hipotezu H_0

Prikazana je asimptotska značajnost. Razina značajnosti je 0,05.

Statistička analiza rezultata ispitivanja znanja nakon obrađene četvrte nastavne jedinice pokazala je da postoji značajna razlika u postignućima studenata dvije eksperimentalne skupine u odnosu na kontrolnu skupinu studenata. Također je vidljivo da ne postoji značajnija razlika u postignućima studenata dvije eksperimentalne skupine. Prikazane mjere srednje vrijednosti postignutih rezultata (AS, MED i ASR) ukazuju na činjenicu da su bolje rezultate za gradivo četvrte nastavne jedinice ostvarili studenti kontrolne skupine, koji su prisustvovali tradicionalnom obliku nastave, tj. *ex cathedra*, uz korištenje prezentacije i projektora.

Prema mišljenju dijela studenata eksperimentalnih skupina, prikupljenih nizom pojedinačnih standardiziranih intervjua, ovo je posljedica njihove usmjerenosti na područja primjene prikazane simulacijama, uz zanemarivanje ostalih potencijalnih područja primjerne.

4.1.6. Analiza ukupnih rezultata provjera znanja nakon nastavnih jedinica

Provedena kvantitativna analiza rezultata studenata kontrolne skupine i dvije eksperimentalne skupine, ostvarenih nakon obrađenih nastavnih jedinica pokazala je da postoje statistički značajne razlike među rezultatima skupina studenata koji su prisustvovali nastavi održanoj na dva načina, u sva četiri slučaja.

Studenti kontrolne skupine, koji su prisustvovali tradicionalnoj nastavi uz korištenje standardnih nastavnih pomagala, tj. pripremljene prezentacije i projektora za prikaz prezentacije, i *ex cathedra* pristup nastavnika koji je nastavno gradivo tumačio verbalno uz

pomoć prezentacija, ostvarili su lošije rezultate od studenata obje eksperimentalne skupine nakon prve tri nastavne jedinice (Definicija, osnovna svojstva i vrste agenata, Definicija osnovna svojstva i vrste okruženja i Interakcija agenata i okruženja i međusobna interakcija agenata u višeagentskim sustavima), a bolje rezultate od studenata obje eksperimentalne skupine nakon četvrte nastavne jedinice (Područja primjene agenata).

Studenti obje eksperimentalne skupine, koji su prisustvovali nastavi uz korištenje simulacijskih okruženja alata *Cellular* i *NetLogo*, i nastavnika koji je, nakon kratkog uvoda u nastavnu jedinicu, ostatak vremena promatrao samostalni rad studenata na pripremljenim simulacijskim primjerima, otklanjao eventualne pojedinačne poteškoće u radu i povremeno ukazivao na bitne značajke simulacije, ostvarili su bolje rezultate od studenata kontrolne skupine nakon prve tri nastavne jedinice (Definicija, osnovna svojstva i vrste agenata, Definicija osnovna svojstva i vrste okruženja i Interakcija agenata i okruženja i međusobna interakcija agenata u višeagentskim sustavima), a lošije rezultate od studenata kontrolne skupine nakon četvrte nastavne jedinice (Područja primjene agenata). Pri tome nisu postojale statistički značajnije razlike u postignućima studenata dvije eksperimentalne skupine nakon bilo koje nastavne jedinice.

Stavovi i dojmovi studenata o održanoj nastavi prikupljeni su nakon provedene nastave korištenjem anketnog upitnika s nizom pitanja procjene u obliku Likertovih ljestvica s 5 ponuđenih kategorija. Njihove primjedbe i preporuke su prikupljene kao odgovori na niz pitanja otvorenog tipa (Prilog 23).

Studenti dviju eksperimentalnih skupina izrazili su zadovoljstvo nastavom u kojoj su korištena simulacijska okruženja, ističući pri tome i mišljenje kako će im ovakav oblik nastave pomoći u daljnjem profesionalnom razvoju te izražavajući želju za većim brojem predmeta u kojem bi se koristili slični nastavni alati. Procjene studenata kontrolne skupine, za ista pitanja, ukazale su na manji stupanj zadovoljstva održanom nastavom, uz stav kako će im stečena znanja iz područja agentske paradigme koristiti u budućnosti.

Na osnovi primjedbi i preporuka studenata kontrolne skupine (KS) ističe se želja nekolicine studenata da se praktični primjeri korišteni u nastavi eksperimentalnih skupina učine dostupnima, kako bi i oni mogli obrađeno gradivo sagledati iz druge perspektive.

Primjedbe i preporuke studenata eksperimentalnih skupina ukazuju na veće zadovoljstvo simulacijskim okruženjem *NetLogo*, koji su svi studenti druge eksperimentalne skupine (ES2)

ocijenili izrazito pozitivnim, uz naglasak na lakšu prilagodbu programskom jeziku koji *NetLogo* koristi zbog ranijeg susretanja s programskim jezikom *Logo*. Dio studenata prve eksperimentalne skupine (ES1) je kao nedostatak simulacijskog okruženja *Cellular* naveo njegov "djetinjast" i "neozbiljan" izgled, koji, prema njihovom mišljenju, nije prilagođen upotrebi na višim godinama studija. Također su istakli jednostavnu prilagodbu na vizualno programiranje slaganjem programskih blokova u *Cellularu*, s kojim su se ranije tijekom studija susretali u alatima *Scratch* i *Enchanting*.

Ukupan broj studenata koji je sudjelovao u prvom dijelu istraživanja bio je relativno malen (27 studenata), pa su ostvareni rezultati koji su ukazivali na uspješnost eksperimentalnog pristupa nastavi uz korištenje simulacijskih okruženja zasnovanih na agentima i pripremljenih prigodnih primjera modela sustava, dodatno potvrđeni kvalitativnom analizom .

4.1.7. Kvalitativna analiza nakon prvog dijela istraživanja

Razumijevanje koncepata agentske paradigme provjereno je nizom pojedinačnih standardiziranih intervjua s pitanjima otvorenog tipa (Prilog 24). U intervjuiima su sudjelovali oni studenti koji su u svakoj od skupina postigli najbolji i najlošiji rezultat, kao i studenti koji su postigli najveći napredak u odnosu na inicijalnu provjeru znanja.

Ovaj fenomenološki pristup predstavlja inačicu kvalitativnih, naturalističkih pristupa u kojem se proučava izravno iskustvo, shvaćeno doslovno. Ponašanje je određeno fenomenima iskustva, a ne vanjskom, objektivnom i fizikalno opisanom stvarnošću (English i English, 1958).

Tijekom ovih intervjua ustanovljen je visok stupanj razumijevanja koncepata inteligentnog agenta, njegovog okruženja i višeagentskih sustava kod studenata čiji je rezultat bio najbolji u skupini, kao i kod studenata čiji se rezultat tijekom istraživanja u nastavi najviše popravio. Kod studenata čiji je rezultat bio slab, ustanovljeno je nedostatan ili pogrešno razumijevanje ovih pojmova i odnosa između njih. U prvom redu studenti nisu mogli dobro razgraničiti svojstva i funkcije agenta i okruženja, što je za posljedicu imalo i pogrešne odgovore pri ispitivanjima znanja nakon nastavnih jedinica.

Za koncept inteligentnog agenta neki od odgovora intervjuiranih studenata koji su pokazali višu razinu razumijevanja ovog koncepta bili su:

- *Agenti su entiteti koji opažaju svoju okolinu sensorima i na nju djeluju putem aktuatora. Inteligentni agenti sudjeluju u procesu prijena i koordiniranja informacija. Koriste se*

u sustavima vezanim za zrakoplove, internet pretraživačima, filterima za poruke, internet trgovini.

- *Inteligentni agent je programski i/ili sklopovski entitet koji opaža okolinu sensorima i na nju djeluje putem aktuatora. Glavno njegovo svojstvo je autonomija. Primjenjuje se u malim sustavima, srednjim sustavima i složenim proširenim sustavima.*

Primjeri studentskih odgovora koji pokazuju nižu razinu razumijevanja istog koncepta su:

- *Inteligentni agenti su programi koji omogućavaju komunikaciju među okolinom.*
- *Inteligentni agenti su entiteti koji posjeduju senzore za obavljanje nekih radnji, tj. pomoću senzora djeluju na okolinu.*

Za koncept okruženja inteligentnog agenta neki od odgovora intervjuiranih studenata koji su pokazali višu razinu razumijevanja ovog koncepta bili su:

- *Okruženje inteligentnog agenta je okruženje u kojem agenti obavljaju određene zadatke i s kojim su u interakciji.*
- *Okruženje u kojem agent djeluje može najbolje opisati ponašanje agenata. Agenti će djelovati u skladu s okruženjem u kojem se nalaze i karakteristikama tog okruženja, te na osnovu podražaja koje iz tog okruženja primaju.*

Primjeri studentskih odgovora koji pokazuju nižu razinu razumijevanja istog koncepta su:

- *Okruženje inteligentnog sustava promatra i mjeri učinak agenta.*
- *Okruženje inteligentnog sustava sastoji se od senzora i aktuatora*

Za koncept međusobne interakcije inteligentnih agenata u višeagentskim sustavima neki od odgovora intervjuiranih studenata koji su pokazali višu razinu razumijevanja ovog koncepta bili su:

- *Interakcija može biti pozitivna i negativna. Kod pozitivne interakcije djelovanje jednog agenta ne ugrožava djelovanje drugih agenata (neovisnost i suradnja), dok je to slučaj kod negativne interakcije (opstrukcija, natjecanje i sukob).*
- *Pozitivna interakcija je slučaj u kojem djelovanje jednog agenta nema negativne posljedice na djelovanje drugog agenta. Oni mogu djelovati neovisno ili surađivati u ostvarivanju svojih ciljeva. Tipovi negativne interakcije su opstrukcija, pojedinačno ili kolektivno natjecanje i sukob oko resursa.*

Primjeri studentskih odgovora koji pokazuju nižu razinu razumijevanja istog koncepta su:

- *Interakcija među agentima je pozitivna ako imaju zajednički cilj, a negativna ako nemaju zajednički cilj.*
- *Do interakcije među agentima dolazi kada se izmjenjuju informacije o okruženju.*

Studenti eksperimentalnih skupina su također iskazali mišljenje da su prikazani primjeri simulacija u nastavnoj jedinici koja je obrađivala područja primjene agenata, njihovu pozornost usmjerili samo na te primjene, zanemarujući ostala potencijalna područja primjerne. Zbog toga su i postignuti rezultati, prema njihovom mišljenju, za ovu nastavnu jedinicu lošiji od rezultata kontrolne skupine.

Rezultati prvog dijela istraživanja ukazuju na prednost eksperimentalnog pristupa nastavi iz područja agentske paradigme u prve tri nastavne jedinice, odnosno klasičnog ex cathedra pristupa u četvrtoj nastavnoj jedinici. Zbog malog broja studenata koji su sudjelovali, planom istraživanja bilo je predviđeno da se ovi rezultati potvrde kvantitativnom analizom podataka dobivenih drugim dijelom istraživanja koje će se provesti u okviru jedne, odabrane nastavne jedinice. Kako su u drugom dijelu istraživanja sudjelovali studenti bez formalnih predznanja iz područja agentske paradigme, jer se tijekom studija nisu susretali s njim, te u dogovoru s predmetnim nastavnicima na predmetima u okviru kojih je drugi dio istraživanja proveden, odabrana je nastavna jedinica Definicija, osnovna svojstva i vrste okruženja inteligentnih agenata.

Izvornim planom istraživanja bilo je predviđeno da se za potrebe drugog dijela istraživanja koriste simulacijska okruženja koja će biti dostupna i na mobilnim aparatima i tablet računalima. U tom trenutku nije postojala mrežna inačica *NetLogo* alata. Kako je do trenutka izbora alata ova inačica postala javno dostupna i omogućila izvođenje simulacija kako na stolnim računalima tako i na tablet računalima i mobilnim uređajima nije bilo potrebe za razvojem novog okruženja. *NetLogo* je, prema stavovima i dojmovima studenata prikupljenim anketnim upitnikom te iznesenim u pojedinačnim intervjuima zadovoljio svoju ulogu, odlučeno je da se za drugi dio istraživanja, kao alat za modeliranje i simuliranje zasnovan na agentima koristi ovaj alat.

4.2. Drugi dio istraživanja

Drugi dio istraživanja proveden je u zimskom semestru akademske 2018./2019. godine, u okviru predmeta Uvod u računarstvo u prvom semestru preddiplomskog studija informatike na Fakultetu prirodoslovno-matematičkih i odgojnih znanosti Sveučilišta u Mostaru (FPMOZ-SUM) i u okviru predmeta Umjetna inteligencija u prvom semestru diplomskih studija strojarstva i računarstva na Fakultetu strojarstva, računarstva i elektrotehnike Sveučilišta u Mostaru (FSRE-SUM).

Na osnovu iskustava iz prvog dijela istraživanja u drugom dijelu odabrana je nastavna jedinica kojom su obrađeni definicija, osnovna svojstva i vrste okruženja u kojima djeluju inteligentni agenti.

4.2.1. Istraživanje u okviru predmeta Uvod u računarstvo na preddiplomskom studiju informatike

Aktualni nastavni plan i program predmeta Uvod u računarstvo uključuje nastavnu cjelinu Upoznavanje s pojmovima umjetne inteligencije. Ovo je omogućilo da se za potrebe istraživanja provede nastava prema redovnom programu, te osmisle i izrade instrumenti provjere znanja.

Tijekom drugog dijela istraživanja, na predmetu Uvod u računarstvo, studenti su bili podijeljeni u dvije skupine, jednu kontrolnu (KS) i jednu eksperimentalnu skupinu (ES). Nastavi je prisustvovalo ukupno 80 studenata, koji su činili neslučajni prirodni uzorak. Studenti su bili ravnomjerno raspoređeni u kontrolnu i eksperimentalnu skupinu slučajnim odabirom.

Nastava iz odabrane nastavne jedinice je za kontrolnu skupinu izvedena na tradicionalan način, tj. *ex cathedra*, dok je za eksperimentalnu skupinu izvedena uz upotrebu simulacijskog okruženja *NetLogo*, dostupnog na računalima, tabletima i pametnim telefonima, te pripremljenih karakterističnih primjera, ranije opisanih u okviru nastavne jedinice NJ2 u prvom dijelu istraživanja. Primjere su studenti mogli sami aktivno koristiti mijenjanjem različitih parametara simulacija.

Nastavu su izvodila dva nastavnika s višegodišnjim iskustvom u provedbi nastave iz područja računarstva i informatike. Jedan nastavnik je izvodio nastavu s kontrolnom skupinom, dok je drugi nastavnik izvodio nastavu s eksperimentalnom kontrolnom skupinom studenata na prvoj godini preddiplomskog studija informatike. Za odabranu nastavnu jedinicu utrošeno je 90 minuta, tj. dva školska sata.

Neposredno nakon obrađene nastavne jedinice provedeno je ispitivanje znanja za utvrđivanje razine usvojenosti obrađenih nastavnih sadržaja.

Ispitivanje znanja, u trajanju od 30 minuta, uključivalo je zadatke otvorenog tipa čije je rješavanje zahtijevalo različite razine znanja definirane Bloomovom taksonomijom u kognitivnom području (Bloom, 1956), a studenti su na svakom od njih mogli ostvariti najviše 70 bodova. Studenti su upućeni u način rješavanja zadataka i upoznati s kriterijem bodovanja usklađenim s razinom potrebnog znanja za njihovo rješavanje (Prilog 22).

Za provjeru pouzdanosti ispita korišten je Cronbach α koeficijent. Ispitivanje znanja nakon obrađene prve nastaven jedinice spada u kategoriju visoko pouzdanih (Cronbach $\alpha = 0.886$).

Na osnovi postignutih rezultata izračunata je aritmetička sredina (AS) kao jedna od mjera srednje vrijednosti te standardna devijacija (SD) i koeficijent varijabilnosti (KV) kao dvije mjere disperzije. Analiza rezultata ispitivanja znanja nakon obrađene nastavne jedinice prikazana je u Tablici 17.

Tablica 17. Analiza rezultata ispitivanja znanja nakon NJ4

Statistička mjera	Kontrolna skupina (KS)	Eksperimentalna skupina (ES)
Aritmetička sredina (AS)	29,88	37,75
Standardna devijacija (SD)	11,97	14,83
Koeficijent varijabilnosti (KV)	40,06	39,28

Postavljena je nulta hipoteza H_0 da ne postoji značajna razlika u postignućima dviju skupina.

Shapiro-Wilkov test i vizualni pregled histograma i Q-Q grafikona pokazuje da su distribucije rezultata testiranja (kontrolna i eksperimentalna skupina na FPMOZ-SUM) normalne, jer je značajnost Shapiro-Wilkovog testa veća od 0.05. Rezultati Shapiro-Wilkovog testa normalnosti distribucije postignutih rezultata za obje skupine prikazani su u Tablici 18.

Tablica 18. Rezultati Shapiro-Wilkovog testa normalnosti distribucije rezultata za kontrolnu i eksperimentalnu skupinu studenata na prvoj godini preddiplomskog studija informatike

Skupina	Shapiro-Wilkov test normalnosti
Kontrolna skupina (KS)	0,248
Eksperimentalna skupina (ES)	0,579

Kako je utvrđeno da je distribucija ostvarenih rezultata normalna, postojanje statistički značajne razlike između rezultata kontrolne i eksperimentalne skupine studenata na prvoj godini preddiplomskog studija informatike provjereno je parametrijskim neparnim t-testom za dvije nezavisne skupine. Rezultat analize, provedene uz pomoć statističkog paketa programske podrške SPSS, prikazan je u Tablici 19.

Tablica 19. Neparni t-test rezultata ispitivanja znanja nakon nastavne jedinice za dvije nezavisne skupine studenata preddiplomskog studija informatike

Standardizirana vrijednost	KS i ES na FPMOZ-SUM
t	-2,613
P	0.011

Broj stupnjeva slobode za test rezultata ostvarenih na FPMOZ-SUM je 78 pa razina rizika, uz razinu značajnosti 0,05, iznosi $p = 1,99$. Obzirom da su apsolutna vrijednost t-testa veća od razine rizika odbacujemo nultu hipotezu H_0 i zaključujemo da postoji statistički značajna razlika u ostvarenim rezultatima kontrolne i eksperimentalne skupine studenata na predmetu Uvod u računarstvo preddiplomskog studija informatike na Fakultetu prirodoslovno-matematičkih i odgojnih znanosti Sveučilišta u Mostaru. Prikazana mjera srednje vrijednosti postignutih rezultata (AS) ukazuju na činjenicu da su bolje rezultate za gradivo obrađene nastavne jedinice ostvarili studenti eksperimentalne skupine, koji su tijekom nastave koristili simulacijsko okruženje *NetLogo*.

4.2.2. Istraživanje u okviru predmeta Umjetna inteligencija na diplomskim studijima strojarstva i računarstva

Istraživanje je ponovljeno i na predmetu Umjetna inteligencija na diplomskim studijima strojarstva i računarstva Fakulteta strojarstva, računarstva i elektrotehnike Sveučilišta u

Mostaru. Aktualni nastavni plan i program predmeta uključuje nastavnu cjelinu Primjeri primjene sustava umjetne inteligencije, što je omogućilo da se za potrebe istraživanja nastava provede prema redovnom programu, te osmisle i izrade instrumenti provjere znanja.

Tijekom drugog dijela istraživanja, na predmetu Umjetna inteligencija, studenti su bili podijeljeni u dvije skupine, jednu kontrolnu (KS) i jednu eksperimentalnu skupinu (ES). Nastavi je prisustvovalo ukupno 72 studenata, koji su činili neslučajni prirodni uzorak. Studenti su bili ravnomjerno raspoređeni u kontrolnu i eksperimentalnu skupinu slučajnim odabirom.

Nastava iz odabrane nastavne jedinice je za kontrolnu skupinu izvedena na tradicionalan način, tj. *ex cathedra*, dok je za eksperimentalnu skupinu izvedena uz upotrebu simulacijskog okruženja *NetLogo*, dostupnog na računalima, tabletima i pametnim telefonima, te pripremljenih karakterističnih primjera koje su studenti mogli sami aktivno koristiti mijenjanjem različitih parametara simulacija.

Nastavu su izvodila dva nastavnika s višegodišnjim iskustvom u provedbi nastave iz područja računarstva i informatike. Jedan nastavnik je izvodio nastavu s kontrolnom skupinom, dok je drugi nastavnik izvodio nastavu s eksperimentalnom kontrolnom skupinom studenata na prvoj godini diplomskih studija strojarstva i računarstva. Pri tome su nastavnici zamijenili skupine u odnosu na prvu iteraciju drugog dijela istraživanja, sa studentima preddiplomskog studija informatike, tj. nastavnik koji je izvodio nastavu s kontrolnom skupinom na predmetu Uvod u računarstvo je u ovoj iteraciji izvodio nastavu s eksperimentalnom skupinom, dok je nastavnik koji je izvodio nastavu s eksperimentalnom skupinom na predmetu Uvod u računarstvo u ovoj iteraciji izvodio nastavu s kontrolnom skupinom. Za odabranu nastavnu jedinicu utrošeno je 90 minuta, tj. dva školska sata.

Neposredno nakon obrađene nastavne jedinice provedeno je ispitivanje znanja za utvrđivanje razine usvojenosti obrađenih nastavnih sadržaja.

Ispitivanja znanja, u trajanju od 30 minuta, uključivalo je zadatke otvorenog tipa čije je rješavanje zahtijevalo različite razine znanja definirane Bloomovom taksonomijom u kognitivnom području (Bloom, 1956), a studenti su na svakom od njih mogli ostvariti najviše 70 bodova. Studenti su upućeni u način rješavanja zadataka i upoznati s kriterijem bodovanja usklađenim s razinom potrebnog znanja za njihovo rješavanje (Prilog 22).

Za provjeru pouzdanosti ispita korišten je Cronbach α koeficijent. Ispitivanje znanja nakon obrađene prve nastaven jedinice spada u kategoriju visoko pouzdanih (Cronbach $\alpha = 0.836$).

Na osnovi postignutih rezultata izračunata je aritmetička sredina (AS) kao jedna od mjera srednje vrijednosti te standardna devijacija (SD) i koeficijent varijabilnosti (KV) kao dvije mjere disperzije. Analiza rezultata ispitivanja znanja nakon obrađene nastavne jedinice prikazana je u Tablici 20.

Tablica 20. Analiza rezultata ispitivanja znanja nakon NJ4

Statistička mjera	Kontrolna skupina (KS)	Eksperimentalna skupina (ES)
Aritmetička sredina (AS)	34,67	42,08
Standardna devijacija (SD)	10,72	9,34
Koeficijent varijabilnosti (KV)	30,92	22,20

Postavljena je nulta hipoteza H_0 da ne postoji značajna razlika u postignućima dviju skupina.

Shapiro-Wilkov test i vizualni pregled histograma i Q-Q grafikona pokazuje da su distribucije rezultata testiranja (kontrolna i eksperimentalna skupina na FSRE-SUM) normalne, jer je značajnost Shapiro-Wilkovog testa veća od 0.05. Rezultati Shapiro-Wilkovog testa normalnosti distribucije postignutih rezultata za obje skupine prikazani su u Tablici 21.

Tablica 21. Rezultati Shapiro-Wilkovog testa normalnosti distribucije rezultata za kontrolnu i eksperimentalnu skupinu studenata na prvoj godini diplomskih studija strojarstva i računarstva

Skupina	Shapiro-Wilkov test normalnosti
Kontrolna skupina (KS)	0,832
Eksperimentalna skupina (ES)	0,612

Kako je utvrđeno da je distribucija ostvarenih rezultata normalna, postojanje statistički značajne razlike između rezultata kontrolne i eksperimentalne skupine studenata na prvoj godini diplomskog studija strojarstva i računarstva provjereno je parametrijskim neparnim t-testom za dvije nezavisne skupine. Rezultat analize, provedene uz pomoć statističkog paketa programske podrške SPSS, prikazan je u Tablici 22.

Tablica 22. Neparni t-test rezultata ispitivanja znanja nakon nastavne jedinice za dvije nezavisne skupine studenata diplomskih studija strojarstva i računarstva

Standardizirana vrijednost	KS i ES na FSRE-SUM
t	-3,130
P	0,003

Broj stupnjeva slobode za test rezultata ostvarenih na FSRE-SUM je 70 pa razina rizika, uz razinu značajnosti 0,05, iznosi $p = 2,00$. Obzirom da su apsolutna vrijednost t-testa veća od razine rizika odbacujemo nultu hipotezu H_0 i zaključujemo da postoji statistički značajna razlika u ostvarenim rezultatima kontrolne i eksperimentalne skupine studenata na predmetu Umjetna inteligencija diplomskih studija strojarstva i računarstva na Fakultetu strojarstva, računarstva i elektrotehnike Sveučilišta u Mostaru. Prikazana mjera srednje vrijednosti postignutih rezultata (AS) ukazuju na činjenicu da su bolje rezultate za gradivo obrađene nastavne jedinice ostvarili studenti eksperimentalne skupine, koji su tijekom nastave koristili simulacijsko okruženje *NetLogo*.

4.2.3. Analiza ujednačenosti skupina

Izvornim planom istraživanja bilo je predviđeno da se u ovom dijelu odabrana nastavna jedinica obradi u vremenu od dva školska sata, tj. 90 minuta, nakon čega je uslijedilo ispitivanje znanja u trajanju od 30 minuta.

Zbog vremenskog ograničenja određenog nastavnim planom i programom predmeta u okviru kojih je ovaj dio istraživanja proveden, kao i zbog činjenice da niti jedna skupina studenata nije imala prethodno formalno obrazovanje iz ovog područja, tj. studenti nisu slušali i polagali predmete iz područja agentske paradigme niti umjetne inteligencije općenito, kao mogući pokazatelj ujednačenosti slučajno odabranih skupina poslužile su njihove ocjene ostvarene iz predmeta u okviru kojih je istraživanje provedeno.

Na osnovi ostvarenih ocjena svih skupina (kontrolna i eksperimentalna skupina na FPMOZ-SUM, kontrolna i eksperimentalna skupina na FSRE-SUM) izračunata je aritmetička sredina (AS) kao jedna od mjera srednje vrijednosti te standardna devijacija (SD) i koeficijent varijabilnosti (KV) kao dvije mjere disperzije. Analiza ostvarenih ocjena iz predmeta u okviru kojih je drugi dio istraživanja proveden prikazana je u Tablici 23.

Tablica 23. Analiza ostvarenih ocjena iz predmeta u okviru kojih je istraživanje provedeno

Statistička mjera	KS na FPMOZ-SUM	ES na FPMOZ-SUM	KS na FSRE-SUM	ES na FSRE-SUM
Aritmetička sredina (AS)	2,63	2,78	3,11	3,14
Standardna devijacija (SD)	0,95	1,23	1,14	1,07
Koeficijent varijabilnosti (KV)	36,12	44,24	36,66	34,08

Postavljena je nulta hipoteza H_0 da ne postoji značajna razlika u ostvarenim ocjenama kontrolne i eksperimentalne skupina na svakom od predmeta.

Shapiro-Wilkov test i vizualni pregled histograma i Q-Q grafikona pokazuje da su distribucije ostvarenih ocjena (kontrolna i eksperimentalna skupina na FPMOZ-SUM, kontrolna i eksperimentalna skupina na FSRE-SUM) normalne, jer je značajnost Shapiro-Wilkovog testa veća od 0.05. Rezultati Shapiro-Wilkovog testa normalnosti distribucije postignutih rezultata za obje skupine prikazani su u Tablici 24.

Tablica 24. Rezultati Shapiro-Wilkovog testa normalnosti distribucije rezultata za kontrolne i eksperimentalne skupine studenata uključenih u drugi dio istraživanja

Skupina	Shapiro-Wilkov test normalnosti
Kontrolna skupina FPMOZ-SUM	0,842
Eksperimentalna skupina FPMOZ-SUM	0,861
Kontrolna skupina FSRE-SUM	0,897
Eksperimentalna skupina FSRE-SUM	0,901

Kako je utvrđeno da je distribucija ostvarenih rezultata normalna, postojanje statistički značajne razlike između ostvarenih ocjena kontrolne i eksperimentalne skupine studenata na prvoj godini preddiplomskog studija informatike, kao i između ostvarenih ocjena kontrolne i eksperimentalne skupine studenata na prvoj godini diplomskih studija strojarstva i računarstva, provjereno je parametrijskim neparnim t-testom za dvije nezavisne skupine. Rezultat analize, provedene uz pomoć statističkog paketa programske podrške SPSS, prikazan je u Tablici 25.

Tablica 25. Neparni t-test ostvarenih ocjena za dvije nezavisne skupine studenata preddiplomskog studija informatike i dvije nezavisne skupine studenata diplomskih studija strojarstva i računarstva

Standardizirana vrijednost	KS i ES na FPMOZ-SUM	KS i ES na FSRE-SUM
t	-0,610	-0,106
P	0,040	0,660

Broj stupnjeva slobode za test rezultata ostvarenih na FPMOZ-SUM je 78 pa razina rizika, uz razinu značajnosti 0,05, iznosi $p = 1,99$. Obzirom da su apsolutna vrijednost t-testa manja od razine rizika zadržavamo nultu hipotezu H_0 i zaključujemo da ne postoji statistički značajna razlika u ostvarenim rezultatima kontrolne i eksperimentalne skupine studenata na predmetu Uvod u računarstvo preddiplomskog studija informatike na Fakultetu prirodoslovno-matematičkih i odgojnih znanosti Sveučilišta u Mostaru.

Broj stupnjeva slobode za test rezultata ostvarenih na FSRE-SUM je 70 pa razina rizika, uz razinu značajnosti 0,05, iznosi $p = 2,00$. Obzirom da su apsolutna vrijednost t-testa manja od razine rizika zadržavamo nultu hipotezu H_0 i zaključujemo da ne postoji statistički značajna razlika u ostvarenim rezultatima kontrolne i eksperimentalne skupine studenata na predmetu Umjetna inteligencija diplomskih studija strojarstva i računarstva na Fakultetu strojarstva, računarstva i elektrotehnike Sveučilišta u Mostaru.

4.2.4. Dodatna analiza postignutih rezultata

Na osnovi izračunatih mjera srednjih vrijednosti (AS) kontrolnih, odnosno eksperimentalnih skupina studenata na oba predmeta, uočeno je da su srednje vrijednosti rezultata studenata preddiplomskog studija niže od srednjih vrijednosti rezultata studenata diplomskih studija. Zbog toga su dodatno međusobno uspoređeni ostvareni rezultati studenata kontrolnih skupina, kao i rezultati studenata eksperimentalnih skupina na dva predmeta.

Postavljena je nulta hipoteza H_0 da ne postoji značajna razlika u postignućima dviju skupina koje su prisustvovala istom obliku nastave.

Ranije je pokazano da su rezultati svih skupina imali normalnu razdiobu. Zbog toga je postojanje statistički značajne razlike između rezultata kontrolnih skupina, odnosno eksperimentalnih skupina na dva predmeta provjereno je parametrijskim neparnim t-testom za

dvije nezavisne skupine. Rezultat analize, provedene uz pomoć statističkog paketa programske podrške SPSS, prikazan je u Tablici 26.

Tablica 26. Neparni t-test rezultata ispitivanja znanja za dvije nezavisne kontrolne skupine studenata i dvije nezavisne eksperimentalne skupine studenata

Standardizirana vrijednost	KS na FPMOZ-SUM i KS na FSRE-SUM	ES na FPMOZ-SUM i ES na FSRE-SUM
T	-1,830	-1,504
P	0,071	0,137

Broj stupnjeva slobode za oba testa je 74 pa razina rizika, uz razinu značajnosti 0,05, iznosi $p = 2,00$. Obzirom da su apsolutne vrijednosti t-testova manje od razine rizika zadržavamo nultu hipotezu H_0 i zaključujemo da ne postoji statistički značajna razlika u ostvarenim rezultatima kontrolnih, odnosno eksperimentalnih skupina na dva fakulteta.

4.3. Treći dio istraživanja

Analizom provedenog drugog dijela istraživanja uočena su dva nedostatka koja bi mogle negativno utjecati na kvalitetu prikupljenih podataka, a time i na rezultate dobivene analizom tih podataka. Prvi nedostatak se odnosi na kratko vrijeme u kojem je nastava izvođena, tj. dva školska sata. Drugi nedostatak je izostanak inicijalne provjere znanja, te ponovne provjere znanja s dovoljnim vremenskim odmakom. Kako bi se ovi nedostaci eliminirali, odlučeno je da se provede i treći dio istraživanja.

Treći dio istraživanja proveden je u zimskom semestru akademske 2019./2020. godine, u okviru predmeta Umjetna inteligencija u prvom semestru diplomskog studija računarstva na Fakultetu strojarstva, računarstva i elektrotehnike Sveučilišta u Mostaru.

Tijekom trećeg dijela istraživanja, studenti su bili podijeljeni u dvije skupine, jednu kontrolnu (KS) i jednu eksperimentalnu skupinu (ES). Nastavi je prisustvovalo ukupno 48 studenata, koji su činili neslučajni prirodni uzorak. Studenti su bili ravnomjerno raspoređeni u kontrolnu i eksperimentalnu skupinu slučajnim odabirom.

4.3.1. Inicijalna provjera znanja

Na početku trećeg dijela istraživanja provedena je inicijalna provjera znanja svih ispitanika iz područja agentske paradigme.

Inicijalna provjera znanja, u trajanju od 30 minuta, uključivala je zadatke otvorenog tipa i zadatke višestrukog izbora, a studenti su mogli ostvariti najviše 60 bodova (Prilog 17).

Studenti su prije početka provjere dobili jasne upute o načinu rješavanja, te su upoznati s razrađenim kriterijem bodovanja, usklađenim s razinom znanja potrebnom za rješavanje pojedinog zadatka. Mogućnost pogađanja umanjena je konstrukcijom pitanja.

Cilj ove provjere bio je ustanoviti predznanja ispitanika i međusobno usporediti ispitivane skupine.

Za provjeru pouzdanosti ispita korišten je Cronbach α koeficijent. Inicijalna provjera znanja spada u kategoriju pouzdanih (Cronbach $\alpha = 0.715$).

Na osnovi postignutih rezultata izračunata je aritmetička sredina (AS) kao jedna od mjera srednje vrijednosti te standardna devijacija (SD) i koeficijent varijabilnosti (KV) kao dvije mjere disperzije. Analiza rezultata ispitivanja znanja nakon obrađene inicijalne provjere znanje prikazana je u Tablici 27.

Tablica 27. Analiza rezultata inicijalne provjer znanja

Statistička mjera	Kontrolna skupina (KS)	Eksperimentalna skupina (ES)
Aritmetička sredina (AS)	21,50	20,17
Standardna devijacija (SD)	5,96	4,81
Koeficijent varijabilnosti (KV)	27,72	23,84

Postavljena je nulta hipoteza H_0 da ne postoji značajna razlika u postignućima dviju skupina.

Shapiro-Wilkov test i vizualni pregled histograma i Q-Q grafikona pokazuje da su distribucije rezultata testiranja (kontrolna i eksperimentalna skupina na FSRE-SUM) normalne, jer je značajnost Shapiro-Wilkovog testa veća od 0.05. Rezultati Shapiro-Wilkovog testa normalnosti distribucije postignutih rezultata za obje skupine prikazani su u Tablici 28.

Tablica 28. Rezultati Shapiro-Wilkovog testa normalnosti distribucije rezultata za kontrolnu i eksperimentalnu skupinu studenata na prvoj godini diplomskog studija računarstva

Skupina	Shapiro-Wilkov test normalnosti
Kontrolna skupina (KS)	0,502
Eksperimentalna skupina (ES)	0,103

Kako je utvrđeno da je distribucija ostvarenih rezultata normalna, postojanje statistički značajne razlike između rezultata kontrolne i eksperimentalne skupine studenata na prvoj godini diplomskog studija strojarstva i računarstva provjereno je parametrijskim neparnim t-testom za dvije nezavisne skupine. Rezultat analize, provedene uz pomoć statističkog paketa programske podrške SPSS, prikazan je u Tablici 29.

Tablica 29. Neparni t-test rezultata ispitivanja znanja nakon nastavne jedinice za dvije nezavisne skupine studenata diplomskih studija strojarstva i računarstva

Standardizirana vrijednost	KS i ES na FSRE-SUM
t	0,213
P	0,832

Broj stupnjeva slobode za test rezultata ostvarenih na FSRE-SUM je 46 pa razina rizika, uz razinu značajnosti 0,05, iznosi $p = 2,02$. Obzirom da su apsolutna vrijednost t-testa manja od razine rizika zadržavamo nultu hipotezu H_0 i zaključujemo da ne postoji statistički značajna razlika u ostvarenim rezultatima kontrolne i eksperimentalne skupine.

4.3.2. Istraživanje u okviru predmeta Umjetna inteligencija na diplomskom studiju računarstva

I za ovu iteraciju istraživanja odabrana je nastavna jedinica NJ2, u okviru koje se definiraju okruženja inteligentnih agenata te navode njihova osnovna svojstva i klasifikacija. Nastava iz odabrane nastavne jedinice je za kontrolnu skupinu izvedena na tradicionalan način, tj. *ex cathedra*, dok je za eksperimentalnu skupinu izvedena uz upotrebu simulacijskog okruženja *NetLogo*, dostupnog na računalima, tabletima i pametnim telefonima, te pripremljenih karakterističnih primjera koje su studenti mogli sami aktivno koristiti mijenjanjem različitih parametara simulacija.

Nastavu za obje skupine je izvodio nastavnik s višegodišnjim iskustvom u provedbi nastave iz područja računarstva i informatike. Za odabranu nastavnu jedinicu utrošeno je 180 minuta, tj. četiri školska sata. Studenti eksperimentalne skupine su, nakon uvodnog predavanja u trajanju od jednog školskog sata, preostalo vrijeme proveli u aktivnom radu s pripremljenim simulacijama.

Neposredno nakon obrađene nastavne jedinice provedeno je ispitivanje znanja za utvrđivanje razine usvojenosti obrađenih nastavnih sadržaja.

Ispitivanja znanja, u trajanju od 30 minuta, uključivalo je zadatke otvorenog tipa čije je rješavanje zahtijevalo različite razine znanja definirane Bloomovom taksonomijom u kognitivnom području (Bloom, 1956), a studenti su na svakom od njih mogli ostvariti najviše 70 bodova. Studenti su upućeni u način rješavanja zadataka i upoznati s kriterijem bodovanja usklađenim s razinom potrebnog znanja za njihovo rješavanje (Prilog 22).

Nastava za dvije skupine odvijala se neposredno jedna iza druge, tako da studenti nisu mogli izmijeniti iskustva i saznanja o nastavnoj jedinici i provedenoj provjeri znanja.

Za provjeru pouzdanosti ispita korišten je Cronbach α koeficijent. Inicijalna provjera znanja spada u kategoriju visoko pouzdanih (Cronbach $\alpha = 0.804$).

Na osnovi postignutih rezultata izračunata je aritmetička sredina (AS) kao jedna od mjera srednje vrijednosti te standardna devijacija (SD) i koeficijent varijabilnosti (KV) kao dvije mjere disperzije. Analiza rezultata ispitivanja znanja nakon obrađene nastavne jedinice prikazana je u Tablici 30.

Tablica 30. Analiza rezultata ispitivanja znanja nakon NJ2

Statistička mjera	Kontrolna skupina (KS)	Eksperimentalna skupina (ES)
Aritmetička sredina (AS)	36,63	42,42
Standardna devijacija (SD)	10,24	9,18
Koeficijent varijabilnosti (KV)	27,96	21,64

Postavljena je nulta hipoteza H_0 da ne postoji značajna razlika u postignućima dviju skupina.

Shapiro-Wilkov test i vizualni pregled histograma i Q-Q grafikona pokazuje da su distribucije rezultata testiranja (kontrolna i eksperimentalna skupina na FSRE-SUM) normalne, jer je

značajnost Shapiro-Wilkovog testa veća od 0.05. Rezultati Shapiro-Wilkovog testa normalnosti distribucije postignutih rezultata za obje skupine prikazani su u Tablici 31.

Tablica 31. Rezultati Shapiro-Wilkovog testa normalnosti distribucije rezultata za kontrolnu i eksperimentalnu skupinu studenata na prvoj godini diplomskog studija računarstva

Skupina	Shapiro-Wilkov test normalnosti
Kontrolna skupina (KS)	0,202
Eksperimentalna skupina (ES)	0,232

Kako je utvrđeno da je distribucija ostvarenih rezultata normalna, postojanje statistički značajne razlike između rezultata kontrolne i eksperimentalne skupine studenata na prvoj godini diplomskog studija strojarstva i računarstva provjereno je parametrijskim neparnim t-testom za dvije nezavisne skupine. Rezultat analize, provedene uz pomoć statističkog paketa programske podrške SPSS, prikazan je u Tablici 32.

Tablica 32. Neparni t-test rezultata ispitivanja znanja nakon nastavne jedinice za dvije nezavisne skupine studenata diplomskog studija računarstva

Standardizirana vrijednost	KS i ES na FSRE-SUM
t	-2,062
P	0,045

Broj stupnjeva slobode za test rezultata ostvarenih na FSRE-SUM je 46 pa razina rizika, uz razinu značajnosti 0,05, iznosi $p = 2,02$. Obzirom da su apsolutna vrijednost t-testa veća od razine rizika odbacujemo nultu hipotezu H_0 i zaključujemo da postoji statistički značajna razlika u ostvarenim rezultatima kontrolne i eksperimentalne skupine studenata na predmetu Umjetna inteligencija diplomskih studija strojarstva i računarstva na Fakultetu strojarstva, računarstva i elektrotehnike Sveučilišta u Mostaru. Prikazana mjera srednje vrijednosti postignutih rezultata (AS) ukazuju na činjenicu da su bolje rezultate za gradivo obrađene nastavne jedinice ostvarili studenti eksperimentalne skupine, koji su tijekom nastave koristili simulacijsko okruženje *NetLogo*.

4.3.3. Provjera znanja s vremenskim odmakom

Provjera znanja iz obrađene nastavne jedinice provedena je ponovno s vremenskim odmakom od pet tjedana, kako bi se utvrdio utjecaj korištene nastavne metode na zadržavanje usvojenog znanja iz obrađenih nastavnih sadržaja.

Ispitivanje retencije znanja provedeno je na isti način i u istim uvjetima kao i testiranje neposredno nakon obrađene nastavne jedinice. Provjera znanja, u trajanju od 30 minuta, uključivala je zadatke otvorenog tipa čije je rješavanje zahtijevalo različite razine znanja definirane Bloomovom taksonomijom u kognitivnom području (Bloom, 1956), a studenti su na svakom od njih mogli ostvariti najviše 70 bodova. Studenti su upućeni u način rješavanja zadataka i upoznati s kriterijem bodovanja usklađenim s razinom potrebnog znanja za njihovo rješavanje (Prilog 22).

Za provjeru pouzdanosti ispita korišten je Cronbach α koeficijent. Inicijalna provjera znanja spada u kategoriju visoko pouzdanih (Cronbach $\alpha = 0.846$).

Na osnovi postignutih rezultata izračunata je aritmetička sredina (AS) kao jedna od mjera srednje vrijednosti te standardna devijacija (SD) i koeficijent varijabilnosti (KV) kao dvije mjere disperzije. Analiza rezultata ispitivanja retencije znanja nakon pet tjedana prikazana je u Tablici 33.

Tablica 33. Analiza rezultata ispitivanja retencije znanja nakon pet tjedana

Statistička mjera	Kontrolna skupina (KS)	Eksperimentalna skupina (ES)
Aritmetička sredina (AS)	32,38	35,83
Standardna devijacija (SD)	5,92	5,14
Koeficijent varijabilnosti (KV)	18,28	14,35

Postavljena je nulta hipoteza H_0 da ne postoji značajna razlika u postignućima dviju skupina.

Shapiro-Wilkov test i vizualni pregled histograma i Q-Q grafikona pokazuje da su distribucije rezultata testiranja (kontrolna i eksperimentalna skupina na FSRE-SUM) normalne, jer je značajnost Shapiro-Wilkovog testa veća od 0.05. Rezultati Shapiro-Wilkovog testa normalnosti distribucije postignutih rezultata za obje skupine prikazani su u Tablici 34.

Tablica 34. Rezultati Shapiro-Wilkovog testa normalnosti distribucije rezultata za kontrolnu i eksperimentalnu skupinu studenata na prvoj godini diplomskog studija računarstva

Skupina	Shapiro-Wilkov test normalnosti
Kontrolna skupina (KS)	0,235
Eksperimentalna skupina (ES)	0,117

Kako je utvrđeno da je distribucija ostvarenih rezultata normalna, postojanje statistički značajne razlike između rezultata kontrolne i eksperimentalne skupine studenata na prvoj godini diplomskog studija strojarstva i računarstva provjereno je parametrijskim neparnim t-testom za dvije nezavisne skupine. Rezultat analize, provedene uz pomoć statističkog paketa programske podrške SPSS, prikazan je u Tablici 35.

Tablica 35. Neparni t-test rezultata ispitivanja retencije znanja nakon pet tjedana za dvije nezavisne skupine studenata diplomskog studija računarstva

Standardizirana vrijednost	KS i ES na FSRE-SUM
T	-2,162
P	0,036

Broj stupnjeva slobode za test rezultata ostvarenih na FSRE-SUM je 46 pa razina rizika, uz razinu značajnosti 0,05, iznosi $p = 2,02$. Obzirom da su apsolutna vrijednost t-testa veća od razine rizika odbacujemo nultu hipotezu H_0 i zaključujemo da postoji statistički značajna razlika u ostvarenim rezultatima kontrolne i eksperimentalne skupine studenata na predmetu Umjetna inteligencija diplomskih studija strojarstva i računarstva na Fakultetu strojarstva, računarstva i elektrotehnike Sveučilišta u Mostaru. Prikazana mjera srednje vrijednosti postignutih rezultata (AS) ukazuju na činjenicu da su bolje rezultate provjere retencije znanja nakon pet tjedana ostvarili studenti eksperimentalne skupine, koji su tijekom nastave koristili simulacijsko okruženje *NetLogo*.

Ponovljena ispitivanje znanja, s vremenskim odmakom od pet tjedana, pokazuje kako su studenti, u prosjeku, postigli lošije rezultate u odnosu na ispitivanje znanja neposredno nakon održane nastave. Ovo je očekivan rezultat, s obzirom da je prošlo određeno vrijeme u kojem studenti nisu koristili stečeno znanje. Usporedbom rezultata studenata koji su tijekom nastave

koristili simulacijsko okruženje *NetLogo* s rezultatima učenika kojima je nastava održana na klasičan način *ex cathedra*, vidljivo je kako studenti koji su koristili simulacijsko okruženje *NetLogo* i nakon određenog vremena posjeduju više znanja od učenika koji su na klasičan, frontalni način. Brojni autori potvrdili su kako učenici koji su aktivno uključeni u učenje zadržavaju informacije duže od onih koji su pasivni sudionici nastave (Modell, 1996; Michael, 2006). Znanja stečena primjenom simulacijskog okruženja zbog toga pokazuju veću stabilnost i trajnost od onog znanja koje je stečeno u okviru frontalne nastave.

4.4. Prijedlog modela poučavanja agentske paradigme korištenjem simulacijskih okruženja

Na osnovu analize rezultata sva tri dijela istraživanja predlažemo sljedeći model poučavanja, oblikovan prema smjernicama koje iznosi Aggerwall (Aggerwall, 2014).

- **Ishodi učenja:** Nakon završenog nastavnog ciklusa student treba moći definirati osnovne pojmove agentske paradigme, razumijeti razlike između pojedinih vrsta agenata, okruženja i interakcija između agenata u višeagentskim sustavima, moći samostalno osmisliti idejni model okruženja u kojem djeluje jedan ili više agenata prema zadanim karakteristikama.
- **Okruženje:** Studentove reakcije tijekom nastave mogu se optimalno pratiti u računalnoj učionici u kojoj će istovremeno boraviti manji broj studenata. Iznimno, zbog činjenice da postoje i mrežne inačice alata za modeliranje i simuliranje zasnovane na agentima koje omogućavaju i pristup simulacijskim okruženjima putem mobilnih uređaja, nastava se optimalno može održavati u klasičnoj učionici, ali bi broj studenta trebao biti ograničan zbog kvalitete izvedbe.
- **Kriteriji izvedbe:** Od studenata se očekuje samostalno praćenje nastave, manipuliranje onim parametrima simulacije za koje ta mogućnost postoji i praćenje rezultata tih promjena u nastavnim jedinicama koje se izvode uz pomoć simulacijskih okruženja. Poželjno je da, na osnovu opaženog studenti mogu samostalno donijeti zaključke o karakteristikama korištenih koncepata agentske paradigme.
- **Mehanizmi koji osiguravaju interakciju studenata međusobno i sa okruženjem:** Interakcija sa okruženjem ostvarena je mogućnošću upravljanja pojedinim parametrima simulacije u nastavnim jedinicama koje se izvode uz pomoć simulacijskih okruženja. Zadaci grupnog rada ili rada u parovima omogućit će međusobnu interakciju studenata i razmjenu stečenih znanja.
- **Znanstveni postupak kojim se modificira ponašanje studenata:** Definiranje problema i dobro osmišljeni i opisani primjeri i zadaci zahtijevat će od studenata aktivno sudjelovanje u nastavnom procesu. Praćenje njihove reakcije u učionici omogućit će nastavniku identificiranje problema u razumijevanju sadržaja, kao i pojedinaca koji su sadržaj dobro usvojili te omogućiti aktivno djelovanje kako bi se moguće pogrešne koncepcije na vrijeme ispravile.

Predloženi model poučavanja agentske paradigme je hibridni model koji uključuje klasičan, *ex cathedra* pristup izvođenju nastave i pristup nastavi koji koristi simulacijska okruženja

realizirana alatima za modeliranje i simuliranje zasnovanim na agentima uz pripremljene karakteristične interaktivne simulacijske modele.

Nastava koja koristi simulacijska okruženja izvodila bi se prilikom obrade nastavnih jedinica:

- Definicija, osnovna svojstva i vrste agenata,
- Definicija, osnovna svojstva i vrste okruženja,
- Interakcija agenata i okruženja i međusobna interakcija agenata u višeagentskim sustavima.

Klasična nastava izvodila bi se pri obradi nastavne jedinice

- Područja primjene agenata.

Izbor alata za modeliranje i simuliranje zasnovanog na agentima prepušten je nastavniku na predmetu koji uključuje nastavne sadržaje iz područja agentske paradigme, prema karakteristikama infrastrukture koju u nastavi može koristiti, predznanjima studenta koja se odnose na ranije naučene programske jezike i razvojna okruženja i formalna predznanja stečena prethodno odslušanim i polaganim ispitima srodnog područja

Na studijima računarstva i informatike na Sveučilištu u Mostaru, prema nastavnim planovima i programima, ne postoji predmet čiji se sadržaj u potpunosti odnosi na opisane koncepte agentske paradigme, već se oni javljaju kao pojedine nastavne jedinice i cjeline u okviru predmeta šireg sadržaja. To uvjetuje i vremenska ograničenja za izvođenje nastave iz područja agentske paradigme u skladu sa nastavnim planom i programom.

Kako ova suvremena tema ima sve širu primjenu u različitim granama ljudskog djelovanja ističemo potrebu za cjelovitijom i opsežnijom pažnjom koju je potrebno posvetiti području agentske paradigme, uvođenjem novih predmeta koji bi se u potpunosti odnosili na izučavanje ovog područja i njegovu moguću primjenu.

Postojeći modeli poučavanja agentske paradigme, kao jednog dijela nastavnih predmeta u okviru kojih je istraživanje provedeno oslanjaju se isključivo na klasičan, *ex cathedra* pristup. Istraživanje je dokazalo prednosti predloženog modela koji uključuje i nastavu uz pomoć simulacijskih okruženja realiziranih nekim od alata za modeliranje i simuliranje zasnovanim na agentima.

Ovaj model je kompletniji od tradicionalnog, sadrži karakteristične primjere pomoću kojih studenti mogu samostalno upravljati simuliranim događajima i na taj način, aktivnim sudjelovanjem u nastavnom procesu bolje usvojiti koncepte obrađene nastavne cjeline, što su rezultati ovog istraživanja i pokazali.

Kolekcija primjera osmišljenih i razrađenih u okviru ovog istraživanja u suradnji sa predmetnim nastavnicima sa višegodišnjim iskustvom u području agentske paradigme i umjetne inteligencije javno je dostupna kroz priloge ovog rada i može se koristiti u nastavi iz područja agentske paradigme.

5. ZAKLJUČAK

Rezultati provedenog istraživanja ukazuju na pozitivan učinak primjene simulacijskih okruženja u poučavanju složenih i apstraktnih koncepata agentske paradigme. Provedeni testovi u prvom dijelu istraživanja pokazuju statistički značajnu razliku u rezultatima kontrolne skupine i dviju eksperimentalnih skupina u korist eksperimentalnih skupina koje su u nastavi koristili jedno od odabranih simulacijskih okruženja *Cellular* ili *NetLogo*. Istovremeno, ne postoji statistički značajna razlika između ostvarenih rezultata dvije eksperimentalne skupine koje su u nastavi koristile različite simulacijske alate.

S obzirom da nije postojala razlika u razini predznanja tri skupine i da su sve skupine istovremeno prisustvovala nastavi u kojoj su obrađivane iste nastavne jedinice i korišteni istovjetni primjeri agentskih i višeagentskih sustava kojima su prikazani i objašnjeni koncepti agentske paradigme, pri čemu je jedina razlika bila nastavna metoda koja je pri tome korištena, razliku u postignutim rezultatima pripisujemo upravo korištenoj nastavnoj metodi. Zaključujemo da je simulacija učinkovito nastavno sredstvo u poučavanju apstraktnih pojmova područja agentske paradigme i da je pozitivan učinak njene upotrebe u nastavi dokazan ovim istraživanjem.

Nizom intervjua sa studentima potvrđena je razina usvojenosti koncepata agentske paradigme iskazana rezultatima provjera znanja nakon nastavnih jedinica. Studenti su, tijekom intervjua, iskazali veće zadovoljstvo radom sa simulacijskim okruženjem *NetLogo*, ističući stav da je ovo okruženje primjerenije visokoškolskoj razini obrazovanja.

Na osnovi rezultata provedenih provjera znanja nakon obrađenih nastavnih jedinica i njihove statističke analiza može se izvesti zaključak da je za prve tri nastavne jedinice, koje obrađuju pojmove agenata i okruženja, njihove interakcije te interakcije agenata u višeagentskim sustavima, eksperimentalna nastava, uz korištenje simulacijskih okruženja i pripremljenih primjera, imala značajno bolji učinak od tradicionalne nastave. Za četvrtu nastavnu jedinicu koja obrađuje područja primjene agenata, bolji učinak je imala tradicionalna nastava što je, prema mišljenju studenata eksperimentalnih skupina iskazanih u pojedinačnim standardiziranim intervjuima rezultat njihove usmjerenosti ka primjenama prikazanim u pripremljenim primjerima simulacija i zanemarivanja ostalih mogućih primjena.

Kako je uzorak u prvom dijelu istraživanja činilo 27 studenata podijeljenih u tri skupine, dobiveni rezultati su u drugom dijelu istraživanja testirani na većem uzorku od ukupno 152 studenta (80 studenata na predmetu Uvod u računarstvo na prvoj godini preddiplomskog studija Informatike na Fakultetu prirodoslovno-matematičkih i odgojnih znanosti Sveučilišta u Mostaru i 72 studenta na predmetu Umjetna inteligencija na prvoj godini diplomskih studija strojarstva i računarstva na Fakultetu strojarstva, računarstva i elektrotehnika Sveučilišta u Mostaru), i odabranoj nastavnoj jedinici NJ2 koja definira okruženja inteligentnih agenata i navodi njihova osnovna svojstva i klasifikaciju, uz korištenje simulacijskog okruženja *NetLogo*. U trećem dijelu istraživanja otkolonjeni su uočeni nedostaci iz drugog dijela.

Analiza rezultata drugog i trećeg dijela istraživanja potvrdila je zaključke iz prvog dijela, tj. utvrđena je statistički značajna razlika u postignućima kontrolne i eksperimentalne skupine. Ovi rezultati dodatno potvrđuju pozitivan stav o uključenju simulacijskih alata kao nastavnog sredstva u poučavanja apstraktnih koncepata agentske paradigme. Dodatno je, analizom rezultata ispitivanja znanja sa vremenskim odmakom, u trećem dijelu istraživanja pokazano da znanje stečena primjenom simulacijskog okruženja pokazuju veću stabilnost i trajnost od znanja stečenog u okviru frontalne nastave

Usporedba postignuća dviju kontrolnih skupina studenta s različitim razina visokoškolskog obrazovanja, kao i postignuća dviju odgovarajućih eksperimentalnih skupina, nije pokazala statistički značajnu razliku. Ipak, bolji rezultati studenata diplomskog studija strojarstva i računarstva u odnosu na studente preddiplomskog studija informatike, iskazani osjetno višom srednjom vrijednošću ostvarenih rezultata, upućuju na zaključak da je područje agentske paradigme, zbog svoje razine apstrakcije i potrebnih predznanja, primjerenije nastaviti na višim godinama visokoškolskog obrazovanja.

Istraživanjem je, dakle, potvrđena polazna hipoteza prema kojoj će studenti informatike i računarstva postići bolje rezultate u usvajanju i razumijevanju osnovnih koncepata agentske paradigme uz poučavanje temeljeno na korištenju simulacijskih okruženja u odnosu na tradicionalan način poučavanja *ex cathedra*.

Također su dani odgovori na postavljena istraživačka pitanja. S obzirom na rezultate provedenog istraživanja, način poučavanja agentske paradigme utječe na uspješnost studenata u usvajanju i razumijevanju nastavnih sadržaja ovog područja. Iako nisu utvrđene značajne razlike u uspješnosti usvajanja i razumijevanja nastavnih sadržaja iz područja agentske paradigme koje su posljedica odabranog simulacijskog okruženja, studenti su iskazali veće

zadovoljstvo simulacijskim okruženjem *NetLogo*, dok su kao nedostatak *Cellulara* naveli njegov "djetinjast" i "neozbiljan" izgled. Zbog toga je za drugi dio istraživanja odabrano simulacijsko okruženje *NetLogo*.

Pouzdanost provedenih ispitivanja znanja osigurana je usklađivanjem načina bodovanja, na način da su bodovi bili težinski usklađeni s razinom Bloomeove taksonomije odgojno-obrazovnih ciljeva potrebne za točan odgovor na postavljano pitanje. Studenti se prije svake od provjera znanja dobili jasne upute i objašnjen im je kriterij bodovanja. Ispitivanja znanja prevedena su u istom periodu za sve sudionike. Pouzdanost provjera znanja na kraju svake nastavne jedinice je analizirana i potvrđena određivanjem Cronbach alpha koeficijenta pouzdanosti, koji je u svim slučajevima bio veći od 0,7 što provedene provjere znanja svrstava u kategoriju pouzdanih.

Prirodno okruženje sudionika, u kojem je istraživač/nastavnik dio istraživanog svijeta, osigurala je, prema naturalističkim istraživanjima, valjanost provjere znanja. Dodatno je unutarnja valjanost osigurana triangulacijom istraživača (kroz aktivno sudjelovanje više nastavnika u prvom i drugom dijelu istraživanja), prostornom triangulacijom (koja će se postići uključivanjem dvije institucije u drugom dijelu istraživanja), triangulacijom unutar metoda (kroz ponavljanje istraživanja učinka simulacije u različitim vremenskim točkama) kao i triangulacijom između metoda (jer su se, osim kvantitativnih podataka o nastavnim metodama, prikupili i stavovi i dojmovi studenata o održanoj nastavi). Konstruktna valjanost je osigurana opsežnim pretraživanjem literature i uzimanjem u razmatranje različitih značenja pojedinih konstrukata agentske paradigme. Sadržajna valjanost provjere znanja nakon svake nastavne jedinice osigurana je činjenicom da su se provjere odnosile na ograničeni opseg sadržaja kao i da su provjere znanja prethodno sadržajno usuglašene s predmetnim nastavnicima s dugogodišnjim iskustvom u nastavi iz ovog područja.

Zbog svega ranije navedenog, predložen je model poučavanja agentske paradigme, primjeren višim godinama studija informatike i računarstva. Ovaj model uključuje korištenje simulacijskih okruženja uz pripremljene karakteristične interaktivne simulacijske modele u nastavnim jedinicama koje obrađuju definicije, osnovna svojstva i vrste agenata i okruženja u kojima oni djeluju, interakciju agenta i okruženja te međusobnu interakciju agenata u višeagentskim sustavima. U nastavnoj jedinici koja obrađuje područja primjene agenata, predloženi model preporuča klasičan, *ex cathedra* pristup izvođenju nastave.

6. LITERATURA

Abar, S., Theodoropoulos, G. K., Lemarinier, P., & O'Hare, G. M. (2017). Agent Based Modelling and Simulation Tools: A Review of the State-of-Art Software. *Computer Science Review*, 24, 13-33.

Adeyeri, M. K., Mpofu, K., & Olukorede, T. A. (2015). Integration of Agent Technology into Manufacturing Enterprise: A Review and Platform for Industry 4.0. *Proceedings of the 2015 International Conference on Industrial Engineering and Operations Management* (pp. 1-10). IEEE.

Adomavicius, G., Gupta, A., & Zhdanov, D. (2009). Designing Intelligent Software Agents for Auctions with Limited Information Feedback. *Information Systems Research*, 20(4), 507-526.

Agentscript: Minimalist Agent Based Modeling Framework based on NetLogo. (2013). Retrieved from: <http://agentscript.org>

Aggerwal, J. C. (2014). *Essentials of Educational Technology: Inovation in Teaching-Learning* (3rd Ed.). New Delhi, India: Vikas Publishing Ltd.

Al Masum, S. M., Islam, M. T., & Ishizuka, M. (2006). ASNA: An Intelligent Agent for Retrieving and Classifying News on the Basis of Emotion-Affinity. *Proceedings of the 2006 International Conference on Computational Inteligence for Modelling Control and Automation and International Conference on Intelligent Agents Web Technologies and International Commerce* (pp. 133-138). IEEE.

Alimisis, D. (2010). Introducing Robotics in Schools - Post-TERECop Experiences from a Pilot Educational Program, *Proceedings of International Conference on Simulation, Modeling and Programming for Autonomous Robots* (pp. 575-585).

Allan, R. J. (2010). *Survey of Agent Based Modelling and Simulation Tools* (Technical report, ISSN 1362-0207). Science & Technology Facilities Council, Warrington, UK.

Alshahad, H. F. (2009). Design and Implementation of Email Filtering Agent by using Clustering Algorithm, *Journal of Kerbala University*, 7(4), 163-172.

Ashton, K. (2009). That "Internet of things" Thing. *RFID Journal*, 22(7), 97-114.

Augar, N. (2003). Intelligent Information Agents: Search Engines of the Future?. *Proceedings of the First Australian Undergraduate Students' Computing Conference* (pp. 7-12).

Babić, N. (2007). Konstruktivizam i pedagogija, *Pedagogijska istraživanja*, 4(2), 217-229.

- Bandura, A., & Walters, R. (1963). *Social Learning and Personality Development*. New York, NY: Holt, Rinehart & Winston.
- Banks, J., Carson, I. I., Nelson, B. L., & Nicol, D. M. (2005). *Discrete-Event System Simulation*. Harlow, UK: Pearson.
- Belbase, S. (2014). Radical Versus social Constructivism: An Epistemological-Pedagogical Dilemma. *International Journal of Contemporary Educational Research*, 1(2), 98-112.
- Ben-Ari, M. (2001). Constructivism in Computer Science Education, *Journal of Computers in Mathematics and Science Teaching*, 20(1), 45-73.
- Bergen, M., Denzinger, J., & Kidney, J. (2002). *Teaching Multi-Agent Systems with the Help of ARES: Motivation and Manual*.
- Blikstein, P., & Wilensky, U. (2006). From Inert to Generative Modeling: Two Case Studies of Multi-Agent-Based Simulation in Undergraduate Engineering Education. *Annual Meeting of the American Educational Research Association 2006*.
- Blikstein, P., & Wilensky, U. (2008). Implementing Multi-Agent Modeling in the Classroom: Lessons from Empirical Studies in Undergraduate Engineering Education. *International Conference of the Learning Sciences 2008*.
- Bloom, B. S. (1956). *Taxonomy of Educational Objectives. Vol. 1: Cognitive Domain*. New York: McKay, 20-24.
- Bodner, G. M. (1986). Constructivism: A Theory of Knowledge, *Journal of Chemical Education*, 63(10), 873-878.
- Bond, A. H., & Gasser, L. (Eds.). (1998). *Readings in Distributed Artificial Intelligence*, San Mateo, CA: Morgan Kaufmann Publishers
- Bohté, S. M., Langdon, W. B., & Poutre, H. L. (2000). On Current Technology for Information Filtering and User Profiling in Agent-Based Systems, Part I: A Perspective. *CWI, Centre for Mathematics and Computer Science*, 1-12.
- Bollen, L., & van Joolingen, W. R. (2013). SimSketch: Multiagent Simulations based on Learner-Created Sketches for Early Science Education. *IEEE Transactions on Learning Technologies*, 6(3), 208-216.
- Bora, Ş., Evren, V., Emek, S., & Çakırlar, I. (2019). Agent-Based Modeling and Simulation of Blood Vessels in the Cardiovascular System. *Simulation*, 95(4), 297-312.

- Bordini, R. H. (2005). A Recent Experience in Teaching Multi-Agent Systems using Jason. In Proceedings of the 2nd AAMAS Workshop on Teaching Multi-Agent Systems.
- Borich, G. D., & Fuller, F. F. (1974). Teacher Concerns Checklist: An Instrument for Measuring Concerns for Self, Task, and Impact, University of Texas Research and Development Center for Teacher Education.
- Bower, G. H. & Hilgard, E. R. (1981). Theories of Learning (5th ed.), Upper Saddle River, NJ: Prentice Hall.
- Brenner, W., Zarnekow, R., & Wittig, H. (2012). Intelligent Software Agents: Foundations and Applications. Berlin - Heidelberg: Springer Science & Business Media.
- Brewington, L., Engie, K., Walsh, S.J., & Mena, C. (2013). Collaborative Learning and Global Education: Human–Environment Interactions in the Galápagos Islands, Ecuador. *Journal of Geography*, 112(5), 179–192.
- Brinson, J. R. (2015). Learning Outcome Achievement in Non-Traditional (Virtual and Remote) vs Traditional (Hands-On) Laboratories: A Review of the Empirical Research. *Computers & Education*, 87, 218-237.
- Brooks, J. G., Brooks, M. G. (1993). In Search of Understanding: The Case for Constructivist Classrooms, Alexandria, VA: Association for Supervision and Curriculum Development.
- Castelfranchi, C. (1994). Guarantees for Autonomy in Cognitive Agent Architecture., International Workshop on Agent Theories, Architectures, and Languages (pp. 56-70). Berlin - Heidelberg: Springer.
- Cavezzali, A., Girotti, A., & Rabino, G. (2003). Multi-Agents Systems and Territory: Concepts, Methods and Applications, Proceedings of the 43rd Congress of the European Regional Science Association: Peripheries, Centres, and Spatial Development in the New Europe, Jyväskylä, Finland.
- Checkland, P. (1981). *Systems Thinking, Systems Practice*. Chichester, UK: John Wiley & Sons.
- Chen, B., & Cheng, H. H. (2010). A Review of the Applications of Agent Technology in Traffic and Transportation Systems. *IEEE Transactions on Intelligent Transportation Systems*, 11(2), 485-497.

- Chen, C. (2003). A Constructivist Approach to Teaching: Implications in Teaching, Computer Networking, Information Technology, Learning, and Performance Journal, (Vol. 21 (2), pp. 17-27).
- Chen, K. J., & Barthès, J. P. (2007). MemoPA: Intelligent Personal Assistant Agents with a Case Memory Mechanism. Proceedings of the International Conference on Intelligent Computing (pp. 1357-1367). Berlin – Heidelberg: Springer.
- Clark, R. C., & Mayer, R. E. (2011). E-Learning and the Science of Instruction: Proven Guidelines for Consumers and Designers of Multimedia Learning (3rd Ed.), San Francisco, CA: John Wiley & Sons.
- Clayton, G., & Gizelis, T. I. (2005). Learning through Simulation or Simulated Learning? An Investigation into the Effectiveness of Simulations as a Teaching Tool in Higher Education. In Proceedings from British International Studies Association Conference.
- Clement, B. J., & Durfee, E. H. (1999). Identifying and Resolving Conflicts among Agents with Hierarchical Plans. Proceedings of the AAI Workshop on Negotiation: Settling Conflicts and Identifying Opportunities, AAI Technical Report WS-99 (Vol. 12, pp. 6-11).
- Coburn, W. (1993). Contextual Constructivism: The Impact of Culture on the Learning and Teaching of Science. In: Tobin, K. (Ed.), The Practice of Constructivism in Science Education, (pp. 51-69), Hillsdale, NJ: Lawrence-Erlbaum.
- Coen, M. H. (1994), SodaBot: A Software Agent Construction System (Working paper No. AI-TR-1493). Massachusetts Institute of Technology Artificial Intelligence Lab.
- Cohen, L., Manion, L., & Morrison, K. (2013). Research Methods in Education, 7th Ed., London: Routledge.
- Cole, M., Wertsch, J.V. (1996). Beyond the Individual: Social Antinomy in Discussions of Piaget and Vygotsky. Human development, 39 (5), 250-256.
- Comparison of Agent-Based Modeling Software. (2019, February 11). Retrieved from https://en.wikipedia.org/wiki/Comparison_of_agent-based_modeling_software
- Concannon, K. H., Hunter, K. I., & Tremble, J. M. (2003). Dynamic scheduling II: SIMUL8-Planner Simulation-Based Planning and Scheduling. In Proceedings of the 35th Conference on Winter Simulation: Driving Innovation (pp. 1488-1493).

- Cooper, J., & James, A. (2009). Challenges for Database Management in the Internet of Things. *IETE Technical Review*, 26(5): 320-329
- Coyle, R. G. (1996). *System Dynamics Modelling: A Practical Approach*. London, UK: Chapman & Hall.
- Das, S., Wu, C., & Truskowski, W. (2001). Distributed Intelligent Planning and Scheduling for Enhanced Spacecraft Autonomy. *Proceedings of the 2001 AAAI Spring Symposium Series*, Palo Alto, CA.
- Denzinger, J., & Kidney, J. (2005). Teaching Multi-Agent Systems using the ARES Simulator. *Innovation in Teaching and Learning in Information and Computer Sciences*, 4(3), 1-23.
- Diamond, B., Krahl, D., Nastasi, A., & Tag, P. (2010). ExtendSim Advanced Technology: Integrated Simulation Database. *IEEE 2010 Winter Simulation Conference*, 32-39.
- Doolittle, P. E., & Camp, W. G. (1999). Constructivism: The Career and Technical Education Perspective. *Journal of Vocational and Technical Education*, 16(1).
- Dougiamas, M. (1998). A Journey into Constructivism, Retrieved from: <https://dougiamas.com/archives/a-journey-into-constructivism/>
- Dunleavy, J., Milton, P., & Crawford, C. (2010). The Search for Competence in the 21st century. *Quest Journal* 2010, 2.
- Durfee, E. H., & Lesser, V. (1989). Negotiating Task Decomposition and Allocation using Partial Global Planning. In Gasser, L., & Huhns, M. (Eds.), *Distributed Artificial Intelligence* (vol. II, pp. 229-244). London: Pitman Publishing / San Mateo, CA: Morgan Kaufmann.
- D'Souza, R. M., Lysenko, M., & Rahmani, K. (2007). SugarScape on Steroids: Simulating over a Million Agents at Interactive Rates. In *Proceedings of Agent2007 Conference*. Chicago, IL
- English, H. B., & English, A. C. (1958). *A Comprehensive Dictionary of Psychological and Psychoanalytic Terms*. London: Longman.
- Ernest, P. (1995). The One and the Many. In Steffe, L. P., & Gale, J. (Eds.), *Constructivism in Education* (pp. 459–486). Hillsdale, NJ: Lawrence Erlbaum.
- Ertmer, P., & Newby, T. (1993). Behaviorism, Cognitivism, Constructivism: Comparing Critical Features from an Instructional Design Perspective, *Performance Improvement Quarterly*, 6(4), 50-72.

- Farias, G., Fabregas, E., Peralta, E., Vargas, H., Dormido-Canto, S., & Dormido, S. (2019). Development of an Easy-to-Use Multi-Agent Platform for Teaching Mobile Robotics. *IEEE Access*, 7, 55885-55897.
- Ferber, J. (1999). *Multi-Agent Systems: An Introduction to Distributed Artificial Intelligence*, Boston, MA: Addison-Wesley Longman Publishing Co.
- Florian, R. V. (2003). *Autonomous Artificial Intelligent Agents* (Technical Report Coneural-03-01). Center for Cognitive and Neural Studies, Cluj - Napoca, Romania.
- Fonseca, J. M., Mora, A. D., & de Caparica, M. (2004). Personal Assistant Autonomous Agents for Intelligent e-Learning Systems. *Proceedings of the 3rd IASTED International Conference on Web-Based Education* (pp. 16-18).
- Fornara, N. (2003). *Interaction and Communication among Autonomous Agents in Multiagent Systems* (Doctoral dissertation). Università della Svizzera Italiana, Retrieved from: <http://doc.rero.ch/record/4379>
- Fortino, G., Gravina, R., Russo, W., & Savaglio, C. (2017). Modeling and Simulating Internet-of-Things Systems: A Hybrid Agent-Oriented Approach. *Computing in Science & Engineering*, 19(5), 68-76.
- Franklin, S., & Graesser, A. (1996), Is it an Agent, or Just a Program? A Taxonomy for Autonomous Agents, *International Workshop on Agent Theories, Architectures, and Languages* (pp. 21-35). Berlin - Heidelberg: Springer.
- Fuller, F. F. (1969). Concerns of Teachers: A Developmental Conceptualization. *American Educational Research Journal*, 6(2), 207-226.
- Fuller, F., Bown, O. (1975). *Becoming a Teacher*. In Ryan, K. (Ed.), *Teacher Education, Part II: The 74th Yearbook of the National Society for the Study of Education* (pp. 25-52). Chicago, IL: University of Chicago Press.
- Galliers, J. R. (1988). *A Theoretical Framework for Computer Models of Cooperative Dialogue, Acknowledging Multiagent Conflict*, (Doctoral Thesis). Open University UK.
- Genesereth M. R., & Ketchpel, S. P. (1994). Software Agents. *Communications of the ACM*, 37(7), 48-53.

- Gerla, M., Lee, E. K., Pau, G., & Lee, U. (2014). Internet of Vehicles: From Intelligent Grid to Autonomous Cars and Vehicular Clouds. *Proceedings of the 2014 IEEE World Forum on Internet of Things* (pp. 241-246). IEEE.
- Ginot, V., & Le Page, C. (1998). MOBIDYC: A Generic Multi-agents Simulator for Modeling Populations Dynamics. In *Proceedings of International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems* (pp. 805-814). Berlin, Heidelberg, Germany, Springer.
- Ginot, V., Le Page, C., & Souissi, S. (2002). A Multi-Agents Architecture to Enhance End-User Individual-Based Modelling. *Ecological Modelling*, 157(1), 23-41.
- Godoy, D., & Amandi, A. (2000). PersonalSearcher: An Intelligent Agent for Searching Web Pages. *Advances in Artificial Intelligence* (pp. 43-52). Berlin – Heidelberg: Springer.
- Gomes, L. (2011). Searching for Intelligent E-Mail Agents, *MIT Technology Review*, Retrieved from: <https://www.technologyreview.com/s/424940/searching-for-intelligent-e-mail-agents/>
- Goodchild, S., & Sriraman, B. (2012). Revisiting the Didactic Triangle: From the Particular to the General. *ZDM - The International Journal on Mathematics Education*, 44(5), 581-585.
- Gordon, J. (1996). Tracks for Learning. *Metacognition and Learning Technologies. Australian Journal of Educational Technology*, 12 (1), 46-55.
- Gubbi, J., Buyya, R., Marušić, S., & Palaniswami, M. (2013). Internet of Things (IoT): A Vision, Architectural Elements, and Future Directions. *Future Generation Computer Systems*, 29(7), 1645-1660.
- Gu, X., & Blackmore, K. L. (2015). A Systematic Review of Agent-Based Modelling and Simulation Applications in the Higher Education Domain. *Higher Education Research & Development*, 34(5), 883–898.
- Hewitt, C. (1977). Viewing Control Structures as Patterns of Passing Messages. *Artificial Intelligence*, 8(3), 323-364.
- Hylton, A., & Sankaranarayanan, S. (2012). Application of Intelligent Agents in Hospital Appointment Scheduling System. *International Journal of Computer Theory and Engineering*, 4(4), 625-630.

- Jansen, J. (1997). Using an Intelligent Agent to Enhance Search Engine Performance. *First Monday*, 2(3).
- Jennings, N. R. (1996). Using Intelligent Agents to Manage Business processes. *Proceedings of the IEE Colloquium on Intelligent Agents and Their Applications (Digest No: 1996/101)* (pp. 5-1). IET.
- Jennings, N. R., Sycara, K., & Wooldridge, M. (1998). A Roadmap of Agent Research and Development. *Autonomous Agents and Multi-Agent Systems*, 1(1), 7-38
- Jennings, N. R., & Wooldridge, M. (1998). Applications of Intelligent Agents. In *Agent technology* (pp. 3-28). Berlin, Heidelberg: Springer.
- Jiao, J. R., You, X., & Kumar, A. (2006). An Agent-Based Framework for Collaborative Negotiation in the Global Manufacturing Supply Chain Network. *Robotics and Computer-Integrated Manufacturing*, 22(3), 239-255.
- Joyce, B. R., Weil, M., & Calhoun, E. (2015). *Models of Teaching* (9th Ed.). London, UK: Pearson.
- Jukić, R. (2013). Konstruktivizam kao poveznica poučavanja sadržaja prirodosnanstvenih i društvenih predmeta, *Pedagogijska istraživanja*, 10(2), 241-263.
- Kadar, M., & Muntean, M. (2014). Intelligent and Collaborative Multi-Agent System to Generate Automated Negotiation for Sustainable Enterprise Interoperability. *Agent and Multi-Agent Systems: Technologies and Applications* (pp. 213-222). Springer.
- Kahn, K. (2007). Building Computer Models from Small Pieces. In *Proceedings of the 2007 Summer Computer Simulation Conference, Society for Computer Simulation International* (pp. 931-936), San Diego, USA.
- Kahn, K., & Noble, H. (2010). The BehaviourComposer 2.0: A Web-based Tool for Composing NetLogo Code Fragments. In Clayson, J., & Kalaš, I. (Eds.) *Proceedings of 12th EuroLogo Conference Constructionism 2010: Constructionists Approaches to Creative Learning, Thinking and Education, Lessons for the 21st Century*, (pp 1-14). Paris: France.
- Kebreau, S., Pierre, S., & Probst, W. G. (1998). Modelling Intelligent Agents for Information Filtering. *Proceedings of the IEEE Canadian Conference on Electrical and Computer Engineering* (Vol. 2, pp. 569-572). IEEE.

- Kilbane, C. R., Milman, N. B. (2014). *Teaching Models: Designing Instruction for 21st Century Learners*. London, UK: Pearson.
- Klopfer, E., Scheintaub, H., Huang, W., Wendel, D., & Roque, R. (2009). The Simulation cycle: Combining Games, Simulations, Engineering and Science using StarLogo TNG. *E-Learning and Digital Media*, 6(1), 71-96.
- Klopfer E., Scheintaub H., Huang W., Wendel D. (2009). StarLogo TNG: Making Agent-Based Modeling Accessible and Appealing to Novices. In Komosinski M., & Adamatzky A. (Eds.) *Artificial Life Models in Software*. London, UK: Springer.
- Klügl, F., & Puppe, F. (1998). The Multi-Agent Simulation Environment SeSaAm. In *Proceedings of Workshop Simulation in Knowledge-Based Systems*, (Report: tr-ri-98-194 of Reihe Informatik, Paderborn).
- Klügl, F., Herrler, R., & Fehler, M. (2006). SeSAM: Implementation of Agent-Based Simulation using Visual Programming. In *Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems*, 1439-1440.
- Krahl, D. (2008). ExtendSim 7. *IEEE 2008 Winter Simulation Conference*, 215-221.
- Krčadinac, U., Stanković, M., Kovanović, V., & Jovanović, J. (2009). Intelligent Multi-Agent Systems. In *Encyclopedia of Information Communication Technology* (pp. 464-469). IGI Global.
- Komosiński, M. (2003). The Framsticks System: Versatile Simulator of 3D Agents and their Evolution. *Kybernetes*, 32(1/2), 156-173.
- Komosiński, M., & Ulatowski, S. (1999). Framsticks: Towards a Simulation of a Nature-like World, Creatures and Evolution. In *Proceedings of European Conference on Artificial Life* (pp. 261-265). Springer, Berlin, Heidelberg, Germany: Springer.
- Kottonau, J. (2011). An Interactive Computer Model for Improved Student Understanding of Random Particle Motion and Osmosis. *Journal of Chemical Education*, 88(6), 772–775.
- Kumari, M., Chilwant, N., Prajapati, A., & Kulkarni, M. S. (2015) Intelligent Shop Floor Scheduling Using Multi Agent systems. *Proceedings of the POMS 26th Annual Conference*, Washington D. C.

- Kurkovsky, A. (2013). Interdisciplinary Systems and Simulation Studies for an Innovative Undergraduate Program. Proceedings of the Emerging M&S Applications in Industry & Academia/Modeling and Humanities Symposium 2013.
- Kwon, O. B., & Lee, K. C. (2002). MACE: Multi-Agents Coordination Engine to Resolve Conflicts among Functional Units in an Enterprise. *Expert Systems with Applications*, 23(1), 9-21.
- Lakoff, G., & Johnson, M. (2008). *Metaphors We Live By*. University of Chicago Press.
- Lane, A., Meyer, B., & Mullins, J. (2012). *Simulation with Cellular: An Project Based Introduction to Programming*, Monash University eBook, Melbourne, AU.
- Loughran, J. J. (1996). *Developing Reflective Practice: Learning about Teaching and Learning through Modelling*. Briston, PA: Falmer Press.
- Law, A. M., & Kelton, W. D. (2000). *Simulation modeling and analysis (3rd Ed.)*. New York, NY: McGraw-Hill.
- Lee, R. S., & Liu, J. N. (2004). iJADE Web-Miner: An Intelligent Agent Framework for Internet Shopping. *IEEE Transactions on Knowledge and Data Engineering*, 16(4), 461-473.
- Lévy, P., & Bononno, R. (1998). *Becoming Virtual: Reality in the Digital Age*. Da Capo Press, Incorporated.
- Li, W., Zhong, N., Yao, Y., & Liu, J. (2009). An Operable Email Based Intelligent Personal Assistant. *World Wide Web*, 12(2), 125-147.
- Lin, Y., Descamps, P., Gaud, N., Hilaire, V., & Koukam, A. (2015). Multi-Agent System for Intelligent Scrum Project Management. *Integrated Computer-Aided Engineering*, 22(3), 281-296.
- Liu, A. S., Newsom, J., Schunn, C. D., & Shoop, R. (2013). Students Learn Programming Faster through Robotic Simulation. *Tech Directions*, 72(8), 16.
- Liu, A. S., Schunn, C. D., Flot, J., & Shoop, R. (2013). The Role of Physicality in Rich Programming Environments. *Computer Science Education*, 23(4), 315-331.
- Liu, C. C., Cheng, Y. B., & Huang, C. W. (2011). The Effect of Simulation Games on the Learning of Computational Problem Solving. *Computers & Education*, 57(3), 1907-1918.
- Maes, P. (1994). Agents that Reduce Work and Information Overload. *Communication of the ACM*, 37(7), 31-40.

- Maes, P. (1995). Artificial Life Meets Entertainment: Life Like Autonomous Agents. *Communications of the ACM*, 38, 108-114.
- Maes, P., Guttman, R. H., & Moukas, A. G. (1999). Agents that Buy and Sell. *Communications of the ACM*, 42(3), 81-82.
- Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The Scratch Programming Language and Environment. *ACM Transactions on Computing Education*, 10(4), 1-15.
- Mancilla, M. C., López Neri, E., & Hermenegildo Dominguez, L. A. (2019). A Cyber-Physical System Modelling Framework for an Intelligent Urban Traffic System. *Computación y Sistemas*, 23(2), 299-312.
- Mateljan, V., Širanović, Ž., & Šimović, V. (2009). Prijedlog modela za oblikovanje multimedijjskih web nastavnih sadržaja prema pedagoškoj praksi u RH. *Informatologia*, 42(1), 38-44.
- Matthews, M. R. (2000). Constructivism in Science and Mathematics Education, . In Phillips, D. C. (Ed.), *Constructivism in Education* (pp. 161-192). Chicago, IL: University of Chicago Press.
- Mayer, R. E. (2009). *Multimedia Learning* (2nd Ed.), Cambridge, UK: Cambridge University Press.
- Mc Gregor, D. W., & Cain, M. J. (2004). *An introduction to SIMUL8*. School of Mathematics & Statistics, University of Plymouth, Plymouth, UK.
- McKeachie, W., & Svinicki, M. (2013). *McKeachie's Teaching Tips*. Belmont, CA: Wadsworth, Cengage Learning.
- Michael, J. (2006). Where's the Evidence That Active Learning Works?. *Advances in Physiology Education*, 30(4), 159-167.
- Mikropoulos, T. A., & Bellou, I. (2013). Educational Robotics as Mindtools. *Themes in Science & Technology Education*, 6, 5-14.
- Min, J., Jackman, J., & Zugg, M. (2017). Visualization Aids for Abstract Concepts Towards Better Learning Outcomes. *Proceedings of the 2017 IEEE Frontiers in Education Conference* (pp. 1-5). IEEE.

- Modell, H. I. (1996). Preparing Students to Participate in an Active Learning Environment. *Advance in Physiology Education*, 270, 69-77.
- Möhring, M. (1995). Social Science Multilevel Simulation with MIMOSE. In Troitzsch, K. G, Müller, U., Gilbert, G. N., & Doran, J. E. (Eds.), *Social Science Microsimulation*, (pp. 123-137), Springer.
- Möhring, M., & Ostermann, R. (1996). MIMOSE: Eine funktionale Sprache zur Beschreibung und Simulation individuellen Verhaltens in interagierenden Populationen. Einführung in die Modellierung. Universität Koblenz-Landau: Institut für Sozialwissenschaftliche Informatik, Koblenz, Germany.
- Murphy, E. (1997). *Constructivism: From Philosophy to Practice*, Education Resource Information Center (ERIC), Retrieved from: <https://files.eric.ed.gov/fulltext/ED444966.pdf>
- Mzahm, A. M., Ahmad, M. S., & Tang, A. Y. (2013). Agents of Things (AoT): An Intelligent Operational Concept of the Internet of Things (IoT). *Proceedings of the 13th International Conference on Intelligent Systems Design and Applications* (pp. 159-164). IEEE.
- Nguyen-Duc, M., Briot, J. P., & Drogoul, A. (2003). An Application of Multi-Agent Coordination Techniques in Air Traffic Management. *Proceedings of the IEEE/WIC International Conference on Intelligent Agent Technology, IAT 2003*. (pp. 622-625). IEEE.
- Nikolai, C., & Madey, G. (2009). Tools of the Trade: A Survey of Various Agent Based Modeling Platforms. *Journal of Artificial Societies and Social Simulation*, 12(2).
- Nwana, H. S. (1996), *Software Agents: An Overview*. *The Knowledge Engineering Review*, 11(3), 205-244.
- Onken, R., & Schulte, A. (2010). *System-Ergonomic Design of Cognitive Automation: Dual-Mode Cognitive Design of Vehicle Guidance and Control Work Systems* (Vol. 235). Springer.
- Palekčić, M. (2007). *Od kurikuluma do obrazovnih standarda*, In Previšić, V. (Ed.), *Kurikulum: Teorije, metodologija, sadržaj, struktura*. Zagreb, HR: Školska knjiga.
- Pan, J. (2016). *Performance Engineering of the StarLogo Nova Execution Engine* (Doctoral dissertation). Retrieved from DSpace@MIT Digital Repository: <https://dspace.mit.edu/handle/1721.1/113161>

- Papastergiou, M. (2009). Digital Game-Based Learning in High School Computer Science Education: Impact on Educational Effectiveness and Student Motivation. *Computers & Education*, 52(1), 1-12.
- Parsons, J., & Taylor, L. (2011). Improving Student Engagement. *Current Issues in Education*, 14(1).
- Pea, R. D. (1987). Logo Programming and Problem Solving. Retrieved from CCSD HAL website: <https://telearn.archives-ouvertes.fr/hal-00190546/document>
- Perkins, D. N. (1992). Technology Meets Constructivism: Do They Make a Marriage?, In Duffy, M., & Jonassen, D. H. (Eds.), *Constructivism and the Technology of Instruction: A Conversation*, Lawrence Erlbaum Associates, (pp. 45-55).
- Piaget, J. (1974). *To Understand is to Invent*, New York, NY: Basic Books.
- Piaget, J. (1977). *The Development of Thought: Equilibrium of Cognitive Structures*. New York, NY: Viking Press.
- Pidd, M. (2004). *Computer Simulation in Management Science (5th Ed.)*. Chichester, UK: John Wiley & Sons.
- Pidd, M. (2009). *Tools for Thinking: Modelling in Management Science (3rd Ed.)*. Chichester, UK: John Wiley & Sons.
- Pietroszek, K. (2007). Providing Language Instructor with Artificial Intelligence Assistant. *International Journal of Emerging Technologies in Learning*, 2(4), 61-65.
- Pinder, J.P. (2013). An Active Learning Exercise for Introducing Agent-Based Modeling. *Decision, Science Journal of Innovative Education*, 11(3), 221–232.
- Prensky, M. (2001). Digital Natives, Digital Immigrants, Part 1. *On the Horizon*, 9(5), 1-6.
- Purković, D. (2016). *Elementi kontekstualnog pristupa učenju i poučavanju kao čimbenici uspješnosti nastave tehničke kulture*, (Doktorski rad). Sveučilište u Splitu, Prirodoslovno-matematički fakultet.
- Rady, H. A. (2011). Multi-Agent System for Negotiation in a Collaborative Supply Chain Management. *International Journal of Video & Image Processing and Network Security*, 11(5).
- Railsback, S. F., Lytinen, S. L., & Jackson, S. K. (2006). Agent-Based Simulation Platforms: Review and Development Recommendations. *Simulation*, 82(9), 609-623.

- Ramaley, J. A., & Zia, L. (2005). The Real vs the Possible: Closing the Gaps in Engagement and Learning. In Oblinger, D., & Oblinger, J. (Eds.), *Educating the Net Generation* (pp. 8.1-8.21), EDUCAUSE.
- Repenning, A., & Sumner, T. (1995). AgentSheets: A Medium for Creating Domain-Oriented Visual Languages. *Computer*, 28(3), 17-25.
- Repenning, A. (2000). AgentSheets: An Interactive Simulation Environment with End-User Programmable Agents. In *Proceedings of the Interaction 2000 Conference*, (pp. 1-8), Tokyo, Japan.
- Resnick, L. B. (1983). Mathematics and Science Learning: A New Conception, *Science*, 220(4596), 477-478.
- Resnick, M., Maloney, J., Monroy-Hernández, A., Rusk, N., Eastmond, E., Brennan, K., Millner, A., Rosenbaum, E., Silver, J., Silverman, B., & Kafai, Y. (2009). Scratch: Programming for All. *Communications of the ACM*, 52(11), 60-67.
- Rezat, S., & Sträßer, R. (2012). From the Didactical Triangle to the Socio-Didactical Tetrahedron: Artifacts as Fundamental Constituents of the Didactical Situation. *ZDM - The International Journal on Mathematics Education*, 44(5), 641-651.
- Richardson, V. (1997). *Constructivist Teaching and Teacher Education: Theory and Practice*. In Richardson, V. (Ed.), *Constructivist Teacher Education: Building New Understandings*, (3-14), Washington, DC: Falmer Press.
- Richey, R. C. (2008). Reflections on the 2008 AECT Definitions of the Field. *TechTrends*, 52, 24-25.
- Richiardi, M. G., & Richardson, R. E. (2017a). Agent-Based Computational Demography and Microsimulation using JAS-mine. In Grow, A., & van Bavel, J. (Eds.) *Agent-Based Modelling in Population Studies: Concepts, Methods, and Applications* (pp. 75-112). Springer.
- Richiardi, M. G., & Richardson, R. E. (2017b). JAS-mine: A New Platform for Microsimulation and Agent-Based Modelling. *International Journal of Microsimulation*, 10(1), 106-134.
- Robinson, S. (2004). *Simulation: The Practice of Model Development and Use*. Chichester, UK: John Wiley & Sons.
- Roblek, V., Meško, M., & Krapež, A. (2016). A Complex View of Industry 4.0. *Sage Open*, 6(2), 2158244016653987.

- Rosenschein J. S., & Genesereth, M. R. (1985). Deals among Rational Agents. Proceedings of the 9th International Joint Conference on Artificial Intelligence (pp. 91–99).
- Russell, S. J., & Norvig, P. (2016). *Artificial Intelligence: A Modern Approach*, (3rd ed.). Harlow, UK: Pearson Education Limited.
- Ruthven, K. (2012). The Didactical Tetrahedron as a Heuristic for Analyzing the Incorporation of Digital Technologies into Classroom Practice in Support of Investigative Approaches to Teaching Mathematics, *ZDM - The International Journal on Mathematics Education*, 44(5), 627-640.
- Rutten, N., van Joolingen, W. R., & van der Veen, J. T. (2012). The Learning Effects of Computer Simulations in Science Education. *Computers & Education*, 58(1), 136-153.
- Saber, B., & Okba, K. (2008). Intelligent agents for a semantic mediation of information systems. Proceedings of the 7th Computer Information Systems and Industrial Management Applications (pp. 237-238). IEEE.
- Sakellariou, I., Kefalas, P., & Stamatopoulou, I. (2008). Teaching Intelligent Agents using NetLogo. *ACM-IFIP IEEIII*, 209-221.
- Salomon, G., & Perkins, D. (1998). Individual and Social Aspects of Learning, In: P. Pearson, P., Iran-Nejad, A. (Eds.), *Review of Research in Education* 23, (pp. 1-24), Washington, DC: American Educational Research Association.
- Salamon, T. (2011). *Design of Agent-Based Models: Developing Computer Simulations for a Better Understanding of Social Processes*. Repin, CZ: Eva & Tomas Bruckner Publishing.
- Seghroucheni, Y. Z., Al Achhab, M., & El Mohajir, B. E. (2014). Revisiting the Didactic Triangle in the Case of an Adaptive Learning System. *International Journal of Engineering Pedagogy*, 4(4), 27-32.
- Selinger, M., Sepulveda, A., & Buchan, J. (2013). *Education and the Internet of Everything: How Ubiquitous Connectedness can Help Transform Pedagogy*. San Jose, CA: Cisco.
- Serenko, A., & Detlor, B. (2002). *Agent Toolkits: A General Overview of the Market and an Assessment of Instructor Satisfaction with Utilizing Toolkits in the Classroom* (Working paper No. 445). School of Business, McMaster University, Hamilton, CA.
- Shen, W., & Norrie, D. (1997). Facilitators, Mediators or Autonomous Agents. Proceedings of the 2nd International Workshop on CSCW in Design, Bangkok, Thailand (pp. 119-124).

- Shiflet, A.B., Shiflet, G.W., & Sanders Jr, W.E. (2013). Undergraduate Module on Computational Modeling: Introducing Modeling the Cane Toad Invasion. *Procedia Computer Science*, 18, 1429–1435.
- Shoham, Y. (1993). Agent-Oriented Programming, *Artificial Intelligence*, 60(1), 51-92.
- Shanbhag, S., Nair, S., Nai, N., & Shaikh , B. (2016). Intelligent Shopping Agent, *International Journal of Computer Science and Engineering*, 4(3), 142-145.
- Shrouf, F., Ordieres, J., & Miragliotta, G. (2014). Smart Factories in Industry 4.0: A Review of the Concept and of Energy Management Approached in Production Based on the Internet of Things Paradigm. *Proceedings of the 2014 IEEE International Conference on Industrial Engineering and Engineering Management* (pp. 697-701). IEEE.
- Sklar, E. (2007). NetLogo, a Multi-Agent Simulation Environment. *Artificial Life*, 13(3), 303–311
- Smith, J. K. (1983). Quantitative versus Qualitative Research: An Attempt to Clarify the Issue. *Educational Researcher*, 12(3), 6-13.
- Sokolowski, J. A., & Banks, C. M. (Eds.). (2009). *Principles of Modeling and Simulation: A Multidisciplinary Approach*. Chichester, UK: John Wiley & Sons.
- Stanley, T. (1997). Intelligent Searching Agents on the Web, *ARIADNE Web Magazine for Information Professionals*, Retrieved from: <http://www.ariadne.ac.uk/issue/7/search-engines/>
- Staton, A. Q. (1992). Teacher and Student Concern and Classroom Power and Control. In Richmond, V. P. & McCroskey, J. C. (Eds.), *Power in Classroom: Communication, Control, and Concern*, Routledge.
- Steels, L. (1995). Building Agents out of Autonomous Behavior Systems. In Steels, L., & Brooks, R. (Eds.), *The Artificial Life Route to Artificial Intelligence: Building Embodied, Situated Agents* (pp. 83-121). Hillsdale, NJ: Lawrence Erlbaum Associates.
- Stieff, M., & Wilensky, U. (2003). Connected Chemistry: Incorporating Interactive Simulations into the Chemistry Classroom. *Journal of Science Education and Technology*, 12(3), 285–302.
- Suzuki, A. (2017). Improving the Ability of StarLogo Nova Users to Save and Organize their Work (Doctoral dissertation). Retrieved from DSpace@MIT Digital Repository: <https://dspace.mit.edu/handle/1721.1/113100>
- Šperka, R. (2014, April). A Simulation Application for Educational Purposes. In *DIVAI 2014*.

- Taillandier, F., & Adam, C. (2018). Games Ready to Use: A Serious Game for Teaching Natural Risk Management. *Simulation & Gaming*, 49(4), 441-470.
- Taylor, P. (1996). Mythmaking and Mythbreaking in the Mathematics Classroom, In *Educational Studies in Mathematics*, Vol. 31, (pp 151-173).
- Taylor, P. (1998). Constructivism: Value Added. In: Fraser, B. & Tobin, K. (Eds.), *The International Handbook of Science Education*, Dordrecht, The Netherlands: Kluwer Academic.
- Terna, P. (2001). Creating Artificial Worlds: A Note on Sugarscape and Two Comments. *Journal of Artificial Societies and Social Simulation*, 4(2), 9.
- Tisue, S., & Wilensky, U. (2004a). Netlogo: A Simple Environment for Modeling Complexity. In *Proceedings of International Conference on Complex Systems* Vol. 21, (pp. 16-21).
- Tisue, S., & Wilensky, U. (2004b). NetLogo: Design and Implementation of a Multi-Agent Modeling Environment. In *Proceedings of the Agent 2004 Conference on Social Dynamics: Interaction, Reflexivity and Emergence* (pp. 7-9).
- Tobias, R., & Hofmann, C. (2004). Evaluation of Free Java-Libraries for Social-Scientific Agent Based Simulation. *Journal of Artificial Societies and Social Simulation*, 7(1).
- Turing, A. M. (1950). Computing Machinery and Intelligence. *Mind*, 49, 433-460.
- Turkle, S., & Papert, S. (1992). Epistemological Pluralism and the Revaluation of the Concrete. *Journal of Mathematical Behavior*, 11(1), 3-33.
- van Joolingen, W. R., Bollen, L., & Leenaars, F. (2012). SimSketch & GearSketch: Sketch-Based Modelling for Early Science Education, *Computer-Supported Inquiry Learning Conference*, Bochum, Germany.
- van Wieringen, R. N. (2019). An Agent Based Approach for Train Traffic Control (Master's thesis). Faculty of Science, Utrecht University.
- Vidal, J. M., & Buhler, P. (2002). Using RoboCup to Teach Multiagent Systems and the Distributed Mindset. In *ACM SIGCSE Bulletin*, 34(1), 3-7. ACM.
- Vidulin-Orbanić, S. (2007). Društvo koje uči: Povijesno-društveni aspekti obrazovanja. *Metodički obzori*, 2(2007) 1(3), 57-71.
- Vilić, D. (2014). Uloga znanja i obrazovanja u savremenom društvu. *Politeia - Naučni časopis Fakulteta političkih nauka u Banjoj Luci za društvena pitanja*, 4(8), 389-404.

- von Glasersfeld, E. (1983). Learning as a Constructive Activity, In Bergeron, J. C., & Herscovics, N. (Eds.). Proceedings of the Fifth Annual Meeting of North American Group for Psychology and Mathematics Education, Montreal, Canada, (Vol. 1, pp. 41-69).
- von Glasersfeld, E. (1998). Why Constructivism Must be Radical. In Laroche, M., Bednarz, N., & Garrison J. (Eds.), Constructivism and Education, (23-28), Cambridge, UK: Cambridge University Press.
- von Glasersfeld, E. (2001). Radical Constructivism and Teaching. Prospects, 31 (2), 161-173.
- von Glasersfeld, E. (2008). An Introduction to Radical Constructivism. Antimatters, 2(3), 5-20.
- Vygotsky, L. S. (1978). Mind in Society, Harvard University Press.
- Walker-Roberts, S., & Hammoudeh, M. (2018). Artificial Intelligence Agents as Mediators of Trustless Security Systems and Distributed Computing Applications. In Guide to Vulnerability Analysis for Computer Networks and Systems (pp. 131-155). Springer.
- Wang, S., Wan, J., Zhang, D., Li, D., & Zhang, C. (2016). Towards Smart Factory for Industry 4.0: A Self-Organized Multi-Agent System with Big Data Based Feedback and Coordination. Computer Networks, 101, 158-168.
- Webster, J., & Watson, R. T. (2002). Analyzing the Past to Prepare for the Future: Writing a Literature Review. MIS Quarterly, 26(2), xiii-xxiii.
- Wellman, M., Wurman, P. R., & Walsh, W. E. (1998). The Michigan Internet AuctionBot: A Configurable Auction Server for Human and Software Agents. Proceedings of the 2nd International Conference on Autonomous Agents (pp. 301-308).
- White, J. E. (1994), Telescript Technology: The Foundation for the Electronic Marketplace (White Paper). General Magic Inc..
- Wilensky, U. (1997a). NetLogo Ants Model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. Retrieved from: <http://ccl.northwestern.edu/netlogo/models/Ants>
- Wilensky, U. (1997b). NetLogo Fire Model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. Retrieved from: <http://ccl.northwestern.edu/netlogo/models/Fire>

- Wilensky, U. (1997c). NetLogo Segregation Model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. Retrieved from: <http://ccl.northwestern.edu/netlogo/models/Segregation>
- Wilensky, U. (1998). NetLogo Virus Model. Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. Retrieved from: <http://ccl.northwestern.edu/netlogo/models/Virus>
- Wooldridge, M. J., & Jennings, N. R. (1994), Agent Theories, Architectures, and Languages: A Survey. International Workshop on Agent Theories, Architectures, and Languages (pp. 1-39). Berlin - Heidelberg: Springer.
- Wooldridge, M. J., & Jennings, N. R. (1995), Intelligent Agents: Theory and Practice. The Knowledge Engineering Review, 10(2), 115-152.
- Wu, S., Ghenniwa, H., Zhang, Y., & Shen, W. (2006). Personal Assistant Agents for Collaborative Design Environments. Computers in Industry, 57(8-9), 732-739.
- Xiao, B., & Benbasat, I. (2007). E-Commerce Product Recommendation Agents: Use, Characteristics, and Impact. MIS quarterly, 31(1), 137-209.
- Yan, Y., Kuphal, T., & Bode, J. (2000). Application of Multiagent Systems in Project Management. International Journal of Production Economics, 68(2), 185-197.
- Yu, H., Shen, Z., & Leung, C. (2013). From Internet of Things to Internet of Agents. Proceedings of the IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing (pp. 1054-1057). IEEE.
- Zierer, K., & Seel, N. M. (2012). General Didactics and Instructional Design: Eyes Like Twins A Transatlantic Dialogue about Similarities and Differences, about the Past and the Future of Two Sciences of Learning and Teaching. SpringerPlus, 1(1), 1-22.
- Zhong, R. Y., Xu, X., Klotz, E., & Newman, S. T. (2017). Intelligent Manufacturing in the Context of Industry 4.0: A Review. Engineering, 3(5), 616-630.
- Zhu, S., Xie, F., & Levinson, D. (2011). Enhancing Transportation Education through Online Simulation using an Agent-Based Demand and Assignment Model. Journal of Professional Issues in Engineering Education and Practice, 137(1), 38-45.

PRILOZI

Prilog 1: NetLogo kod za primjer 1.1

(jednostavni reaktivni agent koji pokreće autonomno vozilo)

```
to setup
  clear-all
  create-turtles 5                ;; kreira se 5 kornjaca (vozila)
  ask turtles
  [
    set color blue                ;; vozila se boje plavom bojom
    set shape "car"
    set size 1
    fd 50 ]
  ask turtles
  [
    set heading 90                ;; vozila se okreću udesno
    ask patch 6 6 [ set pcolor red ] ;; postavlja se crveno svjetlo semafora
    reset-ticks
  ]
end

to go
  ask turtles [ walk ]            ;; pokreće se funkcija walk za vozila
  ask patch 6 6 [ stoplight ]    ;; pokreće se funkcija stoplight za semafor
  tick
end

to walk                          ;; funkcija vozila
  if ([pcolor] of patch 6 6 = green) ;; ako je na semaforu zeleno vozila se kreću
  [fd 1]
end

to stoplight                     ;; funkcija semafora
  ifelse ticks mod 100 > 50
  [ ask patch 6 6 [ set pcolor green ] ]
  [ ask patch 6 6 [ set pcolor red ] ] ;; svakih 50 tickova mijenja se svjetlo
end
```

Prilog 2: NetLogo kod za primjer 1.2

(jednostavni reaktivni agent koji pokreće rover na Marsu)

```
turtles-own [ samples ]

to setup
  clear-all
  clear-output
  create-turtles 1
  [ set samples [] ]           ;; kreira se 1 kornjaca (rover)

  ask turtles
  [
    setxy -8 -6               ;; rover se postavlja u donju lijevu celiju
    set heading 90 ]         ;; i okreće udesno
  ask n-of 5 patches [ set pcolor orange ]
  ask n-of 5 patches [ set pcolor red ]
  ask n-of 5 patches [ set pcolor blue ]
                                   ;; slučajno se u okruženje postavljaju uzorci

  reset-ticks
end

to go
  if ([xcor] of turtle 0 = 8) and ([ycor] of turtle 0 = 6)
  [ stop ]
  ask turtles [ walk ]         ;; pokreće se funkcija walk za rover
  tick
end

to walk                           ;; funkcija rovera
  if ([pcolor] of patch-here = orange)
  [ set samples lput "Orange" samples
    show samples ]
  if ([pcolor] of patch-here = red)
  [ set samples lput "Red" samples
    show samples ]
  if ([pcolor] of patch-here = blue)
  [ set samples lput "Blue" samples
    show samples ]           ;; ukoliko rover nađje na uzorak kupi ga
  ifelse (xcor = 8) and (heading = 90)
  [ ifelse (ycor = 8)
    [ show "Done" ]
    [ set heading 0
      fd 1
      set heading 270 ]]
  [ ifelse (xcor = -8) and (heading = 270)
    [ set heading 0
      fd 1
      set heading 90 ]
    [ fd 1 ] ]
end
```

Prilog 3: NetLogo kod za primjer 1.3

(reaktivni agent sa stanjima koji pokreće rover na Marsu)

```
turtles-own [ samples ]

to setup
  clear-all
  clear-output
  create-turtles 1
  [ set samples [] ]           ;; kreira se 1 kornjaca (rover)
  ask turtles
  [
    setxy -8 -6               ;; rover se postavlja u donju lijevu celiju
    set heading 90 ]         ;; i okreće udesno
  ask n-of 5 patches [ set pcolor orange ]
  ask n-of 5 patches [ set pcolor red ]
  ask n-of 5 patches [ set pcolor blue ]
                                     ;; slučajno se u okruženje postavljaju uzorci

  reset-ticks
end

to go
  if ([xcor] of turtle 0 = 8) and ([ycor] of turtle 0 = 6)
  [ stop ]
  ask turtles [ walk ]         ;; pokreće se funkcija walk za rover
  tick
end

to walk                               ;; funkcija rovera
  if ([pcolor] of patch-here = orange)
  [ if not (member? "Orange" samples )
    [ set samples lput "Orange" samples
      show samples ] ]
  if ([pcolor] of patch-here = red)
  [ if not (member? "Red" samples )
    [ set samples lput "Red" samples
      show samples ] ]
  if ([pcolor] of patch-here = blue)
  [ if not (member? "Blue" samples )
    [ set samples lput "Blue" samples
      show samples ] ]       ;; ukoliko rover naiđe na novi uzorak kupi ga
  ifelse (xcor = 8) and (heading = 90)
  [ set heading 0
    fd 1
    set heading 270 ]
  [ ifelse (xcor = -8) and (heading = 270)
  [ set heading 0
    fd 1
    set heading 90 ]
  [ fd 1 ] ]
end
```

Prilog 4: NetLogo kod za primjer 1.4

(reaktivni agent sa stanjima koji pokreće pametni usisavač)

```
turtles-own [ model
  pos-xy
  x
  y ]

to setup
  clear-all
  clear-output

  ask n-of 50 patches [ set pcolor grey ]      ;; 50 celija se boji sivo (prljavstina)
  create-turtles 1                             ;; kreira se 1 kornjaca (usisivac)
  [ set model []
    set x -8
    set y -6
    loop                                       ;; kreira se model okruzenja
    [ ifelse ([pcolor] of patch x y = grey)
      [ set model lput 1 model ]
      [ set model lput 0 model ]
      set x (x + 1)
      if (x = 9)
      [ set x -8
        set y (y + 1) ]
      if (y = 7)
      [ show model
        stop ]
    ]
  ]
  ask turtles
  [ setxy -8 -6                               ;; usisivac se postavlja u donju lijevu celiju
    set heading 90 ]                           ;; i okrece udesno

  ]
  reset-ticks
end

to go
  ask turtles [ walk ]                         ;; pokrece se funkcija walk za usisivac
  tick
end

to walk                                       ;; funkcija usisivaca
  set pos-xy (position 1 model)
  show pos-xy
  ifelse (is-number? pos-xy)                 ;; dok god postoji prljavstina u prostoriji
  [
    set x (-8 + (pos-xy mod 17))
    set y (-6 + ((pos-xy - (pos-xy mod 17)) / 17))
    show x show y
  ]
end
```



```
facexy x y                ;; usisivac se usmjerava prema njoj
fd (distancexy x y)
set model replace-item pos-xy model 0
set pcolor black          ;; cisti se prljavstina
show model ]
[ stop ]
end
```

Prilog 5: NetLogo kod za primjer 1.5

(agent usmjeren prema cilju koji pokreće pametna kolica u trgovini)

```
turtles-own [ shopping-list
             shopping-cart ]

to setup
  clear-all
  clear-output
  create-turtles 1
  [
    set shopping-list ["Green" "Orange" "Red" "Blue"]
    set shopping-cart []
  ]
  ;; kreira se 1 kornjaca (kolica za kupovinu)

  ask turtles
  [
    setxy -8 -6      ;; kolica se postavljaju u donju lijevu celiju
    set heading 90 ] ;; i okrecu udesno
  ask n-of 5 patches [ set pcolor orange ]
  ask n-of 5 patches [ set pcolor red ]
  ask n-of 5 patches [ set pcolor blue ]
  ask n-of 5 patches [ set pcolor green ]
  ask n-of 5 patches [ set pcolor yellow ]
  ask n-of 5 patches [ set pcolor grey ]
  ;; slucajno se postavljaju proizvodi u trgovini

  reset-ticks
end

to go
  if ([xcor] of turtle 0 = 8) and ([ycor] of turtle 0 = 6)
  [ stop ]
  ask turtles [ walk ] ;; pokrece se funkcija walk za kolica
  tick
end

to walk
  ;; funkcija kolica
  ;; provjerava se je li proizvod u listi za kupovinu i ako je stavlja se u kolica
  if ([pcolor] of patch-here = orange)
  [
    if (member? "Orange" shopping-list) and not (member? "Orange" shopping-cart )
    [
      set shopping-cart lput "Orange" shopping-cart
      set pcolor black
      show shopping-cart ] ]
  if ([pcolor] of patch-here = red)
  [
    if (member? "Red" shopping-list) and not (member? "Red" shopping-cart )
    [
      set shopping-cart lput "Red" shopping-cart
      set pcolor black
```

```

    show shopping-cart ] ]
if ([pcolor] of patch-here = blue)
[
  if (member? "Blue" shopping-list) and not (member? "Blue" shopping-cart )
  [
    set shopping-cart lput "Blue" shopping-cart
    set pcolor black
    show shopping-cart ] ]
if ([pcolor] of patch-here = green)
[
  if (member? "Green" shopping-list) and not (member? "Green" shopping-cart )
  [
    set shopping-cart lput "Green" shopping-cart
    set pcolor black
    show shopping-cart ] ]
if ([pcolor] of patch-here = yellow)
[
  if (member? "Yellow" shopping-list) and not (member? "Yellow" shopping-cart )
  [
    set shopping-cart lput "Yellow" shopping-cart
    set pcolor black
    show shopping-cart ] ]
if ([pcolor] of patch-here = grey)
[
  if (member? "Grey" shopping-list) and not (member? "Grey" shopping-cart )
  [
    set shopping-cart lput "Grey" shopping-cart
    set pcolor black
    show shopping-cart ] ]
if(xcor = -8) and (ycor = -6)
[
  show shopping-list ]
ifelse (xcor = 8) and (heading = 90)
[
  set heading 0
  fd 1
  set heading 270 ]
[
  ifelse (xcor = -8) and (heading = 270)
  [
    set heading 0
    fd 1
    set heading 90 ]
  [ fd 1 ] ]
end

```

Prilog 6: NetLogo kod za primjer 1.6

(agent usmjeren prema uspješnosti koji određuje optimalnu rutu vozila)

```
breed [auta auto]
breed [zidovi zid]
breed [mete meta]           ;; definiranje vozila i prepreka

globals [ bz pz bm pm cilj
           cestarine brzine kraj]

to setup
  clear-all

  create-auta 1              ;; kreira se 1 vozilo
  ask auta
  [ set kraj 0
    set color blue
    set shape "car"
    set size 1
    setxy -6 6               ;; vozila se postavlja u donji lijevi kut
    set heading 90           ;; i okreće udesno
    show "Dozvoljena brzina"
    set brzine [120 90 130]
    show brzine              ;; postavljaju se dozvoljene brzine
    show "Cestarina po km"
    set cestarine [1 3 2]
    show cestarine           ;; postavljaju se cestarine
    set cilj random 2
    ifelse cilj = 0          ;; bira se cilj: najbrza ili najjeftinija ruta
    [show "Najbrza ruta"]
    [show "Najjeftinija ruta"]
  ]

  set bz random 2           ;; slučajnim izborom stvara se prva prepreka
  if bz = 1
  [ create-zidovi bz        ;; ako je prepreka stvorena
    set pz random 3         ;; slučajnim izborom bira se pozicija
    ask zidovi
    [ set color red
      set shape "tile brick"
      set size 1
      ifelse pz = 0
      [ setxy -3 1
        set brzine replace-item 0 brzine 0
        set cestarine replace-item 0 cestarine 100]
      [ ifelse pz = 1
        [ setxy 0 -2
          set brzine replace-item 1 brzine 0
          set cestarine replace-item 1 cestarine 100]
        [ setxy 4 -1
          set brzine replace-item 2 brzine 0
```

```

    set cestarine replace-item 2 cestarine 100]]
]
]

set bm random 2          ;; slucajnim izborom stvara se druga prepreka
if bm = 1
[ create-mete 1          ;; ako je prepreka stvorena
  set pm random 3        ;; slucajnim izborom bira se pozicija
  ask mete
  [ set color red
    set shape "target"
    set size 1
    set size 1
    ifelse pm = 0
    [ setxy -3 0
      set brzine replace-item 0 brzine 0
      set cestarine replace-item 0 cestarine 100]
    [ ifelse pm = 1
      [ setxy 1 -2
        set brzine replace-item 1 brzine 0
        set cestarine replace-item 1 cestarine 100]
      [ setxy 4 0
        set brzine replace-item 2 brzine 0
        set cestarine replace-item 2 cestarine 100]]
    ]
  ]

ask patches [ set pcolor grey ] ;; postavljaju se rute A, B i C
(foreach [-6 -5 -4 -3 -2 -1 0 1 2 3 4] [-5 -5 -5 -5 -5 -5 -5 -5 -5 -5]
[
  [a b] -> ask patch a b [set pcolor black]
])
(foreach [-3 -3 -3 -3 -3 -3 -3 -3 -3] [-4 -3 -2 -1 0 1 2 3 4 5]
[
  [a b] -> ask patch a b [set pcolor black]
])
(foreach [4 4 4 4 4 4 4 4 4] [-4 -3 -2 -1 0 1 2 3 4 5]
[
  [a b] -> ask patch a b [set pcolor black]
])
(foreach [-1 -1 -1 0 1 1 1 1 1 2 3 4] [-4 -3 -2 -2 -2 -1 0 1 2 2 2 2]
[
  [a b] -> ask patch a b [set pcolor black]
])
(foreach [-2 -1 0 1 2 3 5 6] [5 5 5 5 5 5 5]
[
  [a b] -> ask patch a b [set pcolor black]
])
reset-ticks
end

to go

```

```

if kraj = 0
[ ask auta [walk]]
tick
end

to walk                                ;; funkcija vozila
if xcor != -3
[fd 1]
if xcor = -3                            ;; kad dodje na racvanje
[ ifelse cilj = 0                        ;; u ovisnosti o cilju
  [ ifelse ((item 0 brzine > item 1 brzine) and (item 0 brzine > item 2 brzine))
    [show "Optimalna je ruta A" ]
  [ ifelse ((item 1 brzine > item 0 brzine) and (item 1 brzine > item 2 brzine))
    [ show "Optimalna je ruta B"]
  [ show "Optimalna je ruta C"]] ;; bira optimalnu rutu
]
[ ifelse ((item 0 cestarine < item 1 cestarine) and (item 0 cestarine <
  item 2 cestarine))
  [ show "Optimalna je ruta A" ]
[ ifelse ((item 0 cestarine < item 1 cestarine) and (item 0 cestarine <
  item 2 cestarine))
  [ show "Optimalna je ruta B" ]
  [ show "Optimalna je ruta C" ]]
]
set kraj 1
]
end

```

Prilog 7: NetLogo kod za primjer 2.1

(determinističko okruženje po kriteriju determinističnosti)

```
globals [ dirt-count ] ;; brojac "prljavih" celija okruzenja

to setup
  clear-all           ;; briše sve postavke iz prethodnih pokušaja
  set dirt-count 0    ;; postavlja brojac "prljavih" celija na 0
  create-turtles 1    ;; kreira se jedan agent (usisivac)
  [ set shape "bulldozer top" ]
  ask turtles
  [
    setxy -6 -6      ;; usisivac se postavlja u donju lijevu celiju
    set color green  ;; boji u zeleno
    set heading 90   ;; i okreće udesno
  ]
  ask n-of 15 patches ;; "prlja" se slučajnih 15 polja prostorije
  [ set pcolor grey ]
  reset-ticks
end

to go
  if (count patches with [pcolor = gray] = 0)
    ;; ako nema "prljavih" celija
  [ show "Done!"    ;; ispisuje se poruka o završenom ciscenju
    stop ]         ;; i zaustavlja se simulacija
  ask turtles
  [ clean ]       ;; poziva se funkcija kretanja usisivaca
  tick
end

to clean           ;; funkcija kretanja usisivacavozila
  if ([pcolor] of patch-here = grey)
    ;; ukoliko je naisao na prljavstinu
  [ set dirt-count dirt-count + 1
    type "Dirt: " type dirt-count print""
    ;; ispisuje se poruku
    wait 0.3           ;; usisivac ceka 0.3 sekunde
    set pcolor black ] ;; te cisti celiju prostorije (mijenja joj boju u crnu)
  ifelse (xcor = 6) and (heading = 90) ;; okretanje na desnom rubu prostorije
  [ set heading 0
    fd 1
    set heading 270 ]
  [ ifelse (xcor = -6) and (heading = 270) ;; okretanje na lijevom rubu prostorije
  [ set heading 0
    fd 1
    set heading 90]
  [ fd 1 ]]]       ;; kretanje
end
```

Prilog 8: NetLogo kod za primjer 2.2

(stohastičko okruženje po kriteriju determinističnosti)

```
globals [ dirt-probability dirt-count ]
                                ;; vjerojatnost prljanja i brojac prljavih celija

to setup
  clear-all                    ;; briše sve postavke iz prethodnih pokušaja
  set dirt-count 0              ;; postavlja brojac "prljavih" celija na 0
  create-turtles 1              ;; kreira se 1 agent (usisivac)
  [ set shape "bulldozer top" ]
  ask turtles
  [ setxy -6 -6                 ;; usisivac se postavlja u donju lijevu celiju prostorije
    set color blue              ;; boji u plavo
    set heading 90 ]           ;; i okreće udesno
  ask n-of 15 patches           ;; "prlja" se slučajnih 15 polja prostorije
  [ set pcolor grey ]
  reset-ticks
end

to go
  if (count patches with [pcolor = gray] = 0)
    ;; ako nema "prljavih" celija
  [ show "Done!"               ;; ispisuje se poruka o završenom ciscenju
    stop ]                     ;; i zaustavlja se simulacija
  ask turtles
  [ clean ]                    ;; poziva se funkcija kretanja usisivaca
  set dirt-probability random 100 ;; generira se vjerojatnost ponovnog prljanja
  if dirt-probability > 95      ;; i ako je veci od 95
  [ ask n-of 1 patches [ set pcolor gray ] ] ;; "prlja" se nasumicna celija
  tick
end

to clean ;; funkcija kretanja usisivaca
  if ([pcolor] of patch-here = grey) ;; ukoliko je naisao na prljavstinu
  [ set dirt-count dirt-count + 1
    type "Dirt: " type dirt-count print "" ;; ispisuje poruku
    wait 0.3                               ;; i ceka 0.3 sekunde
    set pcolor black ]                     ;; te cisti celiju prostorije
  if (xcor = 6) and (ycor = 6) ;; ako dodje do kraja (a prostorija nije cista)
  [ setxy -6 -6 ]                         ;; vraca se u donju lijevu celiju i cisti iz pocetka
  ifelse (xcor = 6) and (heading = 90) ;; okretanje na desnom rubu prostorije
  [ set heading 0
    fd 1
    set heading 270 ]
  [ ifelse (xcor = -6) and (heading = 270) ;; okretanje na lijevom rubu prostorije
    [ set heading 0
      fd 1
      set heading 90 ]
    [ fd 1 ] ] ;; kretanje
end
```


Prilog 9: NetLogo kod za primjer 2.3

(potpuno vidljivo okruženje po kriteriju dostupnosti informacija)

```
globals [ px py ]           ;; globalna varijable za "crtanje" ceste
to setup
  clear-all                ;; briše sve postavke iz prethodnih pokusaja
  draw-road                 ;; funkcija koja crta cestu
  create-turtles 5          ;; kreira se 5 agenata (vozila)
  ask turtles
  [ set color blue          ;; vozila se boje plavom bojom
    set shape "car"        ;; daje im se oblik automobila
    move-to one-of patches with [pcolor = gray]           ;; postavljaju se na cestu
    set heading 90 ]       ;; i okreću udesno
  ask patch 6 6
  [ set pcolor red ]       ;; pali se srveno svjetlo semafora
  reset-ticks
end
to go
  ask turtles
  [ walk ]                  ;; pokreće se funkcija kretanja vozila
  ask patches
  [ stoplight ]           ;; pokreće se funkcija semafora
  tick
end

to walk                      ;; funkcija kretanja vozila (agentska funkcija)
  if ([pcolor] of patch 6 5 = green) ;; ako je na semaforu zeleno vozila se kreću
  [ fd 1 ]
end

to stoplight                 ;; funkcija semafora (funkcija okruženja)
  ifelse ticks mod 100 > 50    ;; svakih 50 tickova mijenja se svjetlo
  [ ask patch 6 6
    [ set pcolor black ]
    ask patch 6 5
    [ set pcolor green ]]
  [ ask patch 6 6
    [ set pcolor red ]
    ask patch 6 5
    [ set pcolor black ]]
end

to draw-road                 ;; funkcija crtanja ceste
  set px -6
  set py -3
  while [py < 4]
  [ ask patch px py
    [ set pcolor gray]
    set px px + 1
    if px = 7
    [ set px -6
      set py py + 1 ]]
end
```

Prilog 10: NetLogo kod za primjer 2.4

(djelomično vidljivo okruženje po kriteriju dostupnosti informacija)

```
globals [ px py ]                ;; globalna varijable za "crtanje" ceste

breed [cars car]
breed [obstacles x]

to setup
  clear-all                    ;; briše sve postavke iz prethodnih pokusaja
  draw-road                    ;; funkcija koja crta cestu
  create-cars 5                ;; kreira se 5 agenata (vozila)
  ask cars
  [ set color blue            ;; vozila se boje plavom bojom
    set shape "car"          ;; daje im se oblik automobila
    move-to one-of patches with [pcolor = gray and
      (one-of other cars-here = nobody) ] ;; postavljaju se na cestu (siva celija)
    set heading 90 ]         ;; i okreću udesno
  create-obstacles 4          ;; kreiraju se prepreke
  ask obstacles
  [ set color black
    set shape "x"
    move-to one-of patches with [pcolor = gray and (one-of other cars-here =
      nobody) and (one-of other obstacles-here = nobody)]]
  ask patch 6 6
  [ set pcolor red ]         ;; pali se srveno svjetlo semafora
  reset-ticks
end

to go
  ask cars
  [ walk ]                    ;; pokreće se funkcija kretanja vozila
  ask patches
  [ stoplight ]              ;; pokreće se funkcija semafora
  tick
end

to walk                        ;; funkcija kretanja vozila (agentska funkcija)
  if ([pcolor] of patch 6 5 = green) ;; ako je na semaforu zeleno vozila se kreću
  [ ifelse any? obstacles-on patch-here
    [ set color red ]
    [ fd 1 ] ]
end

to stoplight                   ;; funkcija semafora (funkcija okruženja)
  ifelse ticks mod 100 > 50     ;; svakih 50 tickova mijenja se svjetlo
  [ ask patch 6 6
    [ set pcolor black ]
    ask patch 6 5
    [ set pcolor green ] ]
  [ ask patch 6 6
```

```
[ set pcolor red ]  
ask patch 6 5  
[ set pcolor black ]]  
end
```

```
to draw-road      ;; funkcija crtanja ceste  
set px -6  
set py -3  
while [py < 4]  
[ ask patch px py  
  [ set pcolor gray ]  
  set px px + 1  
  if px = 7  
  [ set px -6  
    set py py + 1 ]]  
end
```

Prilog 11: NetLogo kod za primjer 2.5

(epizodično okruženje po kriteriju epizodičnosti)

```
globals [ new-products products badx bady ]
to setup
  clear-all          ;; briše sve postavke iz prethodnih pokušaja
  set new-products 0  ;; postavlja broj dovršenih proizvoda na 0
  set products 0     ;; postavlja broj dovršenih proizvoda na 0
  set badx 6
  set bady 5         ;; koordinate za odlaganje neispravnog proizvoda
  ask patch 6 6
  [ set pcolor green ] ;; signalno svjetlo se postavlja na zelenu boju
  draw-track        ;; crta se proizvodna traka
  reset-ticks
end
to go
  ask turtles
  [ walk ]          ;; pokreće se funkcija za pomicanje proizvoda po traci
  if (products = 30) ;; nakon serije od 30 proizvoda
  [ show "End of Product Series"
    ask patch 6 6
    [ set pcolor red ] ;; signalno svjetlo se postavlja na crvenu boju
    stop ]          ;; zaustavlja se simulacija
  tick
end
to walk            ;; funkcija pomicanja proizvoda po traci
  if ycor = -3    ;; ako je proizvod na traci
  [ ifelse xcor = 6 ;; ako je proizvod dosao so kraja trake
    [ set products products + 1      ;; broji se dovršeni proizvod
      ifelse color = red             ;; neispravan proizvod se "odlaze"
      [ setxy badx bady
        ifelse bady = -1
        [ set badx badx - 1
          set bady 5 ]
        [ set bady bady - 1 ] ]
    [ die ]          ;; ispravan proizvod se uklanja sa ekrana
    [ fd 1 ]        ;; ako proizvod nije na kraju trake pomice se
  ]
end
to start-product
  create-turtles 1 ;; kreira se jedan agent (proizvod)
  [ set color one-of remove gray base-colors ;; daje mu se neka od ne-sivih boja
    set new-products new-products + 1      ;; broji se zapoceti proizvod
    type "New product No. " type new-products print ""
    set shape "crate"                      ;; daje mu se oblik sanduka
    setxy -6 -3                             ;; postavlja na pocetak trake
    set heading 90 ]                       ;; i okreće udesno
end
to draw-track
  ask patches at-points [[-6 -3] [-5 -3] [-4 -3] [-3 -3] [-2 -3] [-1 -3] [0 -3] [1 -3] [2 -3] [3 -3] [4 -3] [5 -3] [6 -3]] [ set pcolor
  grey ]
end
```

Prilog 12: NetLogo kod za primjer 2.6

(sekvencijalno okruženje po kriteriju epizodičnosti)

```
globals [ new-products products badx bady history stop-flag]
```

```
to setup
```

```
clear-all          ;; briše sve postavke iz prethodnih pokušaja
set new-products 0  ;; postavlja broj dovršenih proizvoda na 0
set products 0     ;; postavlja broj proizvoda na 0
set stop-flag FALSE ;; postavlja se zastavica za prekid na FALSE
set badx 6
set bady 5         ;; koordinate za odlaganje neispravnog proizvoda
set history []     ;; kreira se prazna list opazanja
ask patch 6 6
[ set pcolor green ] ;; signalno svjetlo se postavlja na zelenu boju
draw-track        ;; crta se proizvodna traka
reset-ticks
end
```

```
to go
```

```
ifelse not stop-flag
[ ask turtles
  [ walk ]          ;; pokrece se funkcija za pomicanje proizvoda po traci
  set history sort history          ;; sortira se povijest opazanja kako bi 0
                                   ;; (neispravni proizvodi) dosli na pocetak
  check-history     ;; funkcija provjere povijesti opazanja
  if (products = 30) ;; serija proizvoda sadrži 30 proizvoda
  [ show "Success!" ;; serija je uspjela ako je manje od 10% neispravnih
    stop ]
  tick ]
[ stop ]
end
```

```
to walk
```

```
;; funkcija pomicanja proizvoda po traci
if ycor = -3          ;; ako je proizvod na traci
[ ifelse xcor = 6     ;; ako je proizvod dosao do kraja trake
  [ set products products + 1
    ifelse color = red      ;; neispravan proizvod se "odlaze"
    [ set history lput 0 history ;; opazanje se dodaje u povijest opazanja
      setxy badx bady
      ifelse bady = -1
      [ set badx badx - 1
        set bady 5 ]
      [ set bady bady - 1 ] ]
    [ set history lput 1 history ;; opazanje se dodaje u povijest opazanja
      die ]          ;; ispravan proizvod se uklanja sa ekrana
    [ fd 1 ]        ;; ako proizvod nije na kraju trake pomice se
  ]
end
```

```
to new-product
```

```
create-turtles 1      ;; kreira se jedan agent (proizvod)
```

```

[ set color one-of remove gray base-colors ;; boji se u ne-sivu boju
  set new-products new-products + 1      ;; broji se zapoceti proizvod
  type "New product No. " type new-products print ""
  set shape "crate"                      ;; daje mu se oblik sanduka
  setxy -6 -3                            ;; postavlja na pocetak trake
  set heading 90 ]                       ;; i okrece udesno
end

to draw-track
  ask patches at-points [[-6 -3] [-5 -3] [-4 -3] [-3 -3] [-2 -3] [-1 -3] [0 -3] [1 -3] [2 -3] [3 -3] [4 -3] [5 -3] [6 -3]] [ set pcolor
  grey ]
end

to check-history
  if length history > 2                  ;; ako je u listi barem tri opazanja
  [ if (item 2 history = 0)              ;; ako je trece opazanje u sortiranoj listi 0,
    ;; tj. tri su neispravna proizvoda

    [ ask patch 6 6
      [ set pcolor red ] ;; signalno svjetlo se postavlja na crvenu boju
    ]
    show "Fail"                        ;; ukoliko je 10% neispravnih serija nije uspjela
    set stop-flag TRUE ]               ;; postavlja se zastavica zaustavljanja na TRUE
end

```

Prilog 13: NetLogo kod za primjer 3.1

(neovisnost u višeagentskom sustavu)

```
turtles-own [ samples1 samples2 ]

to setup
  clear-all
  clear-output
  create-turtles 2          ;; kreiraju se dva agenta
  [
    set samples1 []
    set samples2 []
  ]
  ask turtle 0
  [ set color brown
    setxy -8 -6           ;; kornjaca se postavlja u donju lijevu celiju
    set heading 90 ]     ;; i okrece udesno
  ask turtle 1
  [ set color green
    setxy 8 6            ;; kornjaca se postavlja u gornju desnu celiju
    set heading 270 ]   ;; i okrece ulijevo
  ask n-of 5 patches [ set pcolor orange ]
  ask n-of 5 patches [ set pcolor red ]
  ask n-of 5 patches [ set pcolor blue ]   ;; postavljaju se nasumicni uzorci
  reset-ticks
end

to go
  if ([xcor] of turtle 0 = 8) and ([ycor] of turtle 0 = 6)
  [ stop ]
  ask turtle 0 [ walk1 ]   ;; pokrece se funkcija walk1 za prvu kornjacu
  ask turtle 1 [ walk2 ]   ;; pokrece se funkcija walk2 za drugu kornjacu
  tick
end

to walk1                      ;; funkcija prve kornjace
  if ([pcolor] of patch-here = orange)
  [ set samples1 lput "Orange" samples1
    show samples1 ]
  if ([pcolor] of patch-here = red)
  [ set samples1 lput "Red" samples1
    show samples1 ]
  if ([pcolor] of patch-here = blue)
  [ set samples1 lput "Blue" samples1
    show samples1 ]
  ifelse (xcor = 8) and (heading = 90)
  [ ifelse (ycor = 8)
    [ show "Done" ]
    [ set heading 0
      fd 1
      set heading 270 ]]
end
```

```

[ ifelse (xcor = -8) and (heading = 270)
  [ set heading 0
    fd 1
    set heading 90 ]
  [ fd 1 ] ]
end

to walk2                                     ;; funkcija druge kornjace
if ([pcolor] of patch-here = orange)
  [ set samples2 lput "Orange" samples2
    show samples2 ]
if ([pcolor] of patch-here = red)
  [ set samples2 lput "Red" samples2
    show samples2 ]
if ([pcolor] of patch-here = blue)
  [ set samples2 lput "Blue" samples2
    show samples2 ]

ifelse (xcor = -8) and (heading = 270)
  [ ifelse (ycor = -6)
    [ show "Done" ]
    [ set heading 180
      fd 1
      set heading 90 ] ]
  [ ifelse (xcor = 8) and (heading = 90)
    [ set heading 180
      fd 1
      set heading 270 ]
    [ fd 1 ] ]
end

```


Prilog 14: NetLogo kod za primjer 3.2

(jednostavna suradnja u višeagentskom sustavu)

```
globals [ samples1 ]

to setup
  clear-all
  clear-output
  create-turtles 2          ;; kreiraju se 2 kornjace
  [ set samples1 [] ]
  ask turtle 0
  [ set color brown
    setxy -8 -6            ;; kornjaca se postavlja u donju lijevu celiju
    set heading 90 ]      ;; i okrece udesno
  ask turtle 1
  [ set color green
    setxy 8 6              ;; kornjaca se postavlja u gornju desnu celiju
    set heading 270 ]     ;; i okrece ulijevo
  ask n-of 5 patches [ set pcolor orange ]
  ask n-of 5 patches [ set pcolor red ]
  ask n-of 5 patches [ set pcolor blue ]   ;; postavljaju se uzorci
  reset-ticks
end

to go
  if ([xcor] of turtle 0 = 8) and ([ycor] of turtle 0 = 6)
  [ stop ]
  ask turtle 0 [ walk1 ]   ;; pokrece se funkcija walk1 za prvu kornjacu
  ask turtle 1 [ walk2 ]   ;; pokrece se funkcija walk2 za drugu kornjacu
  tick
end

to walk1                      ;; funkcija prve kornjace
  if ([pcolor] of patch-here = orange) ;; ako kornjaca naidje na uzorak
  [ if not (member? "Orange" samples1 )   ;; koji ranije nije prikupljen
    [ set samples1 lput "Orange" samples1   ;; dodaje ga u zajednicku listu
      show samples1 ] ]
  if ([pcolor] of patch-here = red)
  [ if not (member? "Red" samples1 )
    [ set samples1 lput "Red" samples1
      show samples1 ] ]
  if ([pcolor] of patch-here = blue)
  [ if not (member? "Blue" samples1 )
    [ set samples1 lput "Blue" samples1
      show samples1 ] ]
  ifelse (xcor = 8) and (heading = 90)
  [ ifelse (ycor = 8)
    [ show "Done" ]
    [ set heading 0
      fd 1
      set heading 270 ] ]
```

```

[ ifelse (xcor = -8) and (heading = 270)
  [ set heading 0
    fd 1
    set heading 90 ]
  [ fd 1 ] ]
end

```

```

to walk2 ;; funkcija druge kornjace
if ([pcolor] of patch-here = orange)
  [ if not (member? "Orange" samples1 )
    [ set samples1 lput "Orange" samples1
      show samples1 ] ]
if ([pcolor] of patch-here = red)
  [ if not (member? "Red" samples1 )
    [ set samples1 lput "Red" samples1
      show samples1 ] ]
if ([pcolor] of patch-here = blue)
  [ if not (member? "Blue" samples1 )
    [ set samples1 lput "Blue" samples1
      show samples1 ] ]
ifelse (xcor = -8) and (heading = 270)
  [ ifelse (ycor = -6)
    [ show "Done" ]
    [ set heading 180
      fd 1
      set heading 90 ]]
  [ ifelse (xcor = 8) and (heading = 90)
    [ set heading 180
      fd 1
      set heading 270 ]
    [ fd 1 ] ]
end

```

Prilog 15: NetLogo kod za primjer 3.3

(opstrukcija u višeagentskom sustavu)

```
globals [ samples1 ]

to setup
  clear-all
  clear-output
  create-turtles 2          ;; kreiraju se dvije kornjace
  [ set samples1 [] ]
  ask turtle 0
  [ set color brown
    setxy -8 -6           ;; kornjaca se postavlja u donju lijevu celiju
    set heading 90 ]     ;; i okrece udesno
  ask turtle 1
  [ set color green
    setxy 8 6            ;; kornjaca se postavlja u gornju desnu celiju
    set heading 270 ]   ;; i okrece ulijevo
  ask n-of 5 patches [ set pcolor orange ]
  ask n-of 5 patches [ set pcolor red ]
  ask n-of 5 patches [ set pcolor blue ]      ;; postavljaju se nasumicni uzorci
  reset-ticks
end

to go
  if ([xcor] of turtle 0 = 8) and ([ycor] of turtle 0 = 6)
  [ stop ]
  ask turtle 0 [ walk1 ]      ;; pokrece se funkcija walk1 za prvu kornjacu
  ask turtle 1 [ walk2 ]     ;; pokrece se funkcija walk2 za drugu kornjacu
  tick
end

to walk1                    ;; funkcija prve kornjace
  if ([pcolor] of patch-here = orange)
  [ if not (member? "Orange" samples1 )
    [ set samples1 lput "Orange" samples1
      show samples1 ] ]
  if ([pcolor] of patch-here = red)
  [ if not (member? "Red" samples1 )
    [ set samples1 lput "Red" samples1
      show samples1 ] ]
  if ([pcolor] of patch-here = blue)
  [ if not (member? "Blue" samples1 )
    [ set samples1 lput "Blue" samples1
      show samples1 ] ]
  ifelse (xcor = 8) and (heading = 90)
  [ ifelse (ycor = 8)
    [ show "Done" ]
    [ set heading 0
      fd 1
      set heading 270 ] ]
```

```

[ ifelse (xcor = -8) and (heading = 270)
  [ set heading 0
    fd 1
    set heading 90 ]
  [ fd 1 ]]
end

```

to walk2

:: funkcija druge kornjace

```

if ([pcolor] of patch-here = orange)
  [ if not (member? "Orange" samples1 )
    [ set samples1 lput "Orange" samples1
      show samples1 ] ]
if ([pcolor] of patch-here = red)
  [ if not (member? "Red" samples1 )
    [ set samples1 lput "Red" samples1
      show samples1 ] ]
if ([pcolor] of patch-here = blue)
  [ if not (member? "Blue" samples1 )
    [ set samples1 lput "Blue" samples1
      show samples1 ] ]
ifelse (xcor = -8) and (heading = 270)
  [ ifelse (ycor = -6)
    [ show "Done" ]
    [ set heading 180
      fd 1
      set heading 90 ] ]
  [ ifelse (xcor = 8) and (heading = 90)
    [ set heading 180
      fd 1
      set heading 270 ]
    [ fd 1 ] ]
end

```

Prilog 16: NetLogo kod za primjer 3.4

(čisto pojedinačno natjecanje u višeagentskom sustavu)

```
turtles-own [ samples1 samples2 ]

to setup
  clear-all
  clear-output
  create-turtles 2                ;; kreiraju se dvije kornjace
  [ set samples1 []
    set samples2 []
  ]
  ask turtle 0
  [ set color brown
    setxy -8 -6                  ;; kornjaca se postavlja u donju lijevu celiju
    set heading 90 ]            ;; i okreće udesno
  ask turtle 1
  [ set color green
    setxy 8 6                    ;; kornjaca se postavlja u gornju desnu celiju
    set heading 270 ]           ;; i okreće ulijevo
  ask n-of 5 patches [ set pcolor orange ]
  ask n-of 5 patches [ set pcolor red ]
  ask n-of 5 patches [ set pcolor blue ]    ;; postavljaju se nasumični uzorci
  reset-ticks
end

to go
  if ([xcor] of turtle 0 = 8) and ([ycor] of turtle 0 = 6)
  [ stop ]
  if (length [samples1] of turtle 0 = 3) or (length [samples2] of turtle 1 = 3)
  [ ifelse (length [samples1] of turtle 0 = 3)
    [ ask turtle 0 [win]                ;; uzorke je prije prikupio prvi agent
      ask turtle 1 [lose] ]
    [ ask turtle 1 [win]                ;; uzorke je prije prikupio drugi agent
      ask turtle 0 [lose] ]
    stop ]
  ask turtle 0 [ walk1 ]                ;; pokreće se funkcija walk1 za prvu kornjacu
  ask turtle 1 [ walk2 ]                ;; pokreće se funkcija walk2 za drugu kornjacu
  tick
end

to walk1                                ;; funkcija prve kornjace
  if ([pcolor] of patch-here = orange) ;; ako naidje na uzorak
  [ if not (member? "Orange" samples1 ) ;; koji nije prikupila
    [ set samples1 lput "Orange" samples1 ;; dodaje ga u listu
      show samples1 ]
    set pcolor black ]
  if ([pcolor] of patch-here = red)
  [ if not (member? "Red" samples1 )
    [ set samples1 lput "Red" samples1
      show samples1 ]
  ]
end
```

```

    set pcolor black ]
if ([pcolor] of patch-here = blue)
  [ if not (member? "Blue" samples1 )
    [ set samples1 lput "Blue" samples1
      show samples1 ]
    set pcolor black ]
ifelse (xcor = 8) and (heading = 90)
  [ ifelse (ycor = 8)
    [ show "Done" ]
    [ set heading 0
      fd 1
      set heading 270 ]]
  [ ifelse (xcor = -8) and (heading = 270)
    [ set heading 0
      fd 1
      set heading 90 ]
    [ fd 1 ]]
end
to walk2
;; funkcija druge kornjace
if ([pcolor] of patch-here = orange)
  [ if not (member? "Orange" samples2 )
    [ set samples2 lput "Orange" samples2
      show samples2 ]
    set pcolor black ]
if ([pcolor] of patch-here = red)
  [ if not (member? "Red" samples2 )
    [ set samples2 lput "Red" samples2
      show samples2 ]
    set pcolor black ]
if ([pcolor] of patch-here = blue)
  [ if not (member? "Blue" samples2 )
    [ set samples2 lput "Blue" samples2
      show samples2 ]
    set pcolor black ]
ifelse (xcor = -8) and (heading = 270)
  [ ifelse (ycor = -6)
    [ show "Done" ]
    [ set heading 180
      fd 1
      set heading 90 ]]
  [ ifelse (xcor = 8) and (heading = 90)
    [ set heading 180
      fd 1
      set heading 270 ]
    [ fd 1 ]]
end
to win
  show ":-)"
end
to lose
  show ":-("
end

```

Prilog 17: Inicijalna provjera znanja

Agentska paradigma

INICIJALNA PROVJERA ZNANJA

1. Što je to agent? (15 bodova)

2. Koji se dijelovi agenta koriste za opažanje i djelovanje na okruženje? (5 bodova)

- a) Senzori i aktuatori
- b) Senzori
- c) Perceptori
- d) Ništa od navedenog

3. Ponašanje agenata najbolje se može opisati pomoću ... (5 bodova)

- a) Slijeda opažanja
- b) Agentske funkcije
- c) Senzora i aktuatora
- d) Okruženja u kojem agent djeluje

4. Što se podrazumijeva pod sekvencom opažanja agenta? (5 bodova)

- a) Koristi se za opažanje okruženja
- b) Potpuna povijest aktuatora agenta
- c) Potpuna povijest opažanja okruženja
- d) Ništa od navedenog

5. Koju sposobnost agent koristi za poboljšanje svojih preformansi? (5 bodova)

- a) Opažanje
- b) Učenje
- c) Promatranje
- d) Ništa od navedenog

6. Što je to okruženje agenta? (15 bodova)

7. Što označava kratica PEAS u vezi okruženja agenta? (5 bodova)

- a) Peer, Environment, Actuators, Sense
- b) Perceiving, Environment, Actuators, Sensors
- c) Performance, Environment, Actuators, Sensors
- d) Ništa od navedenog

8. Koje vrste okruženja, obzirom na promotrivost, su prisutne u umjetnoj inteligenciji? (5 bodova)

- a) Djelomično promotriva
- b) Potpuno promotriva
- c) Okruženja za učenje
- d) Djelomično i potpuno promotriva

Prilog 18: Provjera znanja nakon NJ1 (prvi dio istraživanja)

Definicija, osnovna svojstva i vrste agenata

PROVJERA ZNANJA

1. Koje su razlike između jednostavnog reaktivnog agenta i reaktivnog agenta sa stanjima (3 boda)

2. Agent upravlja roverom na Marsu čiji je zadatak prikupljanje uzoraka pronađenih stijena. Kada senzori agenta opaze da je na rover naišao na stijenu prikuplja se uzorak stijene. Ukoliko na poziciji rovera nema stijene, rover nastavlja kretanje. O kojoj se vrsti agenta radi? (5 bodova)

3. Što bi se kod agenta iz zadatka 3. trebalo izmijeniti kako bi postao usmjeren cilju, tj. najbržem prikupljanju svih uzoraka (7 bodova)

4. Agent upravlja pametnim kolicima u trgovini i obavlja kupovinu u skladu sa popisom za kupovinu. Kada senzori agenta opaze da su kolica naišla na artikl u trgovini, agent provjerava nalazi li se taj artikl na popisu i u tom slučaju stavlja ga u kolica. Ukoliko na poziciji kolica nema artikla, ili se nalazi artikl koji nije na popisu, kolica nastavljaju kretanje. O kojoj se vrsti agenta radi? (5 bodova)

5. Što bi se kod agenta iz zadatka 5. trebalo izmijeniti kako bi se postao reaktivni agent sa stanjima (7 bodova)

6. Što je to agent koji uči i koje su njegove komponente? (3 boda)

Prilog 19: Provjera znanja nakon NJ2 (prvi dio istraživanja)

Definicija, osnovna svojstva i vrste okruženja

PROVJERA ZNANJA

1. Nabroji kriterije klasifikacije okruženja agenata. (2 boda)

2. Okruženje je pravokutna prostorija i u njoj usisivač čiji senzori imaju mogućnost otkrivanja prašine. U prostoriji se nalaze prepreke koje stoje na unaprijed definiranom putu usisivača. Klasificiraj okruženje po kriterijima determinističnosti, osmotrivosti i epizodičnosti. (4 boda)

3. Što bi se u okruženju iz zadatka 2. trebalo izmijeniti kako bi se izmijenila njegova klasifikacija po kriteriju determinističnosti? (6 bodova)

4. Okruženje je proizvodni pogon i u njemu agent koji registrira neispravnost proizvoda koji se kreće po proizvodnoj traci. U slučaju da broj neispravnih proizvoda dostigne predviđeni postotak, proizvodnja se prekida. Klasificiraj okruženje po kriterijima determinističnosti, osmotrivosti i epizodičnosti. (4 boda)

5. Što bi se u okruženju iz zadatka 4. trebalo izmijeniti kako bi se izmijenila njegova klasifikacija po kriteriju determinističnosti? (6 bodova)

6. Opiši svojim riječima okruženje i agenta u njemu koje bi po kriterijima determinističnosti, osmotrivosti i epizodičnosti bilo stohastičko, potpuno osmotrivo i epizodično. (8 bodova)

Prilog 20: Provjera znanja nakon NJ3 (prvi dio istraživanja)

Interakcija agenata i okruženja i međusobna interakcija agenata u višeagentskim sustavima

PROVJERA ZNANJA

1. Koje vrste interakcije u višeagentskim sustavima, zasnovane na ciljevima i sposobnostima agenata te resursima dostupnim u okruženju razlikujemo? (2 boda)

2. Agenti upravljaju roverima na Marsu, a imaju zajednički zadatak prikupljanja uzoraka određenih stijena. Roveri se kreću unaprijed definiranim putanjama i ukoliko naiđu na stijenu odgovarajuće vrste, uzimaju njen uzorak. Jedan agent može prepoznati stijenu, dok drugi može uzeti uzorak? O kojem se tipu interakcije radi? (4 boda)

3. Što bi se u okruženju te sposobnostima i/ili ciljevima agenata iz zadatka 2. trebalo izmijeniti kako bi se radilo o opstrukciji? (6 bodova)

4. Agenti upravljaju roverima na Marsu, a svaki od njih ima zadatak prikupljanja uzoraka određenih stijena. Roveri se kreću unaprijed definiranim putanjama i ukoliko naiđu na stijenu odgovarajuće vrste, uzimaju njen uzorak. Oni se nadmeću koji od njih će prije prikupiti uzorke svih stijena koje se nalaze u okruženju. O kojem se tipu interakcije radi? (4 boda)

5. Što bi se u okruženju te sposobnostima i/ili ciljevima agenata iz zadatka 4. trebalo izmijeniti kako bi se radilo o čistom pojedinačnom natjecanju? (6 bodova)

6. Opiši svojim riječima okruženje, agente u njemu i način njihove interakcije kako bi se radilo o pojedinačnom sukobu oko resursa. (8 bodova)

Prilog 21: Provjera znanja nakon NJ4 (prvi dio istraživanja)

Područja primjene agenata

PROVJERA ZNANJA

1. U kojim vrstama sustava nailazimo na primjenu agentske paradigme (2 boda)

2. Navedi po dva primjera primjene agentske paradigme u simulacijama prirodnih, društvenih i tehničkih sustavima (6 bodova)

3. Što je to Industrija 4.0 i na koji ju način možemo povezati sa agentskom paradigmom. (6 bodova)

4. Opiši svojim riječima prirodni sustav u kojem se primjenjuje agentska paradigma, objasni te koncepte agenata, okruženja i interakcije agente u njemu. (8 bodova)

5. Opiši svojim riječima tehnički sustav iz područja Industrije 4.0 u kojem se primjenjuje agentska paradigma, te objasni koncepte agenata, okruženja i interakcije agente u njemu. (8 bodova)

Prilog 22: Provjera znanja nakon NJ2 (drugi i treći dio istraživanja)

Definicija, osnovna svojstva i vrste okruženja

PROVJERA ZNANJA

1. Nabroji kriterije klasifikacije okruženja agenata. (4 boda)

2. Okruženje je pravokutna prostorija i u njoj usisivač čiji senzori imaju mogućnost otkrivanja prašine. U prostoriji se nalaze prepreke koje stoje na unaprijed definiranom putu usisivača. Klasificiraj okruženje po kriterijima determinističnosti, osmotrivosti i epizodičnosti. (8 bodova)

3. Što bi se u okruženju iz zadatka 2. trebalo izmijeniti kako bi se izmijenila njegova klasifikacija po kriteriju determinističnosti? (10 bodova)

4. Što bi se u okruženju iz zadatka 2. trebalo izmijeniti kako bi se izmijenila njegova klasifikacija po kriteriju osmotrivosti? (10 bodova)

5. Okruženje je proizvodni pogon i u njemu agent koji registrira neispravnost proizvoda koji se kreće po proizvodnoj traci. U slučaju da broj neispravnih proizvoda dostigne predviđeni postotak, proizvodnja se prekida. Klasificiraj okruženje po kriterijima determinističnosti, osmotrivosti i epizodičnosti. (8 bodova)

6. Što bi se u okruženju iz zadatka 5. trebalo izmijeniti kako bi se izmijenila njegova klasifikacija po kriteriju determinističnosti? (10 bodova)

7. Što bi se u okruženju iz zadatka 5. trebalo izmijeniti kako bi se izmijenila njegova klasifikacija po kriteriju epizodičnosti? (10 bodova)

8. Opiši svojim riječima okruženje i agenta u njemu koje bi po kriterijima determinističnosti, osmotrivosti i epizodičnosti bilo stohastičko, potpuno osmotrivo i epizodično. (10 bodova)

Prilog 23: Anketni upitnik o stavovima i dojmovima nakon održane nastave

Stavovi i dojmovi nakon održane nastave

Napomena: Označi polje koje najviše odgovara tvom stavu o održanoj nastavi

(1 - u potpunosti se ne slažem, 2 - uglavnom se ne slažem, 3 - niti se slažem niti ne slažem, 4 - uglavnom se slažem, 5 - u potpunosti se slažem)

Nastavno osoblje		1	2	3	4	5
PNO101	Nastavno osoblje je izrazito dobro objasnilo nastavne sadržaje					
PNO102	Nastavno osoblje se potrudilo da svoje nastavne sadržaje učini zanimljivim					
PNO103	Nastavno osoblje se trudilo da razumije poteškoće na koje sam naišao					
PNO104	Nastavno osoblje pružilo mi je podršku u radu					
PNO105	Nastavno osoblje motiviralo me je da pružim svoj maksimum					
PNO106	Nastavno osoblje je uložilo dosta vremena u komentiranje mog rada					
Razvijene vještine		1	2	3	4	5
POV201	Nastava mi je pomogla da razvijem sposobnost grupnog rada					
POV202	Nastava je unaprijedila moje analitičke vještine					
POV203	Nastava je razvila moje vještine rješavanja problema					
POV204	Nastava je unaprijedila moje komunikacijske vještine					
POV205	Kao rezultat nastave osjećam samopouzdanje u neuobičajenim situacijama					
POV206	Nastava mi je pomogla da razvijem sposobnost planiranja svog rada					
Nastava		1	2	3	4	5
PZN301	Nastava mi je pružila širok pregled znanstvenog područja					
PZN302	Nastava mi je donijela samopouzdanje za istraživanje novih ideja					
PZN303	Fakultet je stimulirao moj entuzijizam za dodatno učenje					
PZN304	Naučio sam primijeniti principe iz ove nastave u drugim situacijama					
PZN305	Smatram da će mi ono što sam naučio u budućnosti koristiti					
PZN306	Korištenje novih tehnologija unaprijeđuje motivaciju za rad					
PZN307	Smatram da bi se ovakav način rada trebao više primjenjivati u nastavi					
PZN308	Zadovoljan sam ukupnom kvalitetom nastave					

Koje značajke održane nastave su vam se najviše svidjele?

Koje značajke održane nastave treba poboljšati i na koji način?

Prilog 24: Standardizirani intervju s pitanjima otvorenog tipa

Razumijevanje osnovnih koncepata agentske paradigme

1. Objasni koncept inteligentnog agenta

2. Objasni koncept okruženja inteligentnog agenta

3. Objasni koncept interakcije agenta i okruženja

ŽIVOTOPIS I POPIS JAVNO OBJAVLJENIH RADOVA

Krešimir Rakić je rođen 25. travnja 1975. godine u Zagrebu. Osnovnu školu završio je u Mostaru, a opću gimnaziju u Vrgorcu. Na Fakultetu prirodoslovno-matematičkih znanosti i odgojnih područja Sveučilišta u Splitu je 2000. godine stekao zvanje profesora matematike i informatike. Od 2001. godine do 2013. godine je zaposlen kao mlađi asistent za znanstveno područje prirodnih znanosti, polje matematika, na Strojarskom fakultetu Sveučilišta u Mostaru. Od 2013. godine do 2017. godine je zaposlen kao asistent za znanstveno područje tehničkih znanosti, polje računarstvo, grana programsko inženjerstvo, na Fakultetu strojarstva i računarstva Sveučilišta u Mostaru. Od 2017. godine zaposlen je kao viši asistent za znanstveno područje tehničkih znanosti, polje računarstvo, grana programsko inženjerstvo, na Fakultetu strojarstva, računarstva i elektrotehnike Sveučilišta u Mostaru. Od 2001. godine angažiran je i kao vanjski suradnik na Građevinskom fakultetu Sveučilišta u Mostaru i Fakultetu prirodoslovno-ministarstvu matematičkih i odgojnih znanosti Sveučilišta u Mostaru. Od 2009. godine angažiran u Poreznoj upravi pri Ministarstvu financija Federacije Bosne i Hercegovine kao član tehničkog povjerenstva za provođenje fiskalizacije u Federaciji. Od 2015. godine do 2019. godine sudjelovao je na projektu Adaptive Courseware based on Natural Language Processing (AC & NL Tutor).

ZNANSTVENI RADOVI

Izvorni znanstveni i pregledni radovi u CC časopisima

Rakić, K., Rosić, M., Boljat, I. (2020). A Survey of Agent-Based Modelling and Simulation Tools for Educational Purpose, Technical Gazette, Vol 27(3), (Article No. TV-200190517110455).

Znanstveni radovi u drugim časopisima

Rakić, K., Bandić Glavaš, M., Majstorović, V. (2013). Development of the Intelligent System for the Use of University Information System, Procedia Engineering 69 (2014) pp 402-409, 24th DAAAM International Symposium on Intelligent Manufacturing and Automation, Zadar.

Znanstveni radovi u zbornicima skupova s međunarodnom recenzijom

Stankov, S., Rosić, M.; Rakić, K. (2001). Testiranje i ocjenjivanje korištenjem kvizova u inteligentnim tutorskim sustavima, Međunarodna konferencija MIPRO, Opatija.

Majstorović, V., Rakić, K. (2001). Mathematical-Statistical Models and Simulation as Elements of Technology of Decision Making in Business Processes, The 2nd International Conference UPS, Mostar.

Majstorović, V., Rakić, K. (2004). Student Office Information System – Data Model, The 3rd International Conference UPS, Mostar.

Majstorović, V., Rakić, K., Bandić, M. (2007). Računalna podrška upravljanju projektima, The 11th International Scientific Conference CIM, Biograd na Moru.

Vrdoljak, A., Rakić, K. (2007). Role of Lesson Study in Developing Good Mathematics Teaching Practice, Zbornik radova FSR, Mostar.

Vrdoljak, A., Banjanin, M., Rakić, K. (2009). Evaluating the Effectiveness of the Implemented Models of ICT Tools in Teaching Mathematics, The 4th International Conference UPS, Mostar.

Majstorović, V., Rakić, K., Bandić Glavaš, M. (2009). Development of Software Package Based on the Mathematical Graph Theory for More Efficient Project Management, The 4th International Conference UPS, Mostar.

Majstorović, V., Bandić Glavaš, M., Rakić, K. (2009). Vision of European Future of the University of Mostar, The 4th International Conference UPS, Mostar.

Vrdoljak, A., Banjanin, M., Rakić, K. (2010) Interaktivni moduli s inteligentnim tutorskim agentima u elementarnoj nastavi matematike, 33. međunarodna konferencija MIPRO, Opatija.

Rakić, K. (2013). Korištenje zadataka objektivnog tipa za procjenu znanja učenika u inteligentnim tutorskim sustavima, 2. međunarodna znanstvena konferencija PON, Mostar.

Rakić, K. (2016) The Proposal of the Intelligent System for Generating a Objective Test Questions in Controlled Natural Language for Domain Knowledge Based on Ontology, International Conference on Smart Systems and Technologies, Osijek, 2016.

Rakić, K., Rosić, M. (2016). Educational Robotics - A Tool for Developing Skills for the 21st Century, 11th International Balkan Education and Science Conference - The Future of Education and Education for the Future, Poreč.

Rakić K., Brajković, E., Rosić, M. (2016). Edukacijska robotika u suvremenoj nastavi informatike, 3. međunarodna znanstvena konferencija PON 2016, Mostar.

Majstorović V., Rakić K. (2017). Internet of Things and Social Media: Tools of a Successful Information Organization, 28th DAAAM International Symposium on Intelligent Manufacturing and Automation 2017, Zadar.

Brajković E., Rakić K., Kraljević G. (2018) Application of Data Mining in E-learning Systems, 17th International Conference INFOTEH, Jahorina.