

Internet stvari kao višeagentni sustav

Špičko, Davorin

Master's thesis / Diplomski rad

2021

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://um.nsk.hr/um:nbn:hr:211:177670>

Rights / Prava: [Attribution-NonCommercial 3.0 Unported / Imenovanje-Nekomercijalno 3.0](#)

Download date / Datum preuzimanja: **2025-02-28**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Davorin Špičko

**INTERNET STVARI KAO VIŠEAGENTNI
SUSTAV**

DIPLOMSKI RAD

Varaždin, 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ž D I N

Davorin Špičko

Matični broj: 43529/14–IZV

Studij: Informacijsko i programsko inženjerstvo

INTERNET STVARI KAO VIŠEAGENTNI SUSTAV

DIPLOMSKI RAD

Mentor:

Izv. prof. dr. sc. Markus Schatten

Varaždin, siječanj 2021.

Davorin Špičko

Izjava o izvornosti

Izjavljujem da je moj završni/diplomski rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor/Autorica potvrdio/potvrdila prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Rad opisuje razvoj računalne tehnologije kroz povijest u smjeru interneta stvari, definira pojam interneta stvari kao i najčešće uređaje korištene za spajanje stvari na internet. Opisane su vrste senzora koji su nezaobilazan dio interneta stvari pomoću kojih uređaji očitavaju vrijednosti iz okoline u kojoj se nalaze i prikupljene vrijednosti pretvaraju u korisnu informaciju za korisnika ili omogućuju korisniku interakciju s okolinom. Nadalje, rad definira pojmove agenata i višeagentnih sustava te kako internet stvari prilagoditi u smjeru interneta agenata. Internet agenata je koncept u kojem uređaji nisu samo izvršitelji akcija korisnika već su samostalni u radu na temelju umjetne inteligencije. Agenti pomoću umjetne inteligencije i međusobne komunikacije djeluju u okviru definiranih parametara sa zadaćom ostvarenja zadanog cilja.

U praktičnom je dijelu rada korištenjem programskog jezika Python i razvojnog okruženja SPADE prikazana upotreba interneta stvari kao višeagentnog sustava u području pametne kuće. Agenti međusobnom komunikacijom i komunikacijom sa vanjskim sustavima djeluju samostalno u rješavanju problema nastalih na naponskoj mreži bez potrebe za interakcijom korisnika.

Ključne riječi: internet stvari; senzor; agent; višeagentni sustav; SPADE; Raspberry Pi

Sadržaj

Sadržaj	iii
1. Uvod	1
2. Internet stvari	4
2.1. Definicija	4
2.2. Područje primjene	6
2.3. Uređaji interneta stvari	10
2.3.1. Arduino.....	10
2.3.2. Raspberry Pi	13
2.3.3. Senzori i njihova primjena	16
2.3.3.1. GPS	16
2.3.3.2. Magnetometar	16
2.3.3.3. Senzori prisutnosti objekta	17
2.3.3.4. Pasivni infracrveni senzori.....	17
2.3.3.5. Senzori udaljenosti	17
2.3.3.6. Senzori rotacije.....	17
2.3.3.7. Senzori nagiba (engl. tilt sensors)	18
2.3.3.8. Žiroskop (engl. gyroscope).....	18
2.3.3.9. Senzori za mjerenje ubrzanja (engl. accelerometer).....	18
2.3.3.10. Senzori vibracije	19
2.3.3.11. Senzori sile.....	19
2.3.3.12. Senzori dodira	19
2.3.3.13. Zaslona na dodir.....	19
2.3.3.14. Senzori razine tekućina	20
2.3.3.15. Senzori protoka tekućina i plinova	20
2.3.3.16. Senzori tlaka plina i tekućina	20
2.3.3.17. Senzor koncentracije plina.....	20

2.3.3.18.	Senzori za mjerenje svjetlosti	21
2.3.3.19.	Senzori temperature	21
2.3.3.20.	Mikrofon.....	22
2.3.3.21.	Senzor struje (engl. current sensor).....	22
2.3.3.22.	Senzor napona (engl. voltage sensor)	22
3.	Višeagentni sustavi.....	26
3.1.	Definicija agenta i višeagentnog sustava.....	26
3.1.1.	Karakteristike agenata.....	27
3.1.2.	Okruženja agenata.....	28
3.1.3.	Klasifikacija višeagentnih sustava	29
3.2.	Postizanje dogovora.....	34
3.2.1.	Dizajn protokola za postizanje dogovora	34
3.2.2.	Aukcije	34
3.3.	Od interneta stvari (IoT) do interneta agenata (IoA)	36
3.3.1.	Upravljanje vozilima u realnom vremenu	36
3.3.2.	Upravljanje prtljagom.....	36
3.3.3.	Nadgledanje industrijske proizvodnje	36
3.3.4.	Upravljanje navodnjavanjem	37
3.3.5.	Praćenje životinja	37
3.3.6.	Upravljanje poljoprivrednim zemljištima.....	37
3.3.7.	Pametni grad – turizam	38
3.3.8.	Medicinske usluge.....	38
4.	Praktični dio	39
4.1.	Opis aplikacijske domene i problema	39
4.2.	Korištene tehnologije.....	39
4.3.	Implementacija.....	40
4.3.1.	Agenti.....	44
4.3.1.1.	Agent razvodne kutije	46
4.3.1.2.	Agent uređaja	48

4.3.1.3. Servisni agent	50
4.3.1.4. Agent aukcije.....	53
4.3.2. Komunikacija agenata	53
5. Zaključak	56
Popis literature	58
Popis slika	63
Popis tablica	65
Prilozi	66

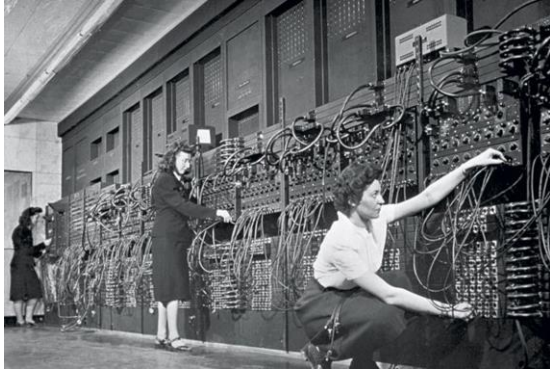
1. Uvod

Primjer korištenja tehnologije za rješavanje matematičkih problema možemo vidjeti u starom vijeku kad su ljudi osmislili abak (engl. *abacus*). Primjenom jednostavnog pomicanja kuglica pokušao se riješiti problem zbrajanja. U 17. stoljeću škotski matematičar Napier izrađuje Napierove kosti pomoću kojih je bilo moguće izračunati umnoške brojeva koristeći samo zbrajanje [1], a nastavno tome nastaju i logaritamske tablice koje je William Oughtred izradio na temelju Napierovih ideja o logaritmima [2]. U istom stoljeću Wilhelm Schickard izrađuje prvu mehaničku izvedbu kalkulatora koja je imala operacije zbrajanja, oduzimanja, množenja i dijeljenja [3]. Nekoliko godina kasnije, kako bi ocu olakšao izračun poreza, Blaise Pascal je izradio mehanički stroj koji je relativno brzo mogao zbrajati i oduzimati velike brojeve [4]. U 19. stoljeću Charles Babbage pokušava ukloniti nedostatke dotadašnjih mehaničkih kalkulatora te najprije izrađuje diferencijalni stroj, a kasnije i analitički stroj za rješavanje različitih zadataka. Po strukturi analitički stroj za računanje imao je sve elemente suvremenih računala: ulazne uređaje, memoriju, centralnu jedinicu, program na bušenim karticama i izlazni uređaj [5]. Matematičarka Augusta Ada Byron (Ada Lovelace) predložila je Charlesu Babbageu korištenje binarnog brojevnog sustava kako bi se analitički stroj mogao koristiti u znanstvene i praktične svrhe. Njezin opis koraka i postavke analitičkog stroja za izračunavanje Bernoullijevih brojeva smatraju se prvim računalnim programom [6].

Nakon završetka Drugog svjetskog rata John W. Mauchley i John Presper Eckert završavaju izgradnju prvog električnog računala opće namjene ENIAC-a (engl. *Electronic Numerical Integrator and Computer*) koji umjesto mehaničkih dijelova koristi elektroničke cijevi. ENIAC je bilo masivno računalo, teško oko 30 tona, koristilo je između 150 i 200 kW snage i bilo je iznimno komplicirano jer je bilo izrađeno od oko 19000 cijevi, 1500 releja i više od 100 000 otpornika, kondenzatora i zavojnica čime je zauzimao jako velik prostor unutar zgrade [7].

Značajan napredak u smjeru smanjenja računala postigli su Walter Brattain, John Bardeen i William Shockley izumom tranzistora 1947. godine koji je mogao zamijeniti elektroničke cijevi [8]. Napretkom tehnologije tranzistori su smanjeni i uz primjenu silicija kao poluvodiča Jack Kilby je 1959. godine razvio integrirane krugove [9]. Računala su postala manja, povoljnija, brža i energetski učinkovitija. Daljnji razvoj u smjeru manjih, bržih i jeftinijih računala dogodio se 1971. godine kad je Intelov inženjer Marcian Hoff izdvojio integrirane krugove u zasebnu jedinicu računala – mikroprocesor [10]. Nova računala svojom su snagom i cijenom omogućila razvoj grafičkog sučelja, korištenje dodatnih ulaznih jedinica uz tipkovnicu i razvoj računalnih mreža. Zbog smanjenih troškova i manjih dimenzija počela su se razvijati i prva osobna računala. IBM je 1981. godine izdao svoje prvo osobno računalo, a zahvaljujući prepoznatljivom brendu i marketingu, računalo je našlo široku primjenu u poslovnom svijetu

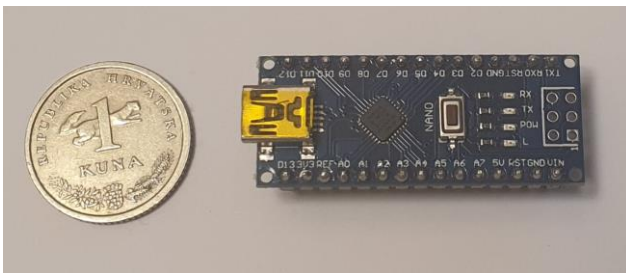
[11]. Nekoliko godina kasnije, Macintosh izdaje svoje prijenosno računalo koje je bilo teško oko 7,5 kilograma i koštalo je oko 6500 američkih dolara [12]. Računala se tijekom narednih godina smanjuju, postaju sve jača, jeftinija, zauzimaju manje prostora, imaju sve širu primjenu. Paralelno s njima razvijaju se i programski jezici koji omogućavaju pisanje računalnih programa i korištenje računala u različite svrhe. No, unatoč tome, rad s mikrokontrolerima je kompliciran te zahtijeva poznavanje strojnog jezika i arhitekturu pojedinog sklopovlja kako bi se napisali programi. Znatna promjena dogodila se 2005. godine u Italiji gdje je grupa studenata kreirala Arduino i time pojednostavnila programiranje i rad s mikrokontrolerima [13]. Arduino je veličine osobne iskaznice i koristi se za razne hobističke projekte kao što su zabava (letjelice i vozila), upravljanje kućanskim uređajima i rasvjetom, mjerenje temperature, vlage i sl. Iako je Arduino zaokupio pažnju svih ljubitelja tehnologije, koristi se samo za programiranje i upravljanje sklopovlja. Sedam godina kasnije (2012.) u Velikoj Britaniji pojavljuje se Raspberry Pi. Raspberry Pi također je veličine kreditne kartice, ali je za razliku od Arduina kompletno računalo koje omogućava povezivanje na mrežu, korištenje zaslona putem HDMI kabela, povezivanje USB uređaja i vlastiti operativni sustav baziran na Linux Debianu (prethodno nazivan Raspbian, danas Raspberry Pi OS). Dvije godine kasnije, Raspberry Pi dolazi u inačici Zero koja je predstavljena kao računalo od pet dolara [14]. Paralelno se razvijaju i ostali proizvodi cjenovno i performansama slični Arduino i Raspberry Piju. Nekoliko alternativa Arduino su: Teensy [15], Particle Photon [16], Sparkfun [17], Adafruit Feather [18], BeagleBoard [19], Digispark [20], a postoji i hrvatska verzija Croduino [21]. Nekoliko alternativa za Raspberry Pi su: Banana Pi [22], Orange Pi [23] i La Frite [24]. Iako automatizacija i upravljanje na daljinu postoji već nekoliko desetljeća, razvoj tehnologije u smjeru izrade manjih i jačih uređaja s manje potrošnje možemo promatrati kao i dolazak osobnih računala u privatnu upotrebu. Bill Gates je s Microsoftom imao viziju o „računalu na svakom stolu u svakom domu” [25], a upravo je smanjenje računala na veličinu osobne iskaznice dovelo do toga da danas u domovima imamo i više računala, ne samo na stolu već i u džepu, na zidu, vratima, četkicama za zube, rasvjetnim tijelima itd.



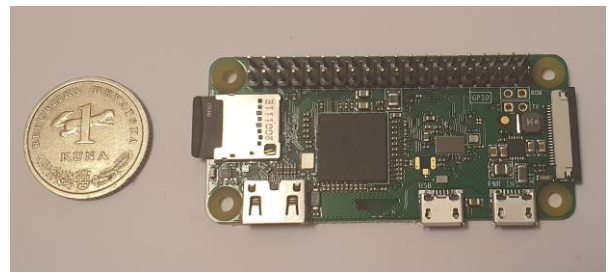
Slika 1. ENIAC [43]



Slika 2. IBM PC [44]



Slika 3. Arduino NANO klon



Slika 4. Raspberry Pi Zero WiFi

2. Internet stvari

Cijena računala i procesora u povijesti je bila u rangu cijene automobila dok danas procesor iste snage možemo dobiti za cijenu menija u McDonald'su. Zbog niske cijene procesore danas možemo koristiti bilo gdje i u bilo koju svrhu, čak i u četkicama za zube, rasvjetnim tijelima, igračkama, mjernim stanicama za vrijeme i sl., a to nam omogućuje da učinimo stvari „inteligentnima”, tj. da prate naše navike, odgovore na naš pokret i odrade interakciju s korisnikom. Unatoč brojnim dostupnim elementima koje možemo povezati, razvoj se nastavlja te svakodnevno svjedočimo novim uređajima koji odrađuju posao bolje ili imaju neku novu namjenu.

Pojam internet stvari (engl. *Internet of things*) dobio je pozornost na EPC simpoziju (*Electronic Product Code*) u Chicagu 2003. godine. Auto-ID Centar predstavio je svoju inicijalnu verziju automatskog identificiranja i praćenja protoka robe u lancima opskrbe robom. Pojam se prvi put pojavio u članku Auto-ID Centra o EPC-u, a brojni su istraživači i entuzijasti nastavili koristiti pojam. [26]

Internet stvari je koncept koji spaja virtualni svijet informacijskih tehnologija s realnim svijetom stvari. Na taj način realni svijet postaje dostupniji preko računala i umreženih uređaja u poslovnom i privatnom svijetu. Pristup realnim informacijama u realnom vremenu omogućava donositelju odluka bolju poziciju za donošenje pravovremene i točne odluke, bilo da je riječ o donošenju poslovne odluke (kao što je na primjer povećanje proizvodnje atraktivnog proizvoda) ili odluke koja nam olakšava svakodnevni život (kao što je na primjer uzimanje kišobrana ili dodatnog sloja odjeće na temelju temperature i vlage izvan kuće).

2.1. Definicija

Definicija interneta stvari mijenjala se s vremenom, uzimajući u obzir nove tehnološke pomake u poljima primjene. Grupa europskih istraživačkih projekata na području interneta stvari (engl. *Cluster of European Research Projects on the Internet of Things, CERP-IoT*) je u svom Strateškom istraživačkom planu (engl. *Strategic Research Agenda, SRA*) internet stvari definirala kao:

"Internet stvari je integriran dio budućeg interneta definiran kao dinamična globalna mrežna infrastruktura koja ima mogućnost samopodešavanja putem standardnih interoperabilnih komunikacijskih protokola u kojima virtualne i fizičke stvari imaju identitet, fizičke atribute i virtualne osobnosti, koriste inteligentna sučelja i integrirane su u informacijsku mrežu. Kod interneta stvari od stvari se očekuje da postanu aktivni dionici u poslu, informacijama i društvenim procesima u kojima imaju mogućnost međusobne interakcije i

komunikacije kao i komunikacije i interakcije s okolinom razmjenu podataka i informacija koje su prikupili. Stvari u tom procesu autonomno reaguju na stvari i događaje iz realnog svijeta te utječu na njega izvođenjem procesa koji pokreću aktivnosti i stvaraju usluge s direktnom ljudskom intervencijom ili bez nje. Sučelja promatraju interakciju pametnih objekata putem interneta, dohvaćaju i mijenjaju njihovo stanje i ostale informacije povezane s njima, uzimajući u obzir aspekte sigurnosti i privatnosti." [27]

Grupa autora Dieter Uckelmann, Mark Harrison i Florian Michahelles smatra da, iako definicija navodi sve tehničke komponente interneta stvari, definicija ima tri nedostatka. Prvi nedostatak je da su sve navedene komponente dovedene u relaciju s ostalim vizijama poput sveprisutnog računanja (engl. *ubiquitous computing*) te ih je teško razlikovati. Drugi nedostatak je izostavljanje povezanosti s Web 2.0 zbog paralelnog razvoja Web 2.0 i interneta stvari. Smatraju da se funkcionalnosti Web 2.0 razmatraju kao dodatak fokusa na korisnika umjesto kao dio interneta stvari. I treći nedostatak je nepovezanost sa samoodrživošću i uspješnim konceptom u budućnosti. Iako ekonomski pogled nikad neće biti u definiciji interneta stvari, smatraju da je ekonomski uspjeh i prilagodba tržištu podjednako važna kao i tehnička održivost. Benefiti za poslovnu okolinu i pojedinca osiguravaju želju za korištenjem interneta stvari i ulaganja. Definiciji također pristupaju i sa strane logistike koja se bavi problemom da pravi proizvod u ispravnoj mjeri u točno vrijeme dođe na pravo mjesto u ispravnom stanju po ispravnoj cijeni. Spajanjem ovih dvaju pogleda, internet stvari definiraju kao:

„Internet stvari povezuje jednoznačno identificirajuće stvari s njihovom virtualnom reprezentacijom na internetu pružajući dodatne informacije o njihovom identitetu, statusu, lokaciji ili bilo kojom drugom relevantnom poslovnom, društvenom ili privatnom informacijom zbog financijskog ili nefinancijskog benefita uzimajući pritom u obzir da benefit premašuje uloženi napor prikupljanja i omogućuje pristup informacijama bez prethodno definiranih dionika. Internet stvari nisu sinonim za sveprisutno računanje (engl. *ubiquitous/pervasive computing*), internet protokol, komunikacijsku tehnologiju, ugrađene uređaje (engl. *embedded devices*), njihovu primjenu, internet ljudi (engl. *Internet of People*) ili intranet stvari (engl. *Intranet of Things*), ali objedinjuje aspekte i tehnologije svih pristupa." [28]

2.2. Područje primjene

Primjenu interneta stvari danas možemo vidjeti gotovo u bilo kojem području. Najprepoznatljiviji i najpoznatiji ljudima općenito su mali uređaji koje nosimo na rukama, pametni satovi i narukvice, koje koristimo za brojanje koraka tijekom dana, pulsa te kao podsjetnik da prošetamo nakon dugog sjedenja ili za praćenje kvalitete sna.

Područja primjene možemo podijeliti na: [29]

1. zdravlje
 - a) otkrivanje pada – posebice kao pomoć starijim ili osobama s invaliditetom koje žive same
 - b) medicinski hladnjaci – upravljaju uvjetima u hladnjacima koji sadrže injekcije, lijekove i organske elemente
 - c) sportski dodaci – prate vitalne životne znakove te mjere učinak i kretanje
 - d) nadgledanje pacijenata – prate stanje pacijenata unutar bolnica ili domova za starije
 - e) ultraljubičasto zračenje – mjere UV zračenje i upozoravaju ljude kad nije dobro biti izložen Suncu
2. poljoprivredu i uzgoj životinja
 - a) hidroponi – održavanje idealnih uvjeta za biljke
 - b) briga o mladuncima – održavanje uvjeta na farmama kako bi se osiguralo preživljavanje i zdravlje mladunaca
 - c) praćenje životinja – lociranje i identifikacija životinja na otvorenim pašnjacima ili u stajama
 - d) razina opasnih plinova – upravljanje klimatizacijom i protokom zraka na farmama kako bi se otkrile razine opasnih plinova i osigurao siguran prostor
3. automatizaciju kuće
 - a) korištenje struje i vode – praćenje potrošnje struje i vode radi smanjenja troškova i potrošnje resursa
 - b) daljinsko upravljanje – upravljanje na daljinu uređajima u kućanstvu (primjerice mikrovalne pećnice da podgrije hranu prije nego uđemo u kuću ili da klimatizacijski uređaj prilagodi temperaturu prostorije kad smo na putu od radnog mjesta do kuće)
 - c) sigurnost – praćenje neovlaštenih ulazaka, otvorenih prozora ili otključavanje vratiju biometrijskim podacima umjesto ključem
 - d) očuvanje umjetnina – nadgledanje uvjeta unutar muzeja kako bi se spriječilo uništavanje umjetnina u njima

4. agrokulturu

- a) povećanje kvalitete vina – nadgledanje vlažnosti tla i osvjetljenja kako bi se osigurala dovoljna količina šećera i kvaliteta grožđa
- b) zelene kuće – upravljanje mikroklimatskim uvjetima za maksimiziranje produktivnosti voća i povrća te njihove kvalitete
- c) golf tereni – selektivno navodnjavanje terena kako bi se smanjila potrošnja vode i navodnjavanje područja s dostatnom vlažnošću
- d) mreža meteoroloških postaja – proučavanje vremenskih uvjeta radi što točnije prognoze vremena na pojedinim područjima kako bi se pravovremeno obavijestilo o mogućnosti tuče, snijega, vjetrova ili drugih nepovoljnih uvjeta za biljke
- e) kompost – upravljanje vlagom i temperaturom u sijenu, travi i drugim biljkama kako bi se izbjeglo stvaranje gljivica i drugih mikrobioloških zagađenja

5. logistiku

- a) kvaliteta prijevoznih uvjeta – nadgledanje vibracija, udaraca, otvaranja kontejnera i drugih događanja kako bi se osigurao transportni lanac
- b) lokacija proizvoda – lociranje proizvoda u velikim skladištima ili lukama
- c) otkrivanje neadekvatnog mjesta skladištenja – upozorenja o emisijama opasnih tvari iz kontejnera koji sadrže zapaljive ili eksplozivne tvari
- d) praćenje voznog parka – praćenje vozila koje prevoze stvari visoke vrijednosti poput umjetnina, nakita ili posebne važnosti kao što su lijekovi, ljudski organi ili opasna roba (vojno oružje, eksplozivi)

6. proizvodnju

- a) M2M (engl. *Machine to machine*) – automatska dijagnoza i kontrola proizvoda
- b) kvaliteta zraka u proizvodnim pogonima – mjerenje i praćenje kvalitete zraka i opasnih plinova u kemijskim postrojenjima radi osiguranja zdravlja i života radnika
- c) praćenje temperature – mjerenje i kontrola unutar industrijskih hladnjaka radi očuvanja osjetljivih proizvoda
- d) mjerenje ozona – mjerenje i praćenje količine ozona u komorama za sušenje mesa
- e) lokacija proizvoda – praćenje lokacije proizvoda unutar skladišta pomoću aktivnih i pasivnih oznaka (ZigBee, RFID/NFC)
- f) automatska dijagnoza stanja vozila – praćenje stanja vozila u realnom vremenu kako bi se pravovremeno obavijestila osoba koja njime upravlja o eventualnim opasnostima i savjetima

7. prodaju (engl. *Retail*)

- a) kontrola lanca opskrbe – praćenje stanja skladišta tijekom cijelog lanca opskrbe i praćenje stanja proizvoda

- b) beskontaktno plaćanje (NFC) – obrada plaćanja u različitim područjima prodaje (supermarketi, teretane, zabavni parkovi, trgovine odjećom i obućom itd.)
 - c) pametna kupovina – dohvaćanje informacija u realnom vremenu sukladno navikama i željama kupaca, informiranje o alergenima i roku isteka hrane
 - d) pametno upravljanje zalihama – praćenje i pomicanje robe na policama trgovina i skladištima i automatizacija obnove zaliha
8. sigurnost i hitne slučajeve (engl. *Security and Emergencies*)
- a) kontrola pristupa – kontrola i nadgledanje pristupa u zaštićene zone i otkrivanje nedopuštenih kretanja
 - b) prisutnost tekućina – otkrivanje prisutnosti tekućina u podatkovnim centrima, skladištima i podrumima kako bi se spriječile štete na uređajima
 - c) otkrivanje radijacije – distribuirano mjerenje i kontrola razine radijacije u nuklearnim elektranama i okolici kako bi se na vrijeme detektirala zračenja koja bi mogla prouzrokovati zdravstvene probleme za radnike
9. pametna mjerenja (engl. *Smart Metering*)
- a) pametna mreža – nadgledanje i upravljanje potrošnjom električne energije u distribucijskoj mreži
 - b) razina sadržaja u bačvama – nadgledanje razina vode, ulja, goriva u bačvama i cisternama
 - c) solarni sustavi – nadgledanje i optimizacija solarnih sustava kako bi se povećala proizvodnja električne energije (npr. praćenje položaja Sunca, vrijeme izlaska i zalaska Sunca, vremenske prilike i sl.)
 - d) kontrola toka – mjerenje i kontrola pritiska tekućina u transportnim cijevima
 - e) zabava – mjerenje sila i brzina u zabavnim parkovima kako bi se osigurala sigurnost korisnika
10. pametnu vodu (engl. *Smart Water*)
- a) kvaliteta vode – nadgledanje i mjerenje kvalitete vode koja se distribuira korisnicima
 - b) otkrivanje kemikalija – mjerenje i otkrivanje kemikalija u rijekama i distribucijskom sustavu
 - c) upravljanje bazenima – praćenje kvalitete vode u bazenima te daljinsko upravljanje uvjetima u bazenu
 - d) mjerenje zagađenosti mora – praćenje i rano otkrivanje otpada u moru kako bi se zaštitila ribolovna područja, područja uzgoja, prirodne ljepote te biljni i životinjski svijet podmorja
 - e) otkrivanje curenja – otkrivanje prisutnosti tekućina izvan transportnih ili skladišnih posuda kao i mjerenje tlaka u posudama kako bi se rano otkrilo oštećenje na spremnicima

- f) poplave – mjerenje i nadgledanje razine vodostaja u rijekama kako bi se što ranije otkrila mogućnost izljeva iz korita rijeke i ugrožavanje okolnog stanovništva

11. pametan okoliš (engl. *Smart Environment*)

- a) otkrivanje požara – nadgledanje i otkrivanje plinova i ranih znakova vatre kako bi se što točnije odredila lokacija na kojoj je izbio požar i reagiralo u što kraćem intervalu kako bi se spriječilo širenje
- b) zagađenje zraka – mjerenje razine emisije CO₂ u tvornicama, zagađenja koje stvaraju automobili ili životinje na velikim farmama
- c) razina snijega – mjerenje kvalitete snijega kako bi se u realnom vremenu dobila informacija sa skijaških staza ili spriječila katastrofa ranim otkrivanjem lavina
- d) klizišta i lavine – kontrola vlažnosti tla, vibracija i gustoće zemlje kako bi se otkrili uzroci koji utječu na prirodne katastrofe i spriječile ili umanjile štete koje prouzrokuju
- e) potresi – distribuirano mjerenje i otkrivanje potresa u specifičnim područjima podrhtavanja tla

12. pametne gradove (engl. *Smart Cities*)

- a) parkirne zone – praćenje i informiranje o slobodnim parkirnim mjestima u gradu
- b) praćenje stanja građevina – mjerenje vibracija i stanja građevinskih materijala u građevinama, mostovima i povijesnim spomenicima
- c) karte buke – mjerenje jačine zvukova u svim područjima grada kako bi se u realnom vremenu dobila karta buke
- d) otkrivanje pametnih telefona – otkrivanje prisutnosti pametnih telefona kako bi se ponudila paleta usluga ili turističkih znamenitosti u gradu
- e) otkrivanje elektromagnetskih zračenja – mjerenje razine elektromagnetskog zračenja antena, odašiljača i bežičnih mreža
- f) promet – praćenje kretanja i mjerenje gustoće prometa vozila i pješaka kako bi se u realnom vremenu mogli optimizirati putevi kretanja
- g) sigurnost u prometu – upozorenje vozačima na vremenske prilike ili neočekivane događaje poput nesreća i prometnih gužvi
- h) osvjetljenje – mjerenje vremenskih uvjeta i vanjskog osvjetljenja kako bi se na najbolji način optimiziralo korištenje ulične rasvjete
- i) upravljanje otpadom – mjerenje količine otpada u kantama za smeće kako bi se optimizirale rute i učestalost odvoza otpada.

2.3. Uređaji interneta stvari

Cijena, jednostavnost upotrebe, visoka razina upotrebljivosti za različite projekte Arduino je svrstala u jednu od najpoznatijih platformi otvorenog koda na svijetu. Zbog otvorenog pristupa danas na tržištu postoje mnogi mikrokontroleri koji omogućavaju jednostavnije programiranje jezicima više razine bez upotrebe programatora. Programiranje se provodi preko integriranog razvojnog okruženja (Arduino IDE) na osobnom računalu i pohranjuje se na Arduino pomoću USB kabela. Programski jezik je Arduino programski jezik i za upravljanje sensorima i ostalim uređajima koriste se dostupne biblioteke u knjižnici koja je dostupna u Arduino IDE.

Raspberry Pi je mini računalo na koje se standardiziranim priključcima (USB, HDMI) mogu priključiti standardni ulazni i izlazni uređaji poput pokazivača, tipkovnice i zaslona. Programiranje se vrši preko proizvoljnog integriranog razvojnog okruženja koristeći programski jezik Python i GPIO biblioteku za upravljanje ulazima i izlazima. S obzirom na to da se koristi Python, moguće je dodati bilo koju biblioteku te jednostavnije izvršavati zadatke više razine kao na primjer programiranje za umjetnu inteligenciju.

2.3.1. Arduino

Arduino je elektronička platforma otvorenog koda temeljena na sklopovlju i programskoj opremi koja je jednostavna za korištenje. Arduino ploče mogu čitati ulaze (senzor svjetla, pritisak tipke ili poruka na Twitteru) i pretvoriti ih u izlaze (aktiviranje motora, uključivanje LED ili objave poruke na internetu). Korisnik može odrediti što će pločica odraditi programiranjem seta instrukcija koje će obraditi mikrokontroler. Programiranje se izvodi u Arduino razvojnom okruženju koristeći Arduino programski jezik baziran na razvojnom okruženju (engl. *framework*) Wiring. [30]

Arduino se koristi u tisućama projekata, od jednostavnih objekata u svakodnevnoj upotrebi do kompleksnih znanstvenih instrumenata. Svjetska zajednica studenata, hobista, umjetnika, programera i profesionalaca daje doprinos razvoju dostupnog znanja kako bi pomogli početnicima i stručnjacima. [30]

Arduino je izrađen na Institutu za interaktivni dizajn u Ivrei (Italija) kako bi olakšao prototipiranje studentima bez pozadinskog znanja elektronike i programiranja. Nakon što je dobio pozornost šire javnosti, Arduino se prilagodio novim potrebama i izazovima te time odskočio od jednostavnih 8 bitnih pločica u proizvod koji se koristi u internet stvarima, 3D ispisu i ugrađenim okruženjima. Sve Arduino pločice otvorenog su koda kako bi potaknuli korisnike

da ih izrađuju i prilagođavaju svojim specifičnim potrebama. Softver je također otvorenog koda i razvoju doprinose korisnici diljem svijeta. [30]

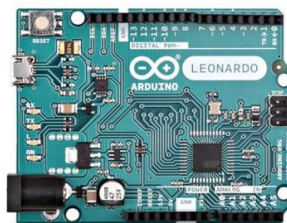
Zahvaljujući svojoj jednostavnosti i niskoj cijeni, profesori i učenici/studenti koriste Arduino u izradi jeftinih instrumenata kako bi proučavali kemijske i fizikalne veličine ili se usmjerili u programiranje i robotiku. Arduino također koriste i umjetnici, dizajneri, glazbenici. Iako postoje mnogi drugi proizvođači mikrokontrolera i platformi mikrokontrolera te svi pokušavaju pojednostavniti komplicirani proces programiranja mikrokontrolera, Arduino ima nekoliko prednosti u odnosu na ostale: [30]

- cijena – Arduino ploče su relativno niže cijene od ostalih platformi mikrokontrolera
- rad na više operativnih sustava – Arduino IDE radi na Windows, MacOSX i Linux operativnim sustavima. S obzirom na otvoreni kod, danas je moguće pronaći integracije postojećih razvojnih okruženja s modulima koji omogućavaju programiranje za Arduino (Microsoft Visual Studio, Atom – PlatformIO itd.).
- jednostavno razvojno okruženje – Arduino IDE je jednostavan za korištenje, a moguće ga je prilagoditi za napredno korištenje. Arduino IDE u svojoj osnovnoj inačici izgledom je ružan te nedostaje automatsko završavanje koda koje olakšava programiranje.
- otvoreni kod – Arduino je objavljen kao alat otvorenog koda što omogućava proširivanje prema potrebi, a isto tako velika zajednica vraća dio koda u buduće funkcionalnosti što omogućava brži razvoj i održivost same platforme. Isto vrijedi i za elektroničke sheme koje su objavljene pod Creative Commons licencom što omogućava korisnicima da prouče na koji način pločica radi, prošire mogućnosti i objave ideje online kako bi ih i drugi mogli koristiti.

Pregled nekih Arduino pločica možemo vidjeti u nastavku. Svaka od njih ima različite performanse i veličinu pa svaka ima drugačiji način primjene.



Slika 5. Arduino UNO [29]



Slika 6. Arduino LEONARDO [30]



Slika 7. Arduino NANO [34]



Slika 8. Arduino MICRO [32]



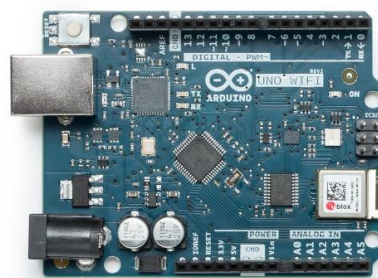
Slika 9. Arduino MEGA [33]



Slika 10. Arduino NANO 33 IoT [33]



Slika 11. Arduino MKR WiFi 1010 [35]



Slika 12. Arduino UNO WiFi REV2 [36]



Slika 13. Arduino MKR GSM 1400 [37]

- Arduino UNO – pločica bazirana na ATmega328P mikrokontroleru. Ima 14 digitalnih ulaza/izlaza i 6 analognih ulaza. Najjednostavnija pločica za početnike jer je vrlo dobro dokumentirana. [31]
- Arduino LEONARDO – pločica bazirana na Atmega32u4 mikrokontroleru. Ima 20 digitalnih ulaza i 12 analognih ulaza. Klasična pločica koja može služiti kao pokazivač ili tipkovnica. [32]
- Arduino NANO – pločica je bazirana na ATmega328 mikrokontroleru. Ima 22 digitalna ulaza/izlaza i 8 analognih ulaza. Arduino NANO je najmanja varijanta pločice. [33]
- Arduino MICRO – pločica bazirana na Atmega32u4 mikrokontroleru. Ima 20 digitalnih ulaza/izlaza i 12 analognih ulaza. Mala pločica koja može služiti kao pokazivač ili tipkovnica. [34]
- Arduino MEGA – pločica bazirana na Atmega2560. Ima 54 digitalna ulaza/izlaza, 16 analognih ulaza i 4 serijska porta. [35]

- Arduino NANO 33 IoT – pločica bazirana na ARM Cortex MO 32-bitnom SAMD21 mikrokontroleru. Ima 14 digitalnih ulaza/izlaza, 8 analognih ulaza te integrirane module za WiFi i Bluetooth. [36]
- Arduino MKR WiFi 1010 – pločica bazirana na ARM Cortex MO 32-bitnom SAMD21 mikrokontroleru. Ima 8 digitalnih ulaza/izlaza, 7 analognih ulaza, 1 analogni izlaz te integrirane module za WiFi i Bluetooth. [37]
- Arduino UNO WiFi Rev2 – pločica bazirana na Atmega4809 mikrokontroleru. Ima 14 digitalnih ulaza/izlaza, 6 analognih ulaza. Pločica je standardni UNO sa dodanim integriranim WiFi i Bluetooth modulima. [38]
- Arduino MRK GSM 1400 - pločica bazirana na ARM Cortex MO 32-bitnom SAMD21 mikrokontroleru. Ima 8 digitalnih ulaza/izlaza, 7 analognih ulaza i 1 analogni izlaz te integrirani modul za GSM/3G povezivanje. [39]

2.3.2. Raspberry Pi

Raspberry Pi je mini računalo s operativnim sustavom Raspberry Pi OS (prethodno poznat pod nazivom Raspbian) baziran na Debian Linux operativnom sustavu. Raspberry Pi ima mogućnost povezivanja s uobičajenim ulaznim i izlaznim jedinicama poput pokazivača, tipkovnice i zaslona standardiziranim priključcima kao što su USB i HDMI. Postoje različite verzije Raspberry Pi uređaja, ali za korisnika to ne predstavlja problem jer su svi međusobno kompatibilni. Raspberry Pi osim standardnih ulaza za ulazne i izlazne jedinice računala ima mogućnost komuniciranja s okolinom digitalnim ulazima i izlazima. Za razliku od Arduina, svi ulazi su digitalni, te je u slučaju potrebe za analognim ulazima potrebno izraditi posebne module koji će omogućiti očitavanje analognih vrijednosti. Na Raspberry Pi možemo priključiti i posebne module kao što su na primjer kamera ili ekran na dodir, a podaci su spremljeni na mikro SD kartici na kojoj se nalazi i operativni sustav.

S obzirom na to da Raspberry Pi ima operativni sustav, programiranje se može vršiti u bilo kojem programskom jeziku koji se može kompajlirati ili interpretirati na Linux operativnom sustavu, no uobičajena je primjena Python programskog jezika zbog jednostavnosti i širokih mogućnosti koje Python pruža programerima. Iako je namjena Raspberry Pi uređaja široka, primarna namjena i cilj Raspberry Pi zajednice je pružanje edukacije u području računalne znanosti.

Verzije Raspberry Pi uređaja možemo vidjeti u nastavku.



Slika 14. Raspberry Pi 1 Model A+



Slika 15. Raspberry Pi 1 Model B+



Slika 16. Raspberry Pi 2 Model B



Slika 17. Raspberry Pi 3 Model A+



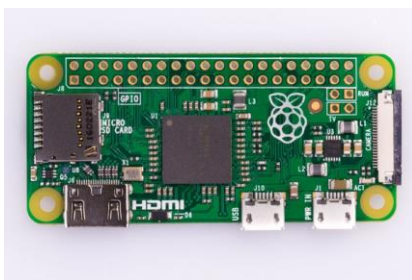
Slika 18. Raspberry Pi 3 Model B



Slika 19. Raspberry Pi 3 Model B+



Slika 20. Raspberry Pi 4 Model B



Slika 21. Raspberry Pi Zero



Slika 22. Raspberry Pi Zero W

Modeli se najlakše mogu usporediti tabličnim prikazom specifikacija. [40]

Model	Procesor	RAM	LAN	WiFi	Bluetooth	Broj digitalnih ulaza/izlaza
Raspberry Pi 1 Model A	ARM1176JZF-S, 700 MHz, 1 jezgra	256 MB	-	-	-	26
Raspberry Pi 1 Model A+	ARM1176JZF-S, 700 MHz, 1 jezgra	512 MB	-	-	-	40
Raspberry Pi 1 Model B+	ARM1176JZF-S, 700 MHz, 1 jezgra	512 MB	10/100M	-	-	40
Raspberry Pi 2 Model B	Cortex-A7, 900 MHz, 4 jezgre	1 GB	10/100M	-	-	40
Raspberry Pi 3 Model A+	Cortex-A53, 1.4 GHz, 4 jezgre	512 MB LPDDR2	10/100M	802.11 b/g/n/ac	4.2 BLE	40
Raspberry Pi 3 Model B	Cortex-A53, 1.4 GHz, 4 jezgre	1 GB LPDDR2	10/100M	802.11 b/g/n	4.2 BLE	40
Raspberry Pi 3 Model B+	Cortex-A53, 1.4 GHz, 4 jezgre	1 GB LPDDR2	10/100M	802.11 b/g/n/ac	4.2 BLE	40
Raspberyy Pi 4 Model B	Cortex-A72, 1.5 GHz, 4 jezgre	2/4/8 GB LPDDR4	10/100/1000M	802.11 b/g/n/ac	5.0	40
Raspberry Pi Zero	ARM1176JZF-S, 1 GHz, 1 jezgra	512 MB	-	-	-	40
Raspberry Pi Zero W	ARM1176JZF-S, 1 GHz, 1 jezgra	512 MB	-	802.11 b/g/n	4.1 BLE	40

Tablica 1. Usporedba specifikacija Raspberry Pi uređaja

2.3.3. Senzori i njihova primjena

Arduino i Raspberry Pi nemaju mogućnost reakcije i osjeta podražaja iz okoline. Kako bi omogućili da mjere vrijednosti i reagiraju sukladno njima u smjeru ostvarivanja cilja, na njih je preko analognih i digitalnih ulaza moguće spojiti senzore ili module sa sensorima. Razlika je u tome što su moduli pripremljeni kako bi se jednostavno i direktno spajali na Arduino i Raspberry Pi, a ako se koriste čiste komponente senzora ili elektroničke komponente, potrebno ih je pripremiti. Najjednostavniji prikaz je spajanje LED na jedan od uređaja. Iako na Raspberry Piju možda nećemo osjetiti neku razliku, kod Arduina sigurno hoćemo. LED ima radne napone (engl. *forward voltage*) između 1.7 i 3.2 V. Izlazni pinovi na Raspberry Pi uređajima imaju napon od 3.3 V dok Arduino na svojim izlazima daje 5 V. Spajanjem LED direktno na izlaz Arduina prouzročit će kratak bljesak i trajno oštećenje diode. Kako bi se smanjio (podijelio) napon, potrebno je u serijski spoj s LED spojiti otpornik određenog otpora i na taj način zaštititi diodu. Moduli su pripremljeni tako da korisnik ne mora razmišljati o elektrotehničkim komponentama i kako ih spojiti u strujni krug, već ih pomoću spojnih žica ili konektora jednostavno može priključiti na IoT kako bi započeo s radom i programiranjem uređaja.

2.3.3.1. GPS

GPS (engl. *Global Positioning System*) je pomoć kod navigacije koji je razvilo Američko zrakoplovstvo. Svaki satelit opremljen je s nekoliko atomskih satova koji precizno mjere vrijeme i pseudogeneratorom slučajnih brojeva u obliku pomičnog registra. GPS uređaj može razlikovati signale minimalno četiriju satelita i usporediti njihove pseudogenerirane slučajne brojeve i izračunati udaljenost uspoređujući vrijeme dolaska signala. Na temelju izračunatih udaljenosti pružaju informaciju o trenutnoj lokaciji.

2.3.3.2. Magnetometar

Postoje linearni magnetometri koji mjere ukupnu snagu magnetskog polja i vektorski magnetometri za mjerenje jačine magnetskih polja u određenom smjeru, odnosno dalje brojčanu vrijednost opisujući veličinu kuta između vektorskog magnetometra i magnetskip polova Zemlje. Modul magnetometra sadrži tri vektorska magnetometra postavljena tako da je između svakoga od njih kut od 90° što omogućuje softveru da interpretira analogne vrijednosti i izračuna gdje se nalaze magnetski polovi Zemlje.

2.3.3.3. Senzori prisutnosti objekta

Senzor prisutnosti objekta koristi se za provjeru prisutnosti ili odsutnosti objekta u prethodno određenom području, bez potrebe za mjerenjem udaljenosti objekta ili njegove brzine. Senzor se često koristi u automatiziranim sustavima kako bi se potvrdila prisutnost objekta na traci (punjenje boca, pakiranje kutija i sl.) ili za brojanje objekata. Senzor je našao upotrebu i u sigurnosnim sustavima za otkrivanje prisutnosti neovlaštenog pristupa ili otvorenih prozora i vrata. Postoje dvije osnovne vrste senzora: optički (fotoosjetljivi i fotoemitirajući) i magnetski.

2.3.3.4. Pasivni infracrveni senzori

Senzor za otkrivanje pokreta pomoću funkcije koja u omjer stavlja njihovu temperaturu i apsolutnu nulu. Senzor otkriva infracrveno zračenje u valnim duljinama od 10 mikrometara što približno odgovara temperaturi tijela ljudi i životinja. Primjenjuje se za otkrivanje pokreta za rasvjetna tijela, sigurnosne sustave (aktiviranje videonadzora ili alarma) i u automobilima (otkrivanje pješaka iza vozila).

2.3.3.5. Senzori udaljenosti

Senzori za mjerenje udaljenosti mogu biti infracrveni, ultrazvučni, kapacitativni, magnetski, induktivni, a mogu koristiti i druge metode za mjerenje udaljenosti objekta. Senzor mjeri udaljenost do fizičkog objekta. Izlaz iz senzora može biti analogni podatak (napon), serijski podatak ili modulacija valne duljine. Senzori se koriste za sprječavanje sudara robota ili automobila, u alarmnim sustavima, sustavima za otkrivanje razine tekućina u spremnicima. Upotrebu su također našli i u prijenosnim uređajima i pametnim telefonima gdje se koriste za uključivanje ekrana kad se mobilni telefon približi licu korisnika ili za gašenje ekrana kad korisnik prisloni mobilni telefon na uho prilikom razgovora.

2.3.3.6. Senzori rotacije

Upravljanje mehaničkim uređajima zahtijeva preciznu informaciju u poziciji pokretnih dijelova. Senzor rotacije daje informaciju o poziciji rotirajućih dijelova te može mjeriti tri veličine: kut, smjer i brzinu rotacije. Primjenjuju se kod solarnih panela za praćenje Sunca, vođenje i navigaciju, pozicioniranje antena te kontrolu vjetrenjača, a mogu se koristiti i za mjerenje brzine okretaja u vozilima i industrijskim procesima.

2.3.3.7. Senzori nagiba (engl. tilt sensors)

Postoje tri tipa senzora nagiba, a razlikuju se po broju osi. Senzor nagiba s jednom osi i jednim izlazom što omogućuje otkrivanje pomaka po horizontalnoj osi u ovisnosti prema okomitoj sili gravitacije. Senzor nagiba s dvije osi i dva izlaza sastoji se od dvaju senzora koji su pozicionirani tako da su im osi okomite pri čemu se očitavaju vrijednosti s obje osi u odnosu na okomitu silu gravitacije. Senzor nagiba s dvije osi i jednim izlazom pri čemu jedan senzor očitava kut pomaka na vertikalnoj osi u odnosu na horizontalnu. Senzor se najčešće sastoji od dviju čeličnih kuglica različitih promjera pri čemu jedna kuglica zatvara strujni krug kako bi se očitao nagib, dok druga djeluje kao uteg povećavajući pritisak i smanjujući vibracije.

2.3.3.8. Žiroskop (engl. gyroscope)

Žiroskop se opire rotaciji po bilo kojoj osi ako sila nije okomita na os. Smjestimo li žiroskop u kućište i omogućimo mu kretanje, možemo mjeriti pomak kućišta (objekta) u odnosu na žiroskop. Uzmimo za primjer letjelicu. Postavimo li kućište u unutrašnjost letjelice i pričvrstimo kućište žiroskopa, prilikom mijenjanja smjera žiroskop će ostati na istom mjestu te mjerenjem pomaka u odnosu na kućište možemo saznati pomak po osi. Prvi žiroskopi bazirani na čipovima koristili su se za otkrivanje pomaka u automobilima za aktivno kočenje, senzore za zračne jastuke i sprečavanje prevrtanja vozila. Danas žiroskope možemo vidjeti u pametnim telefonima, kontrolerima za igru, upravljanju kretanjem u virtualnoj stvarnosti te letjelicama i dronovima.

2.3.3.9. Senzori za mjerenje ubrzanja (engl. accelerometer)

Ubrzanje je promjena brzine u jedinici vremena te može biti pozitivno ako se brzina povećava i negativno ako se brzina smanjuje. Promjenu brzine možemo osjetiti u automobilu kod ubrzanja i kočenja, a isti princip koristi se i u senzoru za mjerenje ubrzanja mjereći silu. Postavimo li tri senzora da mjere sile na međusobno okomite osi, senzori mogu dati informacije o smjeru ubrzanja, nalazi li se objekt u slobodnom padu, gdje se u prostoru nalazi gore i dolje te sili prilikom sudara dvaju objekata. Senzori za mjerenje ubrzanja u prošlosti su bili laboratorijski uređaji za kalibraciju automobila, zrakoplova i drugih vozila, no njihova minijaturizacija i manja cijena dovela je senzor u manje uređaje poput mobilnih uređaja, tvrdih diskova, kamera, pokazivača i drugih uređaja.

2.3.3.10. Senzori vibracije

Senzor vibracije mjeri ponavljajuća mehanička kretanja. Većina verzija senzora sastoji se od sklopke koja je u otvorenom stanju te prilikom vibracije zatvara strujni krug stvarajući pritom frekvenciju napona. Postoje različite verzije senzora poput magnetskih ili izvedenih od piezoelektričnih kristala, a najjednostavnija izvedba je metalni pin u sredini kućišta koje sadrži zavojnicu, prilikom vibracije pin dodiruje zavojnicu i zatvara strujni krug. Senzori vibracije imaju različitu namjenu, a neke od njih su otkrivanje velikih vibracija na industrijskim strojevima, strojevima za pranje rublja, ali i igračkama ili za praćenje transporta osjetljivih uređaja.

2.3.3.11. Senzori sile

Senzor sile mjeri fizički pritisak koji osoba ili objekt stvaraju na njega, a zbog brzog očitavanja vrijednosti mogu mjeriti i promjenjive sile. Uobičajene su dvije izvedbe senzora, preko otpora i piezoelektričnih kristala. Senzor sile našao je primjenu u kontroli pritiska robotske ruke, mjernim vagama, otkrivanju putnika na sjedalima u automobilima, mjerenju koliko je puta pacijent ustao iz bolničkog kreveta dok bi u budućnosti mogao imati puno širu primjenu.

2.3.3.12. Senzori dodira

Senzor dodira otkriva dodir prsta na podlozi pri čemu nije potrebna upotreba sile kako bi tipka ili prekidač zatvorio strujni krug već se šalje električni signal. Senzore dodira u obliku tipki možemo vidjeti na mikrovalnim pećnicama, kuhinjskim aparatima, predočnicima, rasvjetnim tijelima, medicinskim i drugim uređajima. Nepostojanje mehaničkih dijelova i električnih kontakata čini ovaj senzor pouzdanijim, no nedostatak povratne informacije (uobičajeno je koristiti svjetlosni ili zvučni signal kod pritiska) čini ih neupotrebljivima na uređajima kod kojih se brzo pritišće velik broj senzora (npr. tipkovnica).

2.3.3.13. Zaslon na dodir

Iako se na prvu pomisao čini kako je zaslon izlazna jedinica namijenjena prikazu informacija korisniku, zaslon daje mogućnost otkrivanja dodira i upravljanja uređajem. Postoje dvije verzije izvedbe zaslona na dodir: otpornički i kapacitivni. Iako se upotreba otporničkih pogrešno pripisuje staroj tehnologiji koja više nije u upotrebi, prednosti su jednostavnost, niski troškovi i mogućnost korištenja kad korisnik koristi rukavice (npr. industrijska postrojenja, bolnice i sl.), no postoje i nedostaci. Moguće je očitati samo jednu točku ulaza, neki predočnici

ne očitavaju pritisak prsta, već je potrebno koristiti olovku (engl. *stylus*) i membrane na tipkovnici su osjetljive na oštre predmete.

2.3.3.14. Senzori razine tekućina

Postoje mnoge metode za mjerenje količine tekućine u bačvama i drugim posudama. Dvije najjednostavnije su mjerenje razine pomoću binarnih (digitalnih) razina i analognih pri čemu digitalnom razinom možemo označiti razinu do koje primjerice želimo puniti te kad količina tekućine dostigne tu razinu isključimo pumpu ili pak analognom vrijednošću odredimo koliko tekućine ima u posudi i matematičkim izračunom pretvorimo to u jedinicu volumena.

2.3.3.15. Senzori protoka tekućina i plinova

Senzor protoka tekućine mjeri protok tekućine kroz senzor ili uređaj na dva načina, vraćajući digitalni i analogni signal. Digitalni signal sugerira da postoji protok, dok analognim signalom i matematičkim izračunom napon možemo pretvoriti u protok volumena u jedinici vremena. Mjerenje protoka tekućina može biti zahtjevno što ovisi o gustoći tekućina pri čemu protok može biti mali što ga čini teškim za očitati. Iako senzori za protok tekućina imaju istu svrhu kao i senzori za protok plinova, zbog male gustoće plinova, senzori plinova često se svrstavaju u zasebnu kategoriju.

2.3.3.16. Senzori tlaka plina i tekućina

Senzor tlaka plinova i tekućina mjeri silu koju plin ili tekućina stvara na posudu u kojoj se čuva. Pritisak može biti stalan ili se mijenjati jako sporo i dinamičan. Senzori tlaka većinom su izrađeni za jednu namjenu. Primjenu su našli u meteorološkoj opremi (visinometri, barometri), industrijskoj proizvodnji, automobilskim gumama, motorima i sustavu kočenja i mjerenju pritiska u pametnim opskrbnim mrežama vode.

2.3.3.17. Senzor koncentracije plina

Senzori niske cijene bazirani na poluvodičkim komponentama omogućavaju otkrivanje plinova u atmosferi mjerenjem električnog otpora i kapacitivnosti određenog plina. Često se koriste u alarmnim sustavima kako bi upozorili na curenje opasnih i eksplozivnih plinova poput ugljičnog monoksida i propana, ali i za mjerenje kisika u zraku. Ovi se senzori također koriste i za otkrivanje alkohola ili vode u njihovu plinovitom obliku pružajući informaciju o količini alkohola u dahu ili mjerenju vlažnosti zraka s obzirom na količinu vodene pare. Vlažnost zraka je, osim za meteorološka mjerenja, vrlo važan faktor u industriji kako bi se spriječilo hrđanje

ili neispravan rad uređaja te kako bi se kontrolirali uvjeti proizvodnje i čuvanje hrane te spriječilo njezino kvarenje, stvaranje gljivica ili ostalih opasnih mikroorganizama.

2.3.3.18. Senzori za mjerenje svjetlosti

Senzori za mjerenje svjetlosti mjere jačinu svjetla u okolini vraćajući vrijednost koju je potrebno matematičkom formulom preračunati u mjernu jedinicu količine svjetla. Postoje tri vrste izvedbe senzora mjerenja svjetlosti:

- Fotootpornik – kod izloženosti svjetlu mijenja se otpor. U potpunom mraku fotootpornici mogu imati otpor i do $10\text{ M}\Omega$ dok se kod jakog svjetla otpor smanjuje na svega stotinjak Ω . Koristeći Ohmov zakon napon na otporniku možemo preračunati u količinu svjetla. S obzirom na to da se kod proizvodnje fotootpornika često koristi kadmijev sulfid, proizvodnja fotootpornika zabranjena je u pojedinim regijama kao što je npr. Europa.
- Fotodioda – kod izloženosti svjetlu fotodioda generira struju. Zbog vrlo niske struje dodaje se izvor istosmjerne struje. Mjerenjem struje ili napona vrijednosti možemo pretvoriti u količinu svjetla.
- Fototranzistor – kod izloženosti svjetlu kontrolira struju između kolektora i emitera. U potpunom mraku tranzistor propušta struju manju od 100 nA , dok kod izloženosti svjetlu struja može biti do 50 mA . Jačina struje koju fototranzistor može podnijeti veća je od struje koju može podnijeti fotodioda pa je zbog toga i namjena malo drugačija.

2.3.3.19. Senzori temperature

Senzori temperature imaju namjenu u različitim područjima, sustavima za klimatizaciju, uređajima za pranje rublja, hladnjacima, zamrzivačima, perilicama suđa i ostalim kuhinjskim aparatima, u kontroliranju temperature vode u bazenima i saunama, laserskim i 3D pisačima, kontroli industrijskih procesa i u medicini (medicinski hladnjaci za čuvanje uzoraka, cjepiva i lijekova). Postoji nekoliko vrsta izvedbe senzora temperature:

- NTC termistor – otpor termistora se smanjuje povećanjem temperature.
- PTC termistor – otpor termistora se povećava povećanjem temperature.
- Termalni spoj – spoj izrađen od bimetala (dviju vrsta materijala različitih temperaturnih karakteristika). Prilikom izloženosti temperaturi nastaje mali napon koji je često teško mjeriti (izražen u mili ili mikro voltima).

- Otpornički senzor temperature – funkcionalno radi isto kao i PTC termistor (povećanjem temperature raste otpor), ali je izrađen od čistog metala, a ne od poluvodiča.
- Poluvodički senzor temperature – integrirani strujni krug sastavljen od spojeva tranzistora što omogućava gotovo linearnu promjenu i nisku cijenu izrade, a moguće ga je i direktno spojiti na mikrokontroler.
- Infracrveni senzor temperature – za razliku od ostalih izvedbi, infracrveni senzor temperature je beskontaktni što ga čini idealnim za mjerenje temperatura materije s kojom senzori ne smiju biti u doticaju (zbog temperature, opasnih plinova i sl.). Radi na mjerenju radijacije crnog tijela (engl. *black-body radiation*) koju emitiraju svi materijali iznad apsolutne 0 (0° K).

2.3.3.20. Mikrofon

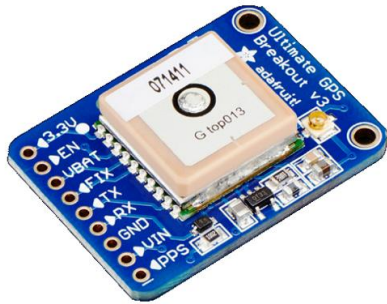
Zvuk se prenosi vibracijama zvučnih valova kroz medij prijenosa, najčešće zrak. Prilikom dolaska valova u uho, bubnjić vibracije pretvara u signale koje mozak interpretira. Mikrofon sadrži membranu koja pretvara vibracije iz zraka u analogni električni signal koji možemo pojačati, snimiti ili prenijeti drugim medijima prijenosa.

2.3.3.21. Senzor struje (engl. current sensor)

Senzor struje mjeri protok električne energije kroz vodič ili uređaj generirajući analognu vrijednost na izlazu koju možemo prikazati vizualno mjernim instrumentima ili proslijediti mikrokontroleru i izvršiti akciju na temelju primljenih vrijednosti (isključiti uređaj, mjeriti potrošnju).

2.3.3.22. Senzor napona (engl. voltage sensor)

Senzor napona mjeri električni potencijal između dviju točki strujnog kruga. Primjenjuje se za kontrolu strujnog kruga ili za mjerenje izlaznih veličina senzora.



Slika 23. GPS



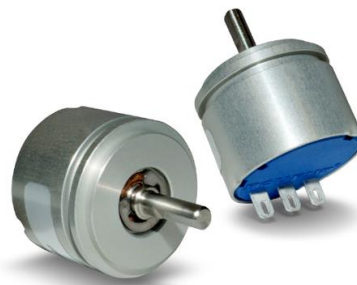
Slika 24. Magnetometar



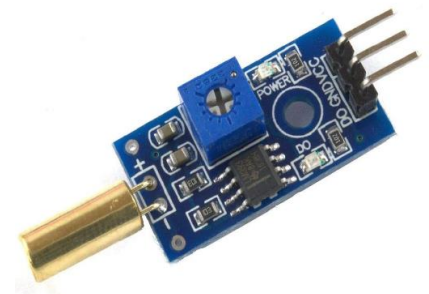
Slika 25. Pasivni infracrveni senzor



Slika 26. Senzor udaljenosti
(utrazvučni)



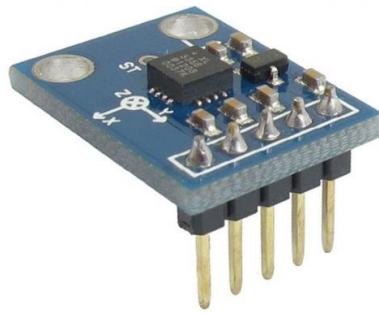
Slika 27. Senzor rotacije



Slika 28. Senzor nagiba



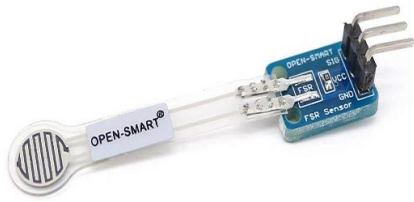
Slika 29. Žiroskop



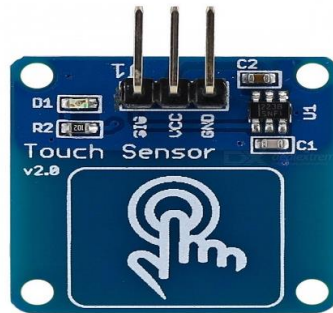
Slika 30. Senzor ubrzanja



Slika 31. Senzor vibracija



Slika 32. Senzor sile



Slika 33. Senzor dodira



Slika 34. Zaslona na dodir



Slika 35. Senzor razine tekućina



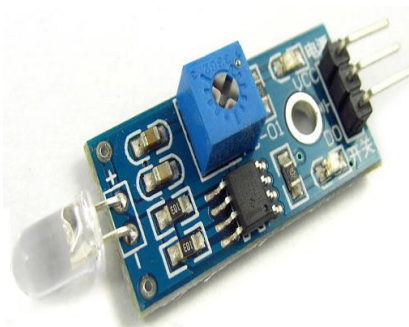
Slika 36. Senzor protoka tekućina



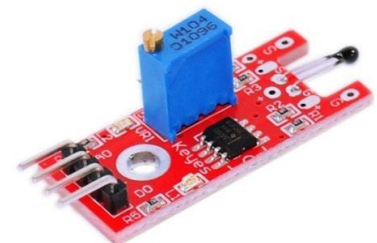
Slika 37. Senzor tlaka tekućina



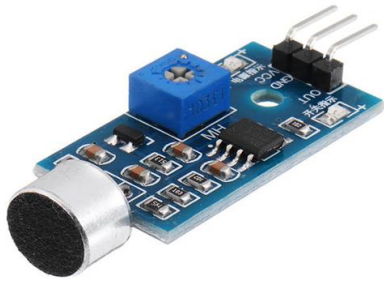
Slika 38. Senzor koncentracije plina



Slika 39. Senzor za mjerenje osvijetljenja sa fotodiodom



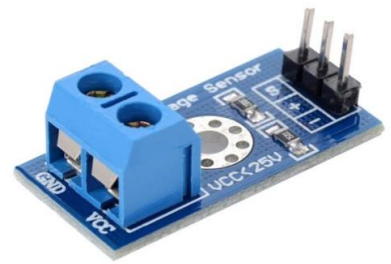
Slika 40. Senzor temperature



Slika 41. Mikrofon



Slika 42. Senzor struje



Slika 43. Senzor napona

3. Višeagentni sustavi

Distribuirana umjetna inteligencija (engl. *Distributed Artificial Intelligence – DAI*) bavi se istraživanjima u polju umjetne inteligencije koristeći distribuirane sustave s ciljem rješavanja kompleksnih problema iz svakodnevnog života. Distribuirana umjetna inteligencija obuhvaća tri različita područja: paralelnu umjetnu inteligenciju (engl. *parallel AI*), distribuirano rješavanje problema (engl. *Distributed problem solving – DPS*) i višeagentne sustave (engl. *Muti-agent systems – MAS*). Paralelna umjetna inteligencija bavi se povećanjem brzine operacija u paralelnim dretvama kako bi što prije riješila određeni problem, dok se distribuirano rješavanje problema usredotočuje na rješavanje problema pomoću dijeljenja resursa i znanja između velikog broja računalnih modula. Višeagentni sustavi bave se ponašanjem računalnih komponenti s ciljem rješavanja zadanog problema. Upute za oblikovanje izgleda rada.

3.1. Definicija agenta i višeagentnog sustava

Agent je bilo koji entitet koji ima mogućnost promatranja ili mjerenja okruženja kroz senzore i reakcije na podražaje iz okruženja kroz akcije. [41]

Autonomni agenti su računalni sustavi smješteni u kompleksno dinamično okruženje, imaju mogućnost promatranja okoline i autonomnog djelovanja u njoj ostvarujući pritom niz ciljeva ili zadataka zbog kojih su izrađeni. [42]

U ovim definicijama može se primijetiti nekoliko ključnih riječi: autonomija, cilj i ponašanje (djelovanje). Autonomija označava da agent radi samostalno, bez intervencije čovjeka i time ima kontrolu nad svojim akcijama. Definiranje cilja agenta označava rješenje zbog kojeg je agent dizajniran i postavljen u određeno okruženje, tj. što agent u specifičnom okruženju pokušava riješiti. Samostalno djelovanje s ciljem ostvarivanja definiranog cilja ostvaruje se definiranim ponašanjem agenta, a ponašanje je bazirano na akcijama koje se događaju nakon očitavanja vrijednosti na sensorima što ponašanja čini specifičnim za pojedinu okolinu u kojoj se agent nalazi.

Agente možemo podijeliti na fizičke i softverske. Ako je agent smješten u okruženje s pripadnim sensorima kako bi prikupio podatke iz njega, agent se smatra fizičkim agentom. Softverske agente možemo pronaći u bilo kojem operativnom sustavu. Pozadinski proces koji

se izvršava kako bi obavijestio korisnika o pristigloj poruci elektroničke pošte promatra se softverskim agentom. Takav agent nema fizičke senzore, ali u mogućnosti je softverski otkriti promjenu u svom okruženju i djelovati na nju.

Višeagentni sustav je fleksibilna mreža agenata koji zajedničkim radom traže odgovor na problem koji nadilazi individualne mogućnost ili znanja svakog agenta. [43]

3.1.1. Karakteristike agenata

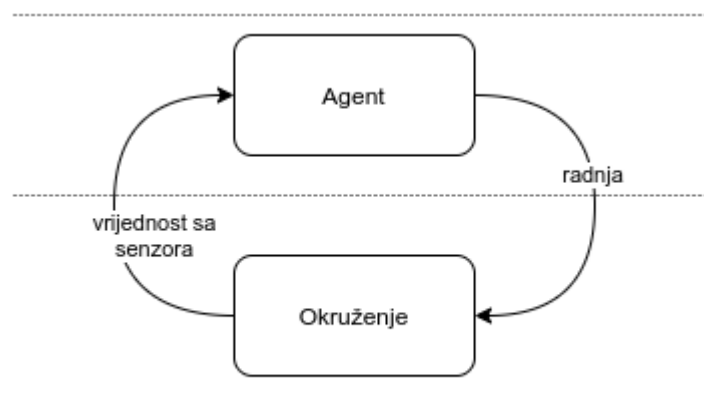
Navedene definicije agenata ne obuhvaćaju sve karakteristike koje bi agent trebao imati i koje ga razlikuju od ekspertnih sustava ili distribuiranih kontrolera. Neke od karakteristika su sljedeće:

- **Situiran (na pravom mjestu) (engl. *Situatedness*)** – označava interakciju agenta s okolinom dohvaćajući podatke o okolini sa senzora i reagirajući na nju putem akcija. Okruženje u kojem se nalazi agent prilagođeno je njegovu dizajnu i namjeni te su svi ulazi očitani direktno kao posljedica agentove reakcije s okruženjem. Agent direktno reagira na okruženje preko aktuatora pri čemu u promjenama sudjeluje direktno bez posrednika i bez uloge savjetnika. Ta karakteristika razlikuje ga od ekspertnih sustava u kojima se odluka izvršava preko posrednika.
- **Autonomija (engl. *Autonomy*)** – agenti su autonomni i mogu donositi i provoditi akcije bez vanjskog utjecaja drugih agenata u mreži ili ljudske intervencije. Na taj način unutarnja stanja agenata zaštićena su od vanjskog utjecaja i samim time izolirana od nestabilnosti koju mogu prouzročiti vanjski faktori.
- **Sposobnost donošenja zaključaka (inferencija) (engl. *Inferential capability*)** – Agent je u mogućnosti raditi unutar apstraktnih specifikacija cilja te pri tome dedukcijom i promatranjem informacija okruženja donijeti zaključak o aktivnosti ili radnji koja će pridonijeti ostvarenju njegova cilja.
- **Odaziv (engl. *Responsiveness*)** – agent ima mogućnost percepcije stanja okruženja i u mogućnosti je odgovoriti na događanja u tom okruženju u nekom zadanom vremenu kako bi poduzeo radnju i promijenio okruženje u kojem se nalazi. Ova je karakteristika ključna kod agenata koji se primjenjuju u okruženjima u kojem je potrebno ostvariti primjenu u realnom vremenu (engl. *real-time application*).

- **Proaktivnost (engl. *Pro-activeness*)** – agent mora biti u mogućnosti reagirati na oportunitetno ponašanje, tj. poduzeti akciju ako/kad se za to stvore uvjeti uzimajući pritom u obzir ostvarenje cilja, a ne samo impulzivno reagiranje na ulaze. Proaktivnost i poduzimanje akcija omogućuju agentu da se prilagodi promjenama u dinamičkom okruženju.
- **Društveno ponašanje (engl. *Social behaviour*)** – iako odluka agenata mora biti donesena neovisno o vanjskim utjecajima, agent mora biti u mogućnosti komunicirati s vanjskim izvorima i sustavima kad je to potrebno radi ostvarenja cilja zbog kojeg je izrađen. Također, agent mora biti u mogućnosti dijeliti znanje i pomoći ostalim agentima unutar sustava u rješavanju specifičnih problema. Dijeljenje znanja omogućuje agentima sposobnost učenja iz iskustava ostalih dionika komunikacije bez obzira na to je li riječ o ljudima, drugim agentima ili statističkim kontrolerima.

3.1.2. Okruženja agenata

Agent postavljen u određeno okruženje pomoću ulaznih parametara ili senzora može utjecati (izvoditi akcije) na svoju okolinu.



Slika 44. Agent u svojoj okolini [61]

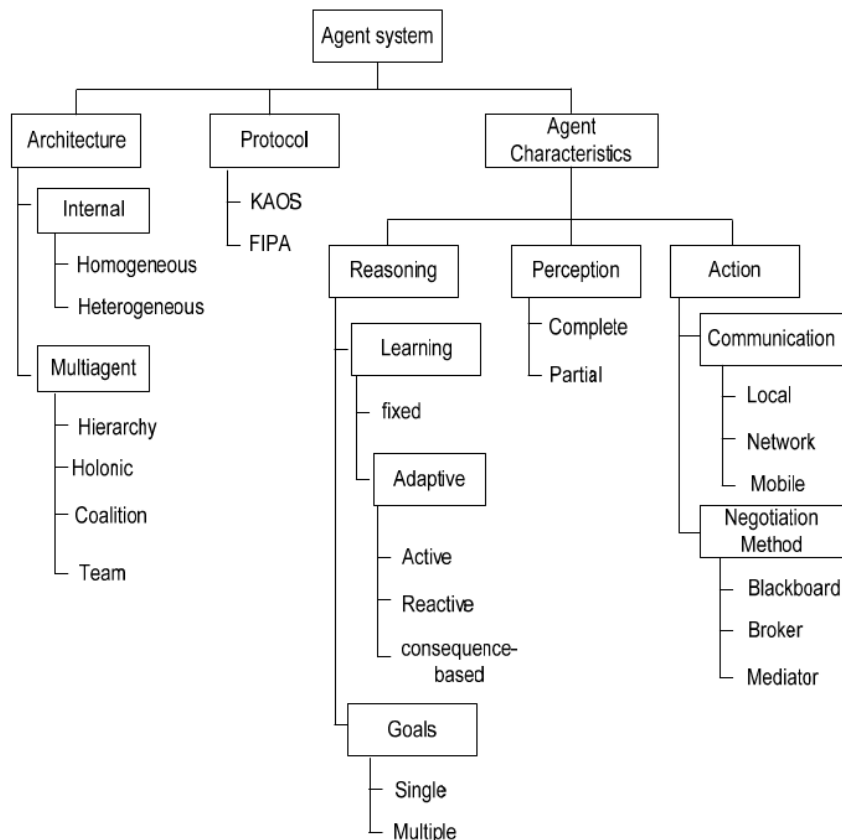
Russell i Norvig u svojoj klasifikaciji okruženja opisuju na sljedeći način: [41]

- **Pristupačna okruženja** – okruženja agenata u kojima agent ima pristup cjelokupnim, točnim i ažuriranim informacijama o okruženju. Uzimajući u obzir pristupačnosti okruženja, gotovo svako realno okruženje možemo smatrati nepristupačnim.

- Deterministička okruženja – okruženja u kojima svaka akcija stvara jedan i samo jedan željeni rezultat što znači da ne postoje dvojbe ili nesigurnosti u kojem stanju će se agent naći prilikom izvođenja pojedine akcije.
- Statička okruženja – okruženja koja se neće promijeniti s vanjskim utjecajima, već će se promjene dogoditi isključivo kao rezultat akcije agenta. S obzirom na to da će se većina realnih okruženja mijenjati ovisno o vanjskim uvjetima i događanjima, većinu njih promatramo kao dinamička okruženja.
- Diskretna okruženja – okruženja u kojima postoji konačan broj akcija i njihovih percepcija, odnosno njihovih rezultata na okruženje.

3.1.3. Klasifikacija višeagentnih sustava

Klasifikaciju višeagentnih sustava Gokulan i Srinivasan vrše na temelju više kriterija kao što su primjerice arhitektura, komunikacija i koordinacija unutar sustava. [44]



Slika 45. Opća klasifikacija višeagentnih sustava s obzirom na atribute [42]

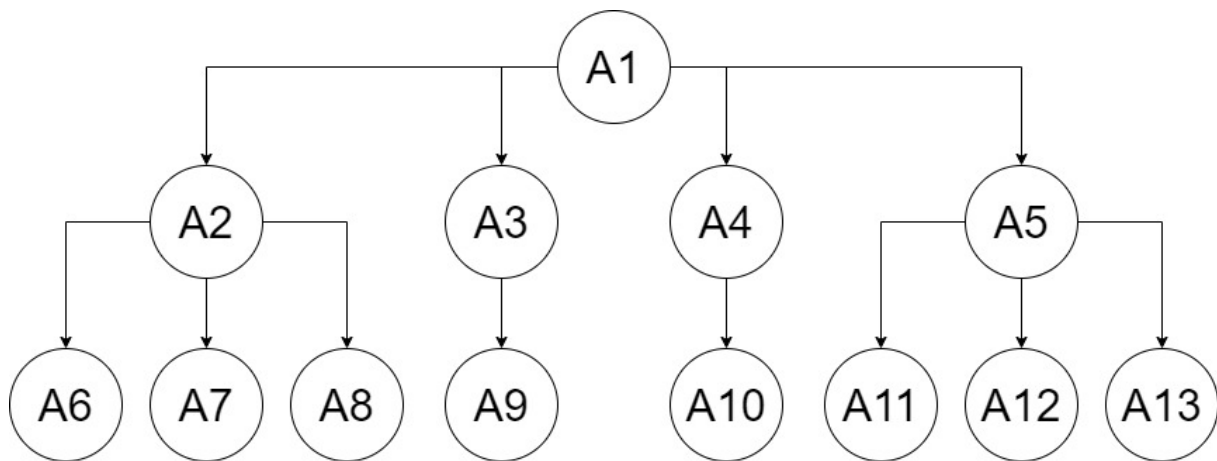
Agente prema unutarnjoj arhitekturi možemo podijeliti na:

- **Homogene strukture** – U homogenim strukturama agenti imaju iste ciljeve, senzore, stanja i akcije koje izvode, a razlikuju se po fizičkoj lokaciji i dijelu okruženja u kojem djeluju.
- **Heterogene strukture** – U heterogenim strukturama agenti se razlikuju po mogućnostima, strukturi i funkcionalnosti. Agenti se razmještaju u okolinu s obzirom na ciljeve i potrebe za dinamičnosti reakcije agenta. Heterogene strukture omogućuju modeliranje primjene agenata u realnom svijetu.

Agente prema organizaciji možemo podijeliti na:

- **Hijerarhijsku organizaciju** – Jedna od najranijih dizajna organizacije višeagentnih sustava te je uspješno primijenjena u velikom broju distribuiranih problema. Organizacija se sastoji od agenata formiranih u stablastu strukturu pri čemu agenti na različitim razinama strukture imaju različitu razinu autonomije. Agenti s niže razine šalju podatke prema višim razinama dok više razine šalju kontrolne signale nižim razinama. Prema načinu slanja i distribucije kontrole između agenata hijerarhijska organizacija može se podijeliti na jednostavnu i ujednačenu hijerarhiju.

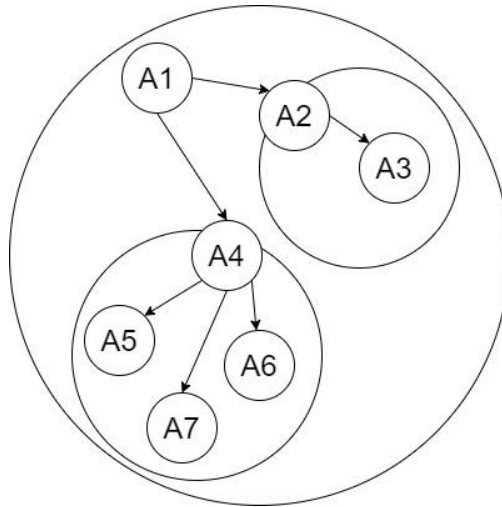
- **Jednostavna hijerarhija (engl. *Simple Hierarchy*)** – agent za donošenje odluka i kontrolu nalazi se na vrhu stabla pri čemu može nastati problem jer greška na tom agentu može prouzrokovati ispad cijelog ustava (engl. *single point of failure*).
- **Ujednačena hijerarhija (engl. *Uniform Hierarchy*)** – agenti za donošenje odluka distribuirani su na različitim razinama kako bi se povećala efikasnost i otpornost na ispade cijelog sustava. Odluke donose agenti koji imaju prikladne informacije te se odluke prosljeđuju na više razine jedino ako postoji konflikt interesa pojedinih agenata.



Slika 46. Hijerarhijska arhitektura višeagentnog sustava

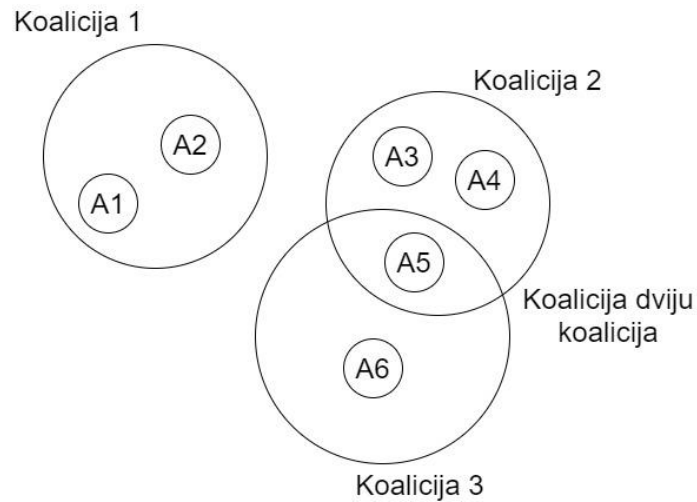
- **Organizaciju autonomnih agenata (engl. *Holonic Agent Organization*)** - „Holon” je stabilna i koherentna ili fraktalna struktura koja se sastoji od nekoliko „holona” kao i njezina podstruktura koja čini veće okruženje. Koncept „holona” predložio je Arhur Koestler kako bi objasnio društveno ponašanje bioloških bića, no njihova struktura i interakcija može se koristiti na većim modelima i organizacijskim ponašanjima unutar proizvodne i poslovne domene. U organizaciji autonomnih agenata agent se sastoji od više podagenata koji su međusobno povezani za ostvarenje jedinstvenog cilja. Podagenti nisu povezani čvrstim ograničenjima ili predefiniranim pravilima, već privrženosti zajedničkog djelovanja. Svaki „holon” odabire glavnog agenta koji komunicira s okruženjem ili drugim agentima u okruženju. Odabir se najčešće vrši na temelju dostupnosti resursa, mogućnosti komunikacije i interne strukture svakog agenta. U homogenim arhitekturama odabir se vrši nasumično ili izmjenično dok se u heterogenim arhitekturama odabir vrši na temelju sposobnosti pojedinog agenta. Agenti se također mogu grupirati u veće skupine i strukture formirajući „superholon”.

Iako struktura podsjeća na stablastu strukturu, kod organizacije autonomnih agenata moguće je izvršavati interakcije između različitih grana stabla kao i preklapanja.



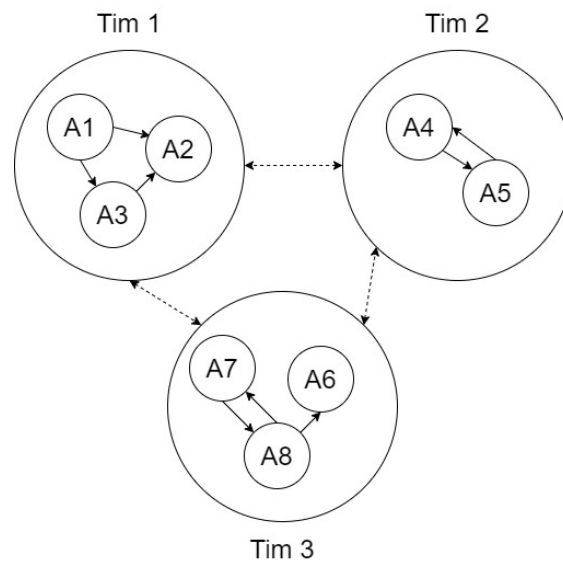
Slika 47. "Superholon"

- **Koalicije (engl. *Coalitions*)** – U arhitekturi koalicija agenti se grupiraju u određeni vremenski period kako bi povećali performanse individualnih agenata u grupi. Koalicija prestaje postojati kad se dosegne zajednički cilj. Agenti koji formiraju koaliciju mogu imati različite arhitekture te iako u originalnoj arhitekturi nemaju vodećeg agenta, on se može odabrati kao predstavnik koalicije. Preklapanje agenata unutar koalicije je dopušteno i poželjno jer se time povećava opće znanje unutar koalicije, no time se također povećava kompleksnost izračuna strategije pregovora. Koalicije su posebice kompleksne unutar dinamičkih okruženja te je potrebno reorganizirati agente kako bi se povećale performanse sustava. Teoretski gledano, najveće performanse sustava postigle bi se objedinjavanjem svih agenata u sustavu, no najčešće je takva koalicija toliko kompleksna da postaje nepraktična s pogleda ograničenja u komunikaciji i resursima.



Slika 48. Koalicija višeagentnog sustava

- **Timovi (engl. Teams)** – Organizacija agenata u timove veoma je slična koalicijama, a razlikuju se u tome da u timovima agenti rade zajednički u grupama kako bi povećali performanse grupe pri čemu agenti rijetko kad rade samostalno. Uloge i ciljevi svakog agenta unutar grupe mogu varirati, a raznolikost pridonosi većoj efikasnosti grupe. Veći timovi omogućuju bolju preglednost okruženja kao i veću količinu relevantnih informacija o okruženju. S druge strane manji timovi omogućuju brže učenje, ali su limitirani količinom informacija o okruženju koje su im dostupne. Zbog tih razloga često se radi kompromis kako bi se postigla optimalna količina informacija, broj agenata u timu i njihove mogućnosti učenja.



Slika 49. Organizacija višeagentnih sustava u timovima

3.2. Postizanje dogovora

U višeagentnim sustavima, gdje svaki agent djeluje samostalno, treba postojati mogućnost međusobnog dogovora u slučaju da dva (ili više) agenata odluče djelovati na istom području. Primjer može biti kontrola grijanja u kućanstvu s dva agenta - jedan upravlja klimatizacijskim uređajem dok drugi upravlja sustavom grijanja na plin. U slučaju da postoji potreba za grijanjem, agenti se trebaju dogovoriti hoće li grijati prostoriju pomoću električne energije ili upotrebom plina.

3.2.1. Dizajn protokola za postizanje dogovora

Svojstva protokola za postizanje dogovora prema Sandholmu su: [45]

- siguran uspjeh – protokol mora osigurati uspjeh na način da agenti eventualno postignu dogovor
- maksimiziranje dobiti zajednice – protokol mora osigurati da postignut dogovor maksimizira sumu dobiti svih pojedinih sudionika dogovora
- pareto efikasnost – rezultat dogovora možemo nazvati pareto efikasnim ako ne postoji dogovor koji bi mogao ići u korist jednog agenta, a pritom naštetiti drugom agentu koji sudjeluje u dogovoru
- individualna racionalnost – protokol se naziva racionalnim ako svaki agent radi po pravilima u najboljem interesu dionika dogovora
- stabilnost – protokol je stabilan ako potiče sve agente da se ponašaju na određen način
- jednostavnost – protokol je jednostavan ako svi dionici (agenti) korištenjem mogu odrediti optimalnu strategiju.

3.2.2. Aukcije

Aukcije, odnosno dražbe poseban su oblik prodaje javnim nadmetanjem većeg broja kupaca u svrhu postizanja što povoljnije ponude ili je robu i/ili usluge potrebno brzo prodati. Jedan su od najstarijih oblika trgovanja u međunarodnom prometu robe koji karakteriziraju tri ključna elementa: mjesto i vrijeme održavanja aukcije moraju biti javno objavljeni, predmet prodaje dobiva najbolji ponuđač, a roba koja se prodaje nije ujednačene kvalitete, nije standardizirana, ima svoju specifičnu vrijednost i nije zamjenjiva jer je svako pakiranje drugačije kvalitete pa ju je nemoguće zamijeniti za neko drugo pakiranje istovrsne robe.

Navedeni elementi aukciju čine pogodnom za prodaju predmeta koji spadaju u kategoriju primarnih proizvoda, oštećene ili zaplijenjene robe te umjetničkih predmeta. [46]

Aukcije organiziraju i provode posebne veletrgovačke tvrtke i društva koja imaju odgovarajuće objekte i opremu potrebnu za urednu, redovitu i nadziranu aukcijsku prodaju. Sudionici aukcija su prodavatelji, potencijalni kupci, prodajni i nabavni posrednici, krajnji kupci i licitator.

Postoji nekoliko vrsta aukcija. [47] **Javne aukcije** su aukcije u kojima je pristup omogućen svim zainteresiranim sudionicima. **Stalne aukcije** održavaju se redovito u određeno vrijeme i na određenom mjestu te posluju po stalnim pravilima koja reguliraju javno nadmetanje kupaca. **Povremene aukcije** održavaju se po potrebi kao posljedica nepredviđenih događaja, kao što su oštećenja u transportu, roba s greškom i sl. **Aukcije na veliko** su posebna vrsta aukcija u kojima veletrgovci prodaju veliku količinu robe, dok se **aukcije na malo** organiziraju u svrhu prodaje manje količine robe, prodaju antikviteta ili umjetničkih djela. **Sudske i prisilne aukcije** organiziraju se nakon provođenja određenih zakonskih postupaka nakon kojih je potrebno namiriti troškove ili naplatiti određeni novčani iznos.

Ovisno o vrsti robe koja se prodaje razvijeni su i različiti sustavi nadmetanja u aukcijama. Postoji **engleski sustav nadmetanja** koji je najrašireniji, a koristi se u aukcijama na malo za prodaju antikviteta, umjetničkih slika i sličnih proizvoda. Nadmetanje započinje najnižom prihvatljivom cijenom za prodavatelja, dok natjecatelj otvorenim izvikivanjem nudi veći iznos sve dok nema većih ponuda. **Nizozemski sustav nadmetanja** najčešće se koristi u aukcijama na veliko za prodaju poljoprivrednih proizvoda, ribe, cvijeća i sl. Nadmetanje započinje od jako visoke cijene koju aukcionar spušta do kad neki od natjecatelja ne prihvati. **Sustav nadmetanja sa zatvorenim pisanim ponudama** (engl. *Frist-Price Sealed-Bid Auction*) u kojem, za razliku od prethodnih dvaju, natjecatelji nude ponudu u pisanom obliku u zatvorenim kuvertama, a pobjednik je natjecatelj s najvišom ponudom. **Vickrey aukcija** (engl. *Vickrey Auction, Uniformed Second-Price Auction*) dobila je ime po dobitniku Nobelove nagrade u području ekonomskih znanosti. Ponude pristižu kuvertirane čime se osigurava tajnost visine ponuda, a pobjednik u nadmetanju je onaj s ponuđenom najvećom cijenom, ali plaća drugu po visini ponuđenu cijenu. **Aukcije s kuvertiranim ponudama** (engl. *Sealed Bid Auction*) najčešće se primjenjuju u SAD-u, a sama aukcija ima diskriminacijsko obilježje. Ponude se dostavljaju kao i kod sustava nadmetanja sa zatvorenim pisanim ponudama. Ako je predmet aukcije više istih artikala, samo prvi pobjednik plaća najvišu, a svi ostali nižu cijenu. Također postoje i **dvostruke aukcije** (engl. *Double Auction*) koje su se primjenjivale još u starom Egiptu i Mezopotamiji, a cijena se formira postupnim usuglašavanjem cijena do konačnog dogovora.

3.3. Od interneta stvari (IoT) do interneta agenata (IoA)

Proizvodi danas često nose naziv pametan. Bilo da je riječ o jednostavnoj pametnoj klupi, pametnom satu, pametnom solarnom drvetu, pametnom zaslonu za informacije u gradu ili o trgovačkom centru. Nameće se pitanje jesu li svi ovi uređaji pametni? Što je zapravo potrebno kako bi uređaj postao pametan? Promotrimo sljedeće primjere kako od interneta stvari krenuti u smjeru interneta agenata. [48]

3.3.1. Upravljanje vozilima u realnom vremenu

Internet stvari (IoT): Vozilo s GPS navigacijom koja pruža informacije o stanju prometnica i stanju prometa na cestama.

Internet agenata (IoA): Pametno vozilo koje izračunava optimalne rute kretanja vozila na cesti prema informacijama o stanju prometnica, stanju prometa na cestama i vremenskim uvjetima. Agent u vozilu može komunicirati s agentom za predviđanje stanja prometnica, agentom za predviđanje stanja prometa na cestama, agentom za predviđanje vremenskih uvjeta i na temelju podataka odlučuje o optimalnom pravcu kretanja. Također, u slučaju nesreće ili kvara na vozilu, agent može pozvati hitnu pomoć ili pomoć na cestama pružajući detaljne informacije o lokaciji, kontekstu, vremenu i opisu događaja.

3.3.2. Upravljanje prtljagom

Internet stvari (IoT): Praćenje prtljage u zračnim lukama u realnom vremenu informirajući vlasnika preko mobilne aplikacije o mjestu i vremenu dolaska prtljage.

Internet agenata (IoA): Pametno praćenje prtljage u realnom vremenu informirajući vlasnika o mjestu i vremenu dolaska prtljage te o spremnosti prtljage za preuzimanje osiguravajući prtljagu od neovlaštenog preuzimanja (krađe) ili njezinog gubitka.

3.3.3. Nadgledanje industrijske proizvodnje

Internet stvari (IoT): Nadgledanje procesa proizvodnje i praćenje fizičkih komponenti sensorima tijekom cijelog proizvodnog ciklusa.

Internet agenata (IoA): Pametan sustav s mogućnošću preporuke za održavanje kako bi se preventivno uklonili problemi u industrijskoj opremi s obzirom na stanje skladišta robe, alata, životnog vijeka dijelova za strojeve i uvjeta u proizvodnom pogonu. Također, agenti su u mogućnosti pokrenuti evakuacijske protokole u slučaju nezgoda informirajući o lokaciji, vremenu i vrsti štete na industrijskoj opremi.

3.3.4. Upravljanje navodnjavanjem

Internet stvari (IoT): Sustav navodnjava koji u realnom vremenu prati vlažnost tla i otvara cijevi za sustav navodnjavanja.

Internet agenata (IoA): Pametan sustav navodnjavanja koji mjeri vlažnost tla i komunicira s agentima za prognozu vremena. Agent zatim donosi odluku otvara li cijevi za navodnjavanje ili će iskoristiti kišu. Agenti također komuniciraju s agentom za pružanje informacija o biljkama te na temelju pojedine biljke određuju kolika količina vode je optimalna kako bi biljka imala optimalne uvjete za rast i razvoj.

3.3.5. Praćenje životinja

Internet stvari (IoT): Praćenje divljih životinja pomoću kamera i senzora za lokaciju kako bi se pratilo kretanje, trenutna pozicija i ponašanje specifične vrste životinja u njihovu prirodnom okruženju.

Internet agenata (IoA): Pametno praćenje okruženja životinja u zoološkom vrtu kako bi se otkrile promjene i anomalije u ponašanju pojedine vrste životinja. Agent može analizirati trenutno ponašanje kao što su kretanje, zvukovi, učestalost prehrane s povijesnim podacima i informirati veterinaru kako bi se pravovremeno reagiralo u slučaju bolesti.

3.3.6. Upravljanje poljoprivrednim zemljištima

Internet stvari (IoT): Automatizirano prikupljanje podataka o stanju okoliša, zemlje, gnojidbi i navodnjavanju kako bi se povećala proizvodnja poljoprivrednih proizvoda.

Internet agenata (IoA): Pametno prikupljanje podataka o zemlji, gnojidbi, vrsti zasađene biljke, meteorološkim podacima i ostalim podacima o okolišu kako bi se preporučio optimalan

način sadnje različitih kultura, navodnjavanja i gnojidbe i na taj način osiguralo povećanje proizvodnje poljoprivrednih proizvoda.

3.3.7. Pametni grad – turizam

Internet stvari (IoT): Zaslona koji pruža informacije o turističkim mjestima u gradu i ima mogućnost sortiranja s obzirom na interese pojedinog turista.

Internet agenata (IoA): Pametan turistički vodič u obliku aplikacije koji predlaže turističke atrakcije prema interesima turista uzimajući u obzir trenutnu kao i prethodne lokacije, gužve u pojedinim turističkim mjestima te meteorološke podatke izračunavajući pritom najbližu rutu. Komunikacijom s drugim agentima može rezervirati ili kupiti kartu za određenu lokaciju ili javni prijevoz.

3.3.8. Medicinske usluge

Internet stvari (IoT): Praćenje stanja pacijenata u realnom vremenu kako bi se na temelju prikupljenih podataka mogla izraditi točnija dijagnoza i identifikacija problema ili bolesti.

Internet agenata (IoA): Pametan sustav za dijagnosticiranje baziran na svim povijesnim podacima pacijenta. Sustav komunicira s agentima koji pružaju informaciju o najbližim zdravstvenim institucijama i dostupnosti specijalista u njima kao i praćenje rasporeda kad pacijent može doći na pregled ili medicinski zahvat.

4. Praktični dio

4.1. Opis aplikacijske domene i problema

Implementacija pametne naponske mreže u stambenim objektima. Sustav je u mogućnosti otkriti ispade u naponskoj mreži, tj. preopterećenje na priključenim kućanskim aparatima i ostalim uređajima u kućanstvu. Agenti u međusobnoj interakciji određuju koji uređaj prouzrokuje preopterećenje fazne linije. Uređaj koji prouzrokuje preopterećenje isključuje se iz naponske mreže pri čemu agenti ponovno priključuju ispravne uređaje na isti vod kako bi nesmetano nastavili s radom. Na temelju identificiranog tipa uređaja, cijene servisa i udaljenosti servisa (ili vremenskog trajanja) agent odlučuje koji servis kontaktirati za obavljanje popravka i ugovoriti najbrži mogući termin servisa. Podaci o servisnim centrima dostupni su preko web servisa koji podatke uzima iz baze podataka. Baza podataka sadrži podatke o servisnim centrima koji su ovlaštene za popravak pojedinog tipa uređaja i imaju sklopljen ugovor o servisiranju.

4.2. Korištene tehnologije

- **Raspberry Pi/Zero** [49] – Agenti koji međusobnom komunikacijom određuju kako postupiti prilikom preopterećenja te jedan od agenata ugovara uslugu popravka uređaja. Agenti koji određuju kako upravljati uređajem povezani su s Arduino uređajem preko serijskog porta.
- **SPADE** [50] (*Smart Python Agent Development Environment*) je razvojno okruženje za programiranje pametnih agenata koji mogu komunicirati s drugim agentima ili ljudima preko poruka (XMPP) [51]. Osnovni koncepti agenta izrađeni su pomoću komunikacijske XMPP porukama i definicijom različitih ponašanja. Za ostvarivanje komunikacije svaki agent ima jedinstveni JID (Jabber ID) oznaku i lozinku kojima se registrira na XMPP poslužitelju. Veza ostaje otvorena sve dok agent radi. SPADE nudi nekoliko preddefiniranih tipova ponašanja: ponavljajuće, neponavljajuće, periodičko, zakazano i ponašanje konačnog automata. Ponašanje je zadatak (kod) koji agent izvršava. Agent može izvršavati više ponašanja istovremeno dok dispečer poruka brine o dostavi poruka u ispravan red čekanja za pojedino ponašanje. Kako bi dispečer mogao odrediti koja poruka pripada kojem ponašanju, svako ponašanje ima predložak poruke.

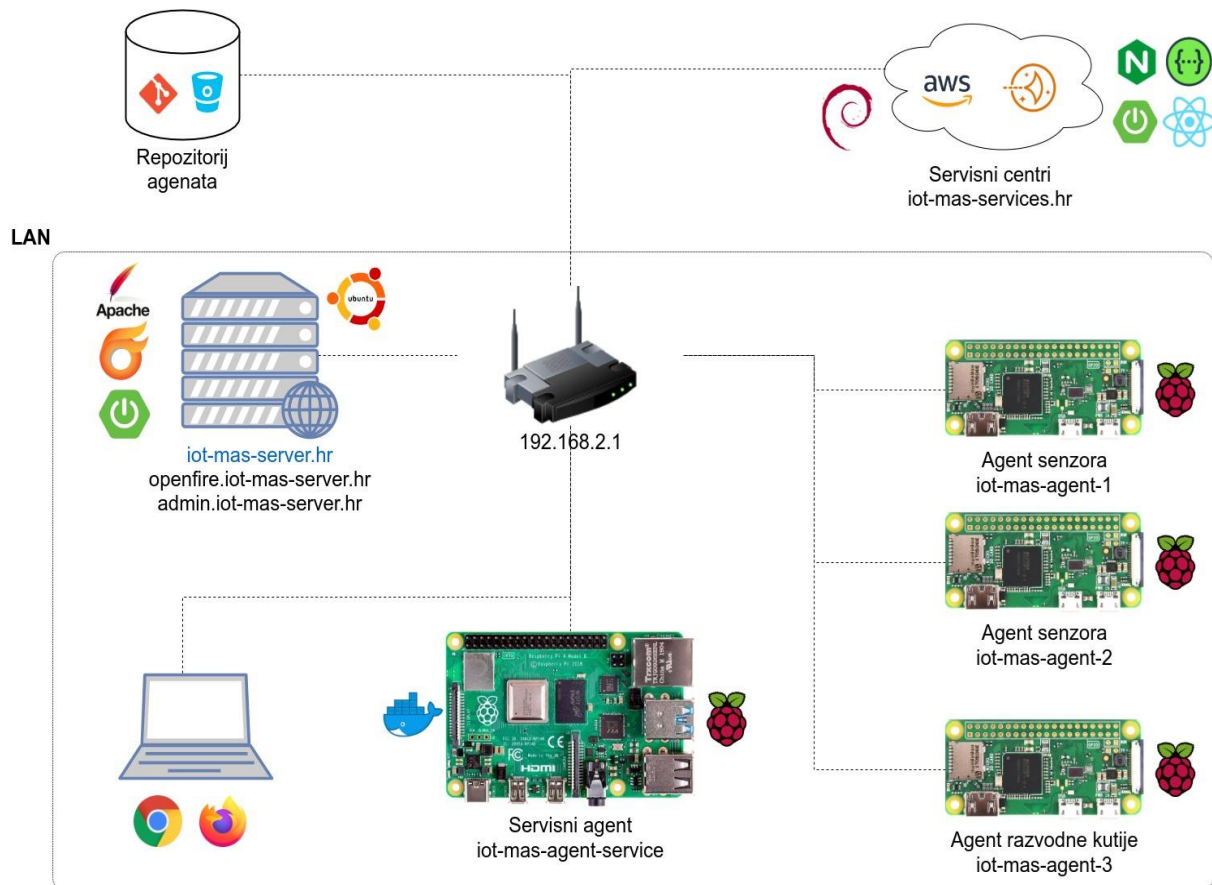
- **OpenFire** [52] – XMPP poslužitelj instaliran na kućnom poslužitelju kako bi omogućio komunikaciju agenata.
- **Spring boot** [53] (**Java**, **HTML5**, **JS - jQuery** [54], **CSS – Bootstrap** [55], **React** [56]) – dvije aplikacije izrađene Spring Boot razvojnim okruženjem. Jedna pruža mogućnost upravljanja agentima, preuzimajući Python skripte iz git repozitorija i pokreće skripte na definiranim agentima, a instalirana je na kućnom poslužitelju. Druga aplikacija (instalirana na AWS Cloud) pruža informacije o servisnim centrima, cijenama usluga, lokaciji i potrebnom vremenu za odrađivanje pojedinih popravaka.
- **Docker** [57] – Agent za ugovaranje servisa koristi Docker kontajnere kako bi omogućio izolirano okruženje za izvođenje agenata za nadmetanje. Agenti u nadmetanju koriste sustav nadmetanja za zatvorenim ponudama (engl. *Frist-Price Sealed-Bid Auction*).

4.3. Implementacija

Sustav se sastoji od dviju web aplikacija, lokalnog web služitelja, git repozitorija, jednog Raspberry Pi računala te proizvoljnog broja Raspberry Pi Zero računala. Upravljanje sustavom vrši se pomoću internetskog preglednika. Web aplikacije izrađene su pomoću Spring Boot razvojnog okruženja te pružaju korisničko sučelje i REST API za komunikaciju s ostalim dionicima sustava.

Admin web aplikacija (*admin.iot-mas-server.hr*) postavljena je na lokalni web poslužitelj koji koristi Linux Ubuntu [58] operativni sustav i Apache aplikacijski web poslužitelj. Aplikacija omogućuje korisniku dodavanje agenata na Raspberry Pi uređaje povezane na lokalnu mrežu. Svakom Raspberry Pi uređaju dodjeljuje se mrežno ime prilikom konfiguracije što omogućuje spajanje na uređaj prilikom promjene IP adrese koju dodjeljuje DHCP poslužitelj. Pomoću aplikacije korisnik dodaje podatke za povezivanje SSH protokolom, dodaju se podaci potrebni za spajanje XMPP protokolom kao i podaci koji omogućuju lakše prepoznavanje i opis zaduženja agenta. Nakon unosa podataka korisnik ima mogućnost pokretanja i zaustavljanja agenata na pojedinom uređaju kao i spajanja na web sučelje svakog pojedinog agenta. Aplikacijski kod agenta nalazi se u git repozitoriju te korisnik u padajućem izborniku u sučelju može odabrati kojeg agenta želi postaviti na određeni uređaj. Prilikom pokretanja agenta aplikacija preko SSH protokola kopira Python datoteke na željeni uređaj i pokreće izvršavanje. Korisnik pomoću zelene i crvene ikone u svakom trenutku može vidjeti izvršava li se agent na pojedinom uređaju. Za ostvarivanje funkcionalnosti korišten je Vanilla Javascript, dok bi za realizaciju produkcijskog sučelja bilo poželjno koristiti razvojno okruženje bazirano na web

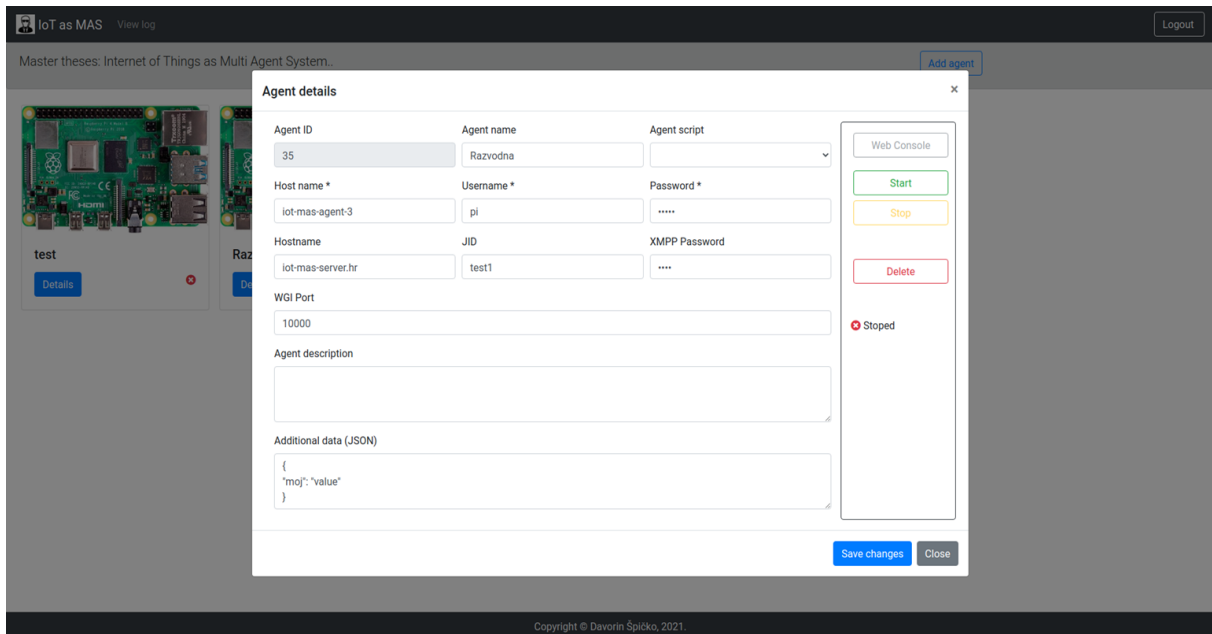
utičnicama (engl. *web sockets*) što bi omogućilo komunikaciju s poslužiteljem u realnom vremenu. Primjer razvojnog okruženja je socket.io [59]. Socket.io omogućuje dvosmjernu komunikaciju u realnom vremenu baziranu na komunikaciji vođenoj događajem (engl. *event-based communication*).



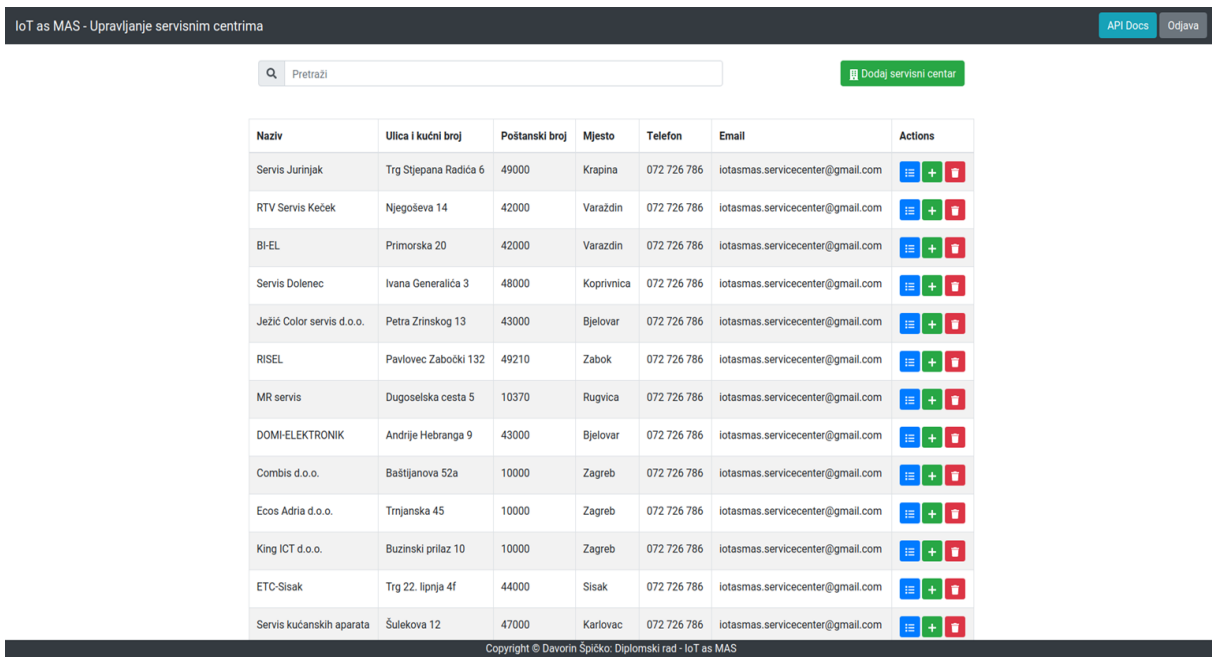
Slika 50. Arhitektura sustava

Druga web aplikacija (*iot-mas-services.hr*) također je izrađena pomoću Spring Boot razvojnog okruženja. Web aplikacija postavljena je u AWS Lightsail [60] virtualni privatni poslužitelj koji koristi Linux Debian [61] operativni sustav i Nginx [62] aplikacijski poslužitelj. Korisnički dio izrađen je pomoću React biblioteke koja omogućuje bolje korisničko iskustvo za dodavanje i filtriranje servisnih centara i pretraživanje usluga servisa. Komunikacija s ostalim dionicima sustava omogućena je pomoću REST servisa s prijenosom podataka u JSON obliku. REST API opisan je pomoću Swagger OpenAPI [63] specifikacije što omogućuje preglednu dokumentaciju za korisnike REST servisa. Aplikacija omogućava dodavanje servisnih centara i usluga servisa koje pojedini servisni centar pruža, pri čemu se definira adresa koja služi za izračunavanje udaljenosti, cijene usluge servisa kao i trajanje servisa u minutama. Ovi osnovni

podaci služe za izračun konačne cijene servisa, odnosno težine koju agent aukcije koristi prilikom zastupanja servisa na natječaju za popravak uređaja.



Slika 51. Web aplikacija - upravljanje agentima



Slika 52. Web aplikacija - Upravljanje servisnim centrima i uslugama

Na lokalnom web poslužitelju nalazi se i aplikacijski server OpenFire (*openfire.iot-mas-server.hr*) koji omogućuje registraciju i komunikaciju agenata preko XMPP protokola. Prilikom

kreiranja agenta u administracijskom sučelju korisnik može odabrati koje korisničko ime i lozinku koristi agent za spajanje na XMPP poslužitelj. Lokalni XMPP poslužitelj omogućuje komunikaciju agenata na lokalnoj mreži čime se uklanja nemogućnost komunikacije u slučaju prekida internetske veze, dok je latencija manja i osigurava se brža komunikacija. Razvoj 5G mreža uklonit će neke od nedostataka trenutne tehnologije te osigurati da uređaji povezani na internet ostvare najbolje performanse.

The screenshot shows the OpenFire web interface. At the top, there is a navigation bar with 'Server', 'Users/Groups', 'Sessions', 'Group Chat', and 'Plugins'. Below this, there is a 'User Summary' section. The table below shows a list of users with the following columns: Online, Username, Name, Groups, Created, Last Logout, Edit, and Delete.

Online	Username	Name	Groups	Created	Last Logout	Edit	Delete
1	admin	Administrator	None	Aug 18, 2020	Never logged in before.		
2	agent-bidder-1		None	Mar 14, 2021	2 hours, 3 minutes		
3	agent-bidder-10		None	Mar 14, 2021	2 hours, 3 minutes		
4	agent-bidder-11		None	Mar 14, 2021	2 hours, 3 minutes		
5	agent-bidder-12		None	Mar 14, 2021	2 hours, 3 minutes		
6	agent-bidder-13		None	Mar 14, 2021	2 hours, 3 minutes		
7	agent-bidder-14		None	Mar 14, 2021	2 hours, 3 minutes		
8	agent-bidder-15		None	Mar 14, 2021	2 hours, 3 minutes		
9	agent-bidder-2		None	Mar 14, 2021	2 hours, 3 minutes		
10	agent-bidder-3		None	Mar 14, 2021	2 hours, 3 minutes		
11	agent-bidder-4		None	Mar 14, 2021	2 hours, 3 minutes		
12	agent-bidder-5		None	Mar 14, 2021	2 hours, 3 minutes		
13	agent-bidder-6		None	Mar 14, 2021	2 hours, 3 minutes		
14	agent-bidder-7		None	Mar 14, 2021	2 hours, 3 minutes		
15	agent-bidder-8		None	Mar 14, 2021	2 hours, 3 minutes		
16	agent-bidder-9		None	Mar 14, 2021	2 hours, 3 minutes		
17	agent-bidder-mycentre		None	Mar 14, 2021	2 hours, 37 minutes		
18	agent-razvodne-kudje		None	Mar 14, 2021	2 hours, 0 minutes		
19	mailman		None	Mar 16, 2021	1 hour, 54 minutes		
20	servisni-agent		None	Mar 14, 2021	2 hours, 0 minutes		
21	test		None	Mar 16, 2021	4 hours, 34 minutes		
22	test-sensor-001		None	Mar 14, 2021	2 hours, 3 minutes		

Slika 53. OpenFire - XMPP poslužitelj

Raspberry Pi uređaji koriste sliku sustava (engl. *system image*) koja sadrži predinstaliranu programsku podršku i zavisnosti (engl. *dependencies*). Slika sadrži konfiguriran Raspberry Pi OS [64], Python, Pip i te je instaliran SPADE kako bi se omogućilo izvršavanje agenata. Prilikom konfiguracije novog uređaja potrebno je snimiti sliku na memorijsku karticu uređaja te prije spajanja na mrežu definirati pristupne podatke za spajanje na mrežu kao i mrežno ime koje će se koristiti za spajanje na uređaj neovisno o dodijeljenoj IP adresi. Također je potrebno urediti `/etc/hosts` datoteku gdje će se postaviti mrežna imena ostalih dionika sustava, primjerice OpenFire poslužitelja, ili podesiti DNS poslužitelj. Uređaji koriste mrežna imena (`iot-mas-agent-n`) gdje n označava redni broj uređaja.

4.3.1. Agenti

Kako bi pojednostavnili razvoj programskog koda agenata kao i njegovu distribuciju na pojedine uređaje, koristi se sustav verzioniranja Git [65], dok je kod smješten u repozitorij Bitbucket [66]. Prilikom dodavanja novog agenta u sustav ili njegova uređivanja, web aplikacija dohvaća sve agente iz git repozitorija te kopira odabrani kod agenta na željeni uređaj. Aplikacija pokreće agenta izvršavanjem naredbe preko SSH protokola pri čemu se kod agenta pokreće u pozadini čime ne blokira izvršavanje naredbi za provjeru je li agent aktivan. Uz naziv Python datoteke šalju se i podaci za svakog agenta kao prvi parametar komandne linije. Podaci se predaju u JSON obliku koji je pretvoren u base64 oblik. Svaki agent prilikom pokretanja čita podatke, pretvara ih u JSON te ih pretvara u Python objekt kako bi se pojednostavnilo korištenje podataka kod svakog agenta.

```
1 import base64
2 import json
3
4 class XmppData:
5     def __init__(self, hostname, jid, password) -> None:
6         super().__init__()
7         self.hostname = hostname
8         self.jid = f"{jid}@{hostname}"
9         self.password = password
10
11
12 class AgentData:
13     def __init__(self, data) -> None:
14         super().__init__()
15         self.readDataFromParameters(data)
16
17     def readDataFromParameters(self, receivedData):
18         receivedData_bytes = receivedData.encode('utf-8')
19         message_bytes = base64.b64decode(receivedData_bytes)
20         message = message_bytes.decode('utf-8')
21         data = json.loads(message)
22
23         self.name = data["name"]
24         self.xmpp = XmppData(data["xmpp"]["hostname"], data["xmpp"]["jid"], data["xmpp"]["password"])
25         try:
26             self.wgiPort = data["wgiPort"]
27         except KeyError:
28             self.wgiPort = 10880
29         try:
30             self.data = data["data"]
31         except KeyError:
32             self.data = {}
33
34     def __str__(self) -> str:
35         return f"Name: {self.name} | WGI Port: {self.wgiPort} | XMPP Hostname: {self.xmpp.hostname} | XMPP JID: {self.xmpp.jid} | XMPP password: {self.xmpp.password} | Data: {self.data}"
36
```

Slika 54. Klasa podataka agenata

Obavezni podaci za svakog agenta su naziv agenta, XMPP podaci (naziv poslužitelja, korisničko ime – JID, lozinka). Ako WGI port nije definiran, koristi se zadana vrijednost. Agentu je također moguće poslati proizvoljan set podataka koji se pretvara u Python rječnik i podaci se mogu koristiti za ostvarivanje funkcionalnosti agenata. Primjer takvih podataka može biti zadano vrijeme isteka čekanja poruka, JID podaci ostalih agenata, proizvoljne poruke i sl. Proizvoljni podaci mogu se definirati kao JSON objekt u admin sučelju za svaki agent zasebno.

```

1  {
2      "name": "Test sensor 001",
3      "wgiPort": 10001,
4      "xmpp": {
5          "hostname": "openfire.iot-mas-server.hr",
6          "jid": "test-sensor-001",
7          "password": "0000"
8      },
9      "data": {
10         "DEVICE": "TV",
11         "JUNCTION_BOX_AGENT_JID": "agent-razvodne-kutije",
12         "SERVICE_AGENT_JID": "servisni-agent"
13     }
14 }
15

```

Slika 55. Primjer podataka koji se šalju agentu

Kako bi se olakšalo praćenje uređaja na mreži, agenti bilježe svaku akciju u centralni dnevnik. Pri tome se bilježi vrijeme unosa u dnevnik, naziv agenta i akcija koju je agent proveo. Zapisi se izvršavaju izvođenjem HTTP POST metode na REST API admin web aplikacije.

```

1  import requests
2      from urllib.parse import quote
3      from requests.auth import HTTPBasicAuth
4
5  class WebAppAPI():
6      WEB_APP_URL = "http://admin.iot-mas-server.hr"
7      EMAIL_URL = f"{WEB_APP_URL}/api/v1/email"
8      LOG_URL = f"{WEB_APP_URL}/api/v1/log"
9
10     AUTH = HTTPBasicAuth("admin", " ")
11     agentName = None
12
13     def __init__(self, agentName) -> None:
14         super().__init__()
15         self.agentName = agentName
16
17     def log(self, message):
18         requests.post(self.LOG_URL, data={"agentName": self.agentName, "message": message}, auth=self.AUTH)
19

```

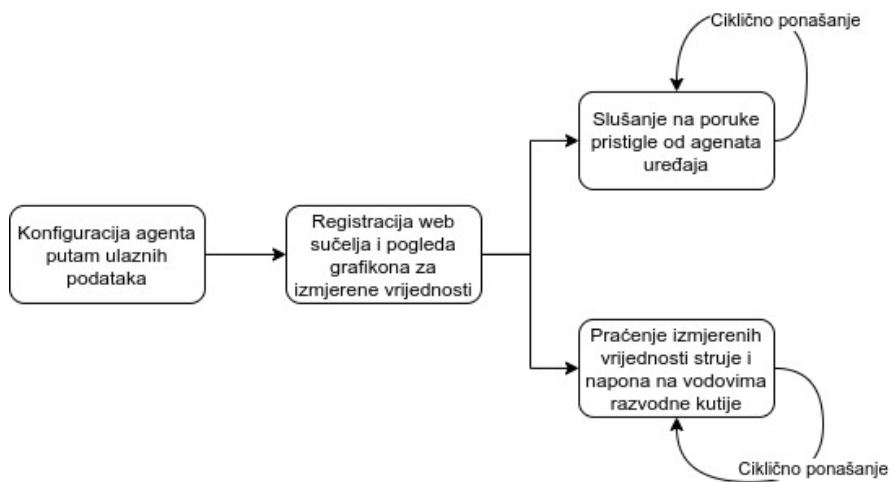
Slika 56. Pomoćna klasa za spremanje akcija agenta u dnevnik rada

Sustav se sastoji od četiriju ili više agenata. Agent razvodne kutije zadužen je za provjeru stanja mreže u kući, servisni agent zadužen je za informiranje servisnih centara i održavanje aukcija, agent aukcije zastupa servisni centar prilikom provođenja aukcija, dok agenata uređaja može biti onoliko koliko to dopušta adresni prostor lokalne mreže, jedini uvjet

je da svaki agent dobije jedinstveno mrežno ime i da se podesi kako bi se mogao spojiti na bežičnu mrežu.

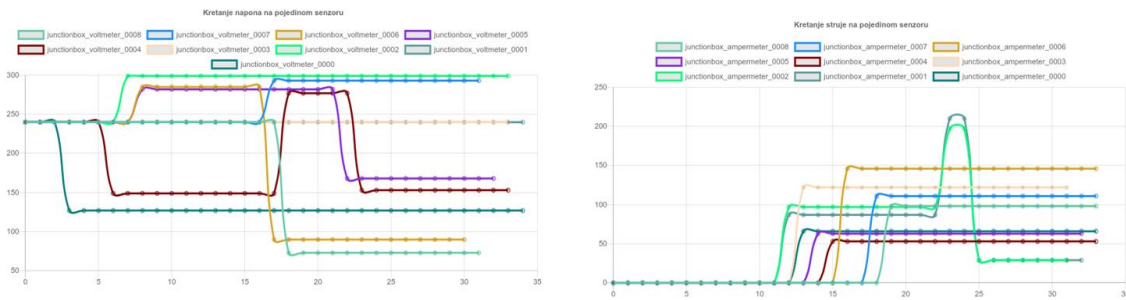
4.3.1.1. Agent razvodne kutije

Agent razvodne kutije je agent koji prati stanje na razvodnoj kutiji, tj. zadužen je za cjelokupnu mrežu unutar kuće. Agent upravlja relejima preko GIO izlaza, dok za određivanja nepravilnosti koristi ampermetar i voltmetar. Ako postoji preopterećenje na pojedinom električnom vodu, agent razvodne kutije isključuje cijeli vod. To će rezultirati nestankom struje na pojedinom uređaju o čemu brine agent uređaja.



Slika 57. Agent razvodne kutije - Ponašanja

Agent preko base64 JSON objekta zaprima podatke od admin web aplikacije. Pomoću zaprimljenih podataka agent registrira grafičko web sučelje na zadanom portu (engl. *WGI – Web Graphical Interface* [67]). Agent uz standardni pogled, u kojem je moguće pratiti stanja i zaprimljene poruke, ima i dodatni pogled u kojem je moguće vidjeti kretanje napona i struje na pojedinim električnim vodovima. Agent ima definirano ciklično ponašanje u kojem mjeri vrijednosti napona i struje tijekom vremena i vrijednosti sprema u lokalnu bazu podataka. Baza podataka je SQLite [68] baza podataka koja sadrži jednostavne podatke vrijednosti napona i struje u određenom trenutku. Spade koristi MVC uzorak dizajna pri čemu se vrijednosti čitaju iz baze pomoću kontrolera te se predaju pogledu koji pomoću definiranog seta podataka i Jinja2 [69] jezika za predloške (engl. *template engine*) generira JSON podatke te se pomoću Chart.js [70] biblioteke prikazuje grafikon kretanja vrijednosti napona i struje u linijskom grafikonu.



Slika 58. Prikaz kretanja vrijednosti napona i struje koje prati agent razvodne kutije

Kako bi se osigurala komunikacija s agentima uređaja, agent u drugom cikličkom ponašanju sluša na poruke koje imaju zadani predložak i šalje povratnu informaciju o stanju mreže. Agent uređaja na temelju odgovora može odrediti daljnje korake.

```

57 class JunctionBoxAgent(Agent):
58     async def setup(self):
59         webAppAPI.log(f"Registered agent: {agentData}")
60         print(f"Registered agent: {agentData}")
61         checkFromDeviceTemplate = Template()
62         checkFromDeviceTemplate.set_metadata("performative", "verification")
63         checkFromDeviceTemplate.set_metadata("ontology", "device")
64         self.add_behaviour(self.MessageListeningBehaviour(), template=checkFromDeviceTemplate)
65         self.add_behaviour(self.LogValuesToLocalDatabaseBehaviour())
66
67     async def displayChart(self, request):
68         db = SensorValueStorage(DB_NAME)
69         voltageData = db.fetchVoltageGraphData()
70         currentData = db.fetchCurrentGraphData()
71         return {"voltageGraphData": voltageData, "currentGraphData": currentData}
72
73     class MessageListeningBehaviour(CyclicBehaviour):
74         async def run(self):
75             print("Čekam poruku...")
76             message = await self.receive(timeout=DEFAULT_TIMEOUT)
77             if message:
78                 reply = Message(to=str(message.sender))
79                 reply.set_metadata("performative", "confirmation")
80                 reply.set_metadata("ontology", "device")
81
82                 status = True
83                 for voltmeter in voltmeters:
84                     voltage = voltmeter.getValue()
85                     status = status and voltage > 200 and voltage < 250
86
87                 reply.body = "OK" if status else "ERROR"
88                 await self.send(reply)
89             else:
90                 print(f"Nije bilo upita u posljednjih {DateTimeUtils.secondsToHumanTime(DEFAULT_TIMEOUT)}")
91

```

Slika 59. Isječak koda - Ciklično ponašanje za čekanje poruka od strane agentata uređaja

4.3.1.2. Agent uređaja

Agent uređaja baziran je na stroju s konačnim brojem stanja (engl. *FSM – Finite State Machine*). Definirana stanja stroja su:

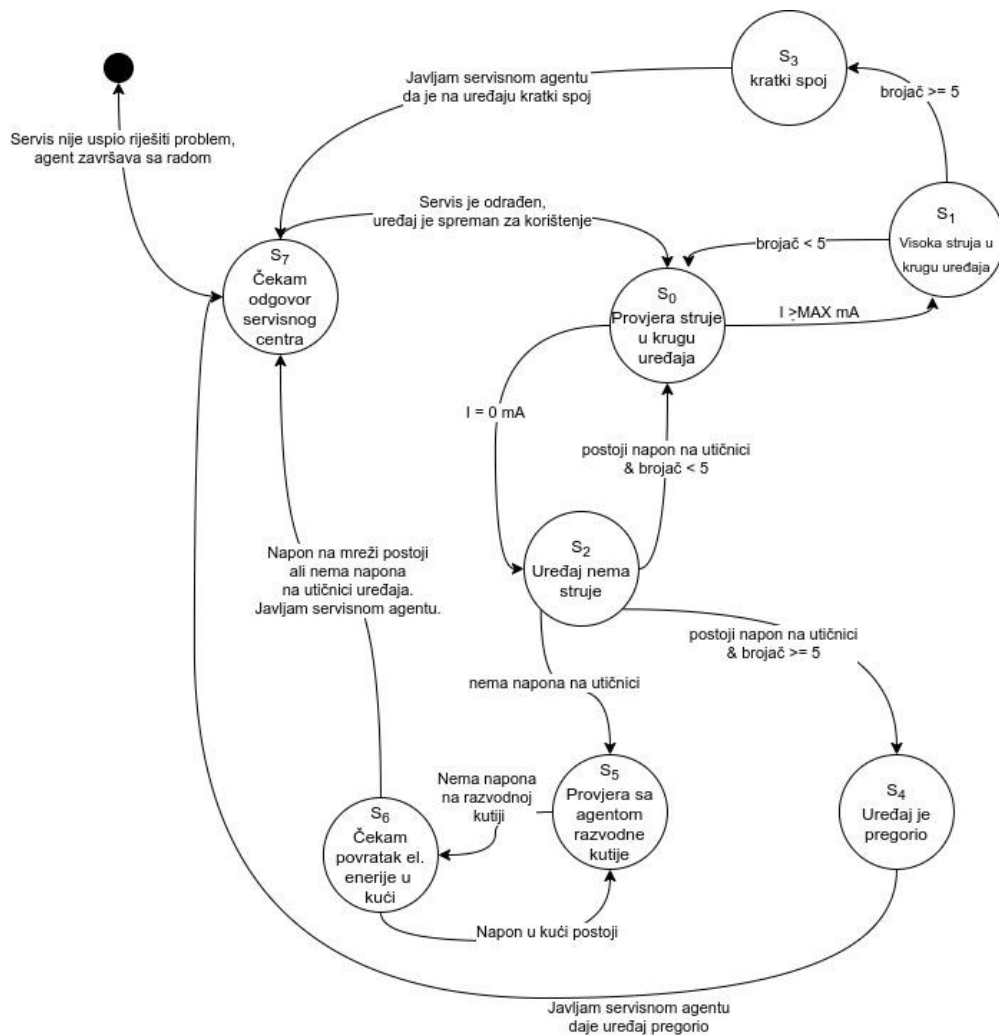
- S_0 – Početno i očekivano stanje agenta
- S_1 – Stanje visoke struje u krugu uređaja
- S_2 – U krugu uređaja nema struje
- S_3 – Kroz uređaj teče prevelika struja, mogućnost kratkog spoja
- S_4 – Uređaj je pregorio
- S_5 – Provjera stanja kućne mreže sa agentom razvodne kutije
- S_6 – Čekanje na povratak električne energije u kući
- S_7 – Čeka se akcija servisnog centra.

Agent provjerava vrijednosti napona i struje na uređaju. Prilikom nesmetanog rada agent će u ovom stanju provesti najviše vremena. Kao i kod agenta razvodne kutije, agent bilježi kretanja vrijednosti napona i struje na uređaju te ih sprema u lokalnu SQLite bazu podataka i prikazuje ih u obliku linijskog grafikona.

U slučaju da agent na temelju vrijednosti s ampermetra utvrdi da je struja u strujnom krugu (krugu uređaja) viša od maksimalnog iznosa struje podržanog za pojedini uređaj, agent prelazi u stanje u kojem isključuje uređaj iz strujnog kruga te određuje sljedeći korak. Isključivanje uređaja izvršava se bez aktivnosti koje zahtijevaju mrežnu komunikaciju kako bi se osigurala što brža reakcija i preveniralo oštećenje uređaja. Ako je broj provjera manji od definiranog broja pokušaja, agent će ponovno prijeći u stanje S_0 kako bi potvrdio da nije došlo do anomalije. Ako je vrijednost struje u krugu uređaja viša od maksimalne nakon definiranog broja pokušaja, agent prelazi u stanje S_3 u kojem obavještava servisnog agenta o potrebnoj aktivnosti koju treba provesti. Nakon obavještavanja agent čeka na odgovor servisnog agenta o statusu servisa.

S druge strane, kad agent utvrdi da strujnim krugom uređaja ne teče struja, agent prelazi u stanje S_2 . Ako je broj pokušaja manji od definiranog broja pokušaja, agent se pokušava vratiti u stanje S_0 . U slučaju da je broj pokušaja veći od zadanog broja pokušaja i postoji napon na utičnici, agent pretpostavlja da je uređaj pregorio te prelazi u stanje S_4 u kojem informira servisnog agenta o nastaloj situaciji. U slučaju da ne postoji napon na utičnici, agent mora komunicirati s agentom razvodne kutije kako bi utvrdio je li nedostatak napona samo na toj utičnici ili napona nema na cijelom vodu. Ako napona nema na razini cijelog voda, agent čeka povratak električne energije u intervalima od jedne minute. Ako agent razvodne kutije potvrdi da napon na glavnom vodu postoji, to znači da je agent ustanovio problem s

uređajem ili utičnicom (linijom) na koji je uređaj spojen. Obavještava servisnog agenta o nastaloj situaciji i prelazi u stanje S_7 u kojem čeka odgovor kako bi utvrdio nastavlja li s radom.



Slika 60. Stanja konačnog automata agenta uređaja

Stanja agenata su klase koje proširuju klasu State. Prilikom definiranja ponašanja agenta konačnog automata stanja je potrebno dodati u ponašanje, definirati početno ponašanje kao i sve moguće prijelaze koji definiraju ponašanje agenta i odnos stanja. Prijelazi se definiraju metodom koja prima dva parametra, trenutno stanje i sljedeće stanje. Bitno je također označiti i stanja koja imaju prijelaze sama u sebe jer će u suprotnom agent završiti s radom te nećemo dobiti očekivano ponašanje. Komunikacija s agentima razvodne kutije i servisnim agentom vrši se preko predefiniranih predložaka poruka na temelju kojih navedeni agenti mogu odrediti koje ponašanje pokrenuti kad pristigne poruka koja zadovoljava zadani predložak.

```

86 class StateInitial(State):
87     async def run(self):
88         webAppAPI.log("Senzor je u stanju S0: Provjera struje u krugu uređaja")
89         print("Senzor je u stanju S0: Provjera struje u krugu uređaja")
90         # TODO: enable relay
91         voltage = voltmeterMain.getValue()
92         if voltage >= 0:
93             storage.addVoltage(voltmeterMain.name, voltage) # Storing value to local DB to display it
94
95         current = ampermeterMain.getValue() # Mocking value from sensor
96         if current >= 0: # If sensor error occurs
97             storage.addCurrent(ampermeterMain.name, current) # Storing value to local DB to display it
98             if current > DEVICE_MAX_CURRENT:
99                 self.set_next_state(S1)
100             elif current == 0:
101                 self.set_next_state(S2)
102             else:
103                 self.set_next_state(S0)
104                 time.sleep(1)

```

Slika 61. Početno i željeno stanje agenta uređaja

```

234 class SensorAgent(Agent):
235     counterNoCurrent = 0
236     counterHighCurrent = 0
237
238     async def setup(self):
239         fsm = SensorFSMBehaviour()
240         fsm.add_state(name=S0, state=StateInitial(), initial=True)
241         fsm.add_state(name=S1, state=StateHighCurrent())
242         fsm.add_state(name=S2, state=StateNoCurrent())
243         fsm.add_state(name=S3, state=StateShortCircuit())
244         fsm.add_state(name=S4, state=StateBurnedDevice())
245         fsm.add_state(name=S5, state=StateVerifyWithJunctionBox())
246         fsm.add_state(name=S6, state=StateWaitForPower())
247         fsm.add_state(name=S7, state=StateWaitForErrorToBeResolved())
248         fsm.add_transition(S0, S0)
249         fsm.add_transition(S0, S1)
250         fsm.add_transition(S0, S2)
251
252         fsm.add_transition(S1, S0)
253         fsm.add_transition(S1, S3)
254
255         fsm.add_transition(S2, S0)
256         fsm.add_transition(S2, S4)
257         fsm.add_transition(S2, S5)

```

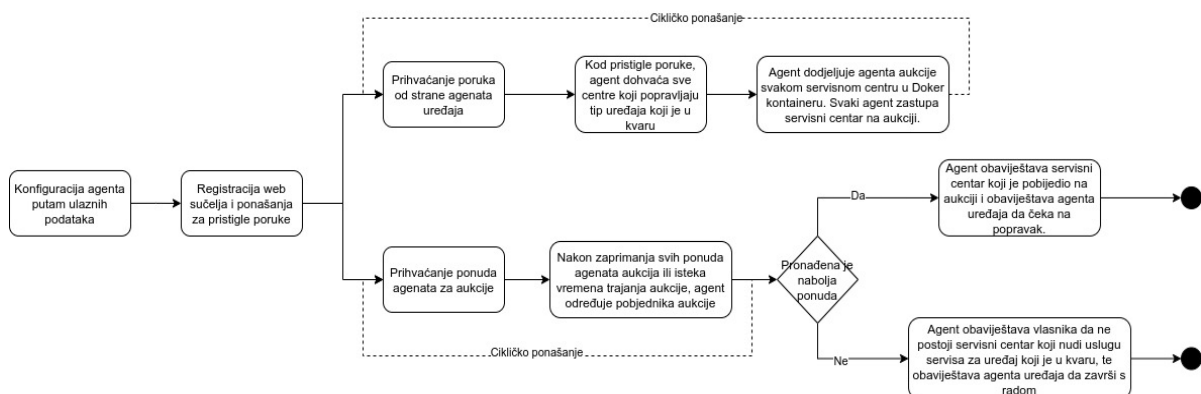
Slika 62. Definiranje stanja i prijelaza u konačnom automatu

4.3.1.3. Servisni agent

Servisni agent definiran je pomoću triju ponašanja: dvaju cikličkih ponašanja koja čekaju na poruke koje pristižu od agenata uređaja i agenata aukcija te jednokratnog ponašanja koje utvrđuje rezultat aukcije. Agent prima podatke iz admin web aplikacije, registrira se na temelju XMPP podataka te započinje ciklička ponašanja prihvaćanja poruka. Ponašanja čekaju

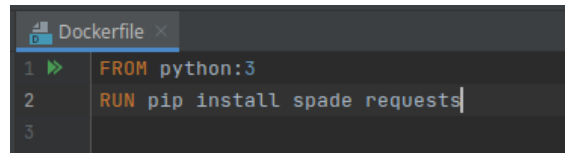
na poruke određenog predloška i na temelju pristigle poruke određuju daljnje korake. Kad je zaprimljena poruka agenta uređaja, servisni agent preko REST API dohvaća sve servisne centre koji pružaju uslugu popravka za vrstu uređaja na kojem je nastao problem. Servisni centri koji su dodani u web aplikaciju označavaju servisne centre koji imaju potpisan ugovor i prihvaćaju uvjete korištenja posrednika u dogovaranju poslova. Servisni centri mogu ažurirati svoje usluge, cijene i trajanje izvršenja usluga kako bi postigli što konkurentniju poziciju za pobjedu na aukciji. Na temelju dohvaćenog popisa servisnih centara servisni agent svakom servisnom centru dodjeljuje agenta aukcije koji djeluje u ime servisnog centra nastojeći pobijediti u aukciji čime bi servisni centar osigurao posao. Nakon dodjeljivanja agenata aukcije servisni agent prihvaća odgovore (ponude) svih agenata koji sudjeluju u aukciji. Aukcija ima vrijeme trajanja pa nakon isteka vremena trajanja aukcije ili pristizanja ponuda svih agenata aukcija servisni agent zaključuje aukciju i pokreće određivanje pobjednika aukcije. Aukcija se provodi na temelju sustava nadmetanja sa zatvorenim pisanim ponudama pri čemu najniža ponuda pobjeđuje. Najniža ponuda označava najmanji trošak i najkraće trajanje izvršenja usluge popravka. Nakon završetka aukcije servisni agent obavještava servisni centar da je ponuda prihvaćena i obavještava agenta uređaja da čeka na popravak. Ako aukcija nije bila uspješna, servisni agent obavještava agenta uređaja da završi s radom te obavještava vlasnika da postoji problem koji zahtijeva njegovu pozornost. Aukcija može biti neuspješna ako u sustavu ne postoji servisni centar koji vrši uslugu popravka pojedinog uređaja ili ne pristigne ni jedna ponuda za vrijeme trajanja aukcije.

Određivanje pobjednika aukcije jednokratno je ponašanje koje završava s radom nakon uspješnog ili neuspješnog utvrđivanja pobjednika aukcije. Servisni agent nastavlja s radom i čeka na pristizanje poruka od agenata uređaja.



Slika 63. Servisni agent - Ponašanja

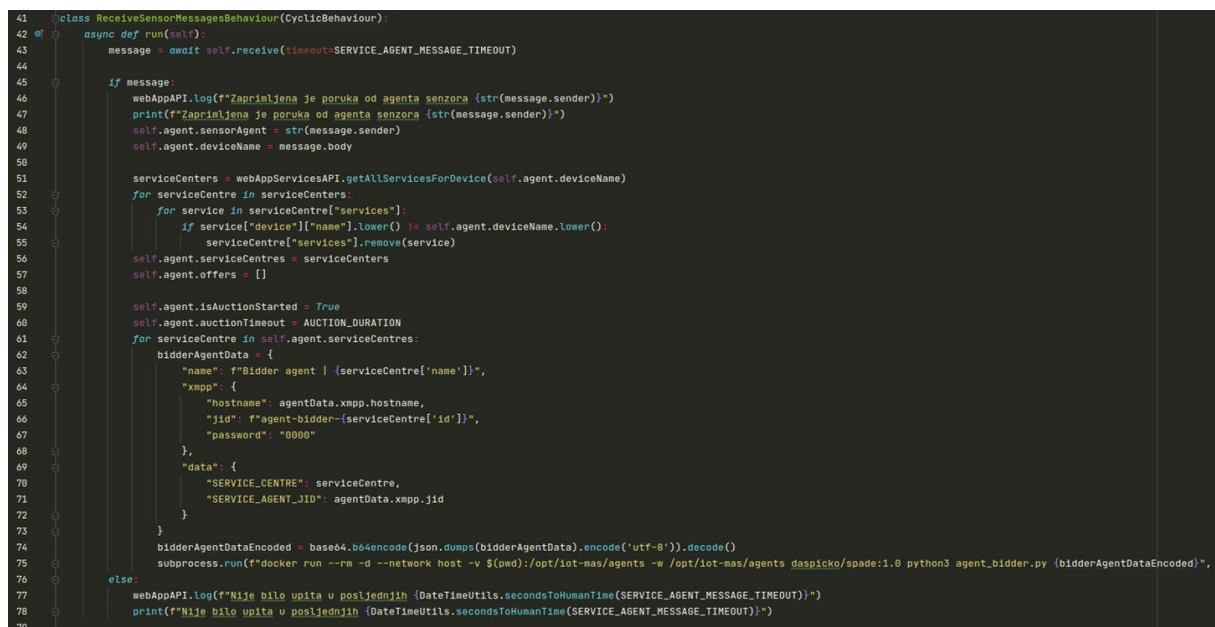
Kako bi se osiguralo paralelno izvođenje agenata aukcije u izoliranom okruženju, za pokretanje agenata koriste se Docker kontejneri. Docker osigurava izolirano programsko okruženje u kojem se virtualizira samo softversko okruženje, bez potrebe za virtualizacijom operativnog sustava. Docker slika izrađena je iz osnovne slike za Python i instaliranog programskog okruženja koje osigurava izvršavanje SPADE agenata.



```
Dockerfile
1 FROM python:3
2 RUN pip install spade requests
3
```

Slika 64. Docker datoteka za kreiranje slike kontejnera

Kontejneri se pokreću pomoću subprocess Python modula čime se osigurava pokretanje kontejnera u zasebnim procesima. Docker kontejneri izvršavaju se opcijom dostupnom samo za Linux operativne sustave kod čega se postavke spajanja na mreže mogu preuzeti od sustava na kojem se kontejner izvršava te kontejner nije izoliran samo na komunikaciju s domaćinom.



```
41 class ReceiveSensorMessagesBehaviour(CyclicBehaviour):
42     async def run(self):
43         message = await self.receive(timeout=SERVICE_AGENT_MESSAGE_TIMEOUT)
44
45         if message:
46             webAppAPI.log(f"Zaprimljena je poruka od agenta senzora {str(message.sender)}")
47             print(f"Zaprimljena je poruka od agenta senzora {str(message.sender)}")
48             self.agent.sensorAgent = str(message.sender)
49             self.agent.deviceName = message.body
50
51             serviceCenters = webAppServicesAPI.getAllServicesForDevice(self.agent.deviceName)
52             for serviceCentre in serviceCenters:
53                 for service in serviceCentre["services"]:
54                     if service["device"]["name"].lower() != self.agent.deviceName.lower():
55                         serviceCentre["services"].remove(service)
56             self.agent.serviceCenters = serviceCenters
57             self.agent.offers = []
58
59             self.agent.isAuctionStarted = True
60             self.agent.auctionTimeout = AUCTION_DURATION
61             for serviceCentre in self.agent.serviceCenters:
62                 bidderAgentData = {
63                     "name": f"Bidder agent | {serviceCentre['name']}",
64                     "xmpp": {
65                         "hostname": agentData.xmpp.hostname,
66                         "jid": f"agent-bidder-{serviceCentre['id']}",
67                         "password": "0000"
68                     },
69                     "data": {
70                         "SERVICE_CENTRE": serviceCentre,
71                         "SERVICE_AGENT_JID": agentData.xmpp.jid
72                     }
73                 }
74                 bidderAgentDataEncoded = base64.b64encode(json.dumps(bidderAgentData).encode('utf-8')).decode()
75                 subprocess.run(f"docker run --rm -d --network host -v ${pwd}:/opt/iot-mas/agents -w /opt/iot-mas/agents daspicko/spade:1.0 python3 agent_bidder.py {bidderAgentDataEncoded}")
76             else:
77                 webAppAPI.log(f"Nije bilo upita u posljednjih {DateTimeUtils.secondsToHumanTime(SERVICE_AGENT_MESSAGE_TIMEOUT)}")
78                 print(f"Nije bilo upita u posljednjih {DateTimeUtils.secondsToHumanTime(SERVICE_AGENT_MESSAGE_TIMEOUT)}")
79
```

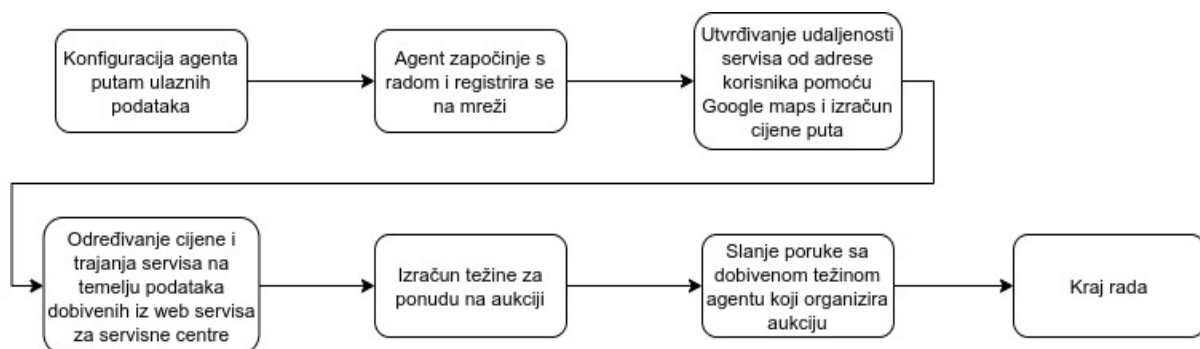
Slika 65. Servisni agent - Ponašanje za prihvaćanje poruka agenata uređaja

Servisni agent prilikom pokretanja agenata aukcija svakom agentu dodjeljuju naziv koji se sastoji od naziva agenta i ID-a servisnog centra kako bi ga bilo jednostavnije razlikovati i odrediti koji agent zastupa koji servisni centar.

4.3.1.4. Agent aukcije

Agent aukcije definiran je jednokratnim ponašanjem u kojem određuje vrijednost ponude na temelju podataka koje definira servisni centar. Agent započinje s radom prilikom pokretanja Docker kontejnera i učitavanjem ulaznih podataka. Agent zatim pokreće izračun vrijednosti težine ponude. U izračun vrijednosti uzima se u obzir udaljenost od korisnika te vrijednost utječe na konačni rezultat izračunom 2 kn po km. Udaljenost se određuje od adrese korisnika do adrese servisnog centra preko Google Maps API [71] za udaljenost.

U rezultat se uračunava i cijena izvršenja usluge te vrijeme trajanja usluge. Konačan broj šalje se servisnom agentu koji provodi aukciju.



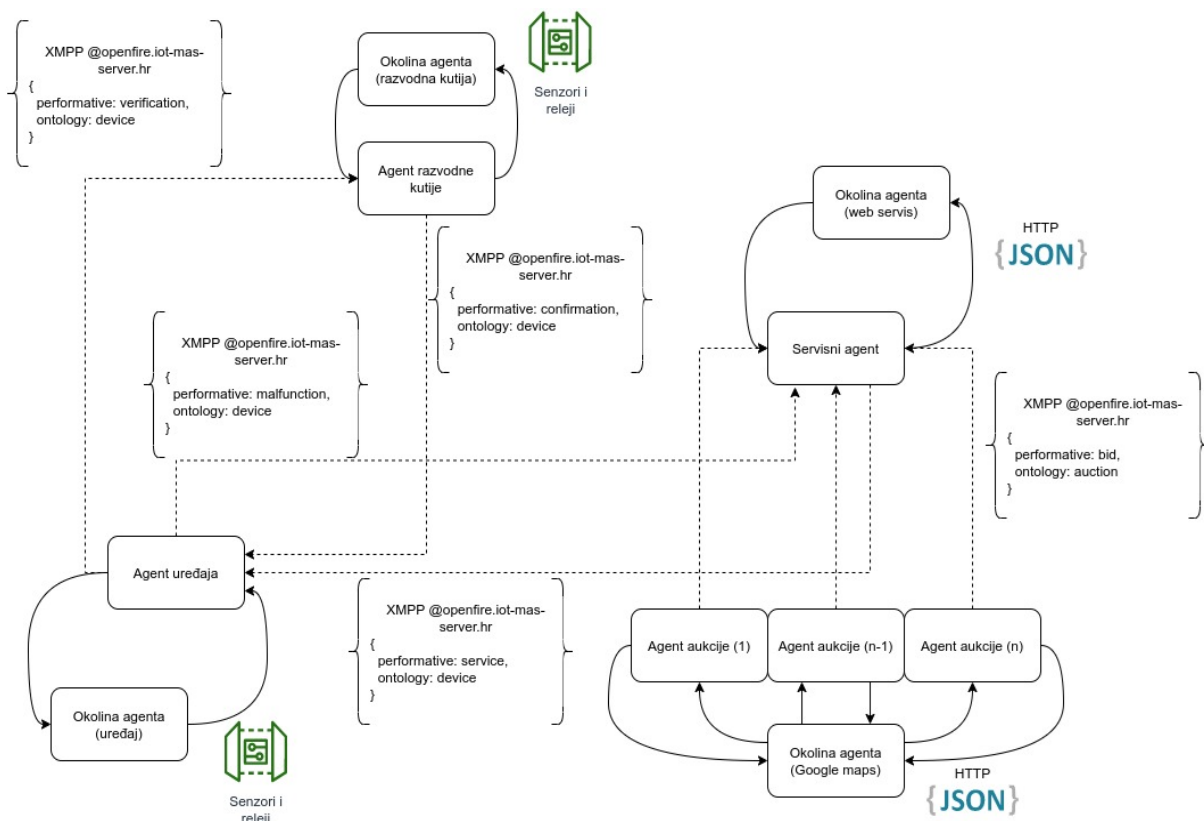
Slika 66. Agent aukcije - Koraci jednokratnog ponašanja

Agent završava s radom nakon slanja ponude nakon čega se zaustavlja Docker kontejner te se oslobađaju resursi.

4.3.2. Komunikacija agenata

Komunikacija se vrši preko XMPP poruka. Prilikom pokretanja agenta koriste se podaci koje je definirao korisnik u admin web aplikaciji. Podaci su proizvoljni, a agenti se automatski registriraju na poslužitelju bez dodatne provjere jesu li podaci korisničkog računa ispravni.

OpenFire omogućava stvaranje korisničkih računa preko REST API poziva te je moguće podesiti sustav da prihvaća samo registrirane korisnike. S obzirom na to da je XMPP poslužitelj instaliran na lokalnoj mreži s ograničenim pristupom, ta mogućnost je isključena i agenti se mogu slobodno registrirati na temelju podataka koje je korisnik unio u obrazac agenta. XMPP poruke sastoje se od primatelja, tijela poruke koja sadrži podatke te metapodataka koji definiraju vrstu poruke i na koji se objekt odnose. Na temelju metapodataka u predlošku SPADE određuje koje ponašanje pokrenuti.



Slika 67. Komunikacija agenata i interakcija s okruženjem

U sustavu postoji pet predložaka koji određuju ponašanje i kome je poruka namijenjena. Oznaku uređaj (engl. *device*) imaju poruke koje se odnose na provjeru ili stanje uređaja, dok oznaku aukcija (engl. *auction*) imaju poruke koje se šalju kao natječaj aukcije. Akcije koje su definirane u predlošcima poruka su provjera (engl. *verification*), greška u radu (engl. *malfunction*), potvrda (engl. *confirmation*) i usluga popravka (engl. *service*).

Komunikacija agenata s okolinom vrši se na dva načina. Komunikacija i interakcija agenata s okolinom moguća je očitavanjem vrijednosti senzora (ampermetara i voltmetara) te reakcijom na izlaznim jedinicama (relejima) ili pak utjecajem na programsko okruženje i

dohvaćanje podataka iz drugih sustava pri čemu se komunikacija odvija preko web servisa (REST API).

```
142 class ServiceAgent(Agent):
143     deviceName = ""
144     serviceCentres = []
145     offers = []
146     determineAuctionWinnerBehaviour = None
147     isAuctionStarted = False
148     auctionTimeout = AUCTION_DURATION
149
150     async def setup(self):
151         sensorMessageTemplate = Template()
152         sensorMessageTemplate.set_metadata("performative", "malfunction")
153         sensorMessageTemplate.set_metadata("ontology", "device")
154         self.add_behaviour(ReceiveSensorMessagesBehaviour(), template=sensorMessageTemplate)
155
156         bidMessageTemplate = Template()
157         bidMessageTemplate.set_metadata("performative", "bid")
158         bidMessageTemplate.set_metadata("ontology", "auction")
159         self.add_behaviour(ReceiveBidderMessagesBehaviour(), template=bidMessageTemplate)
160
```

Slika 68. Servisni agent - Definicija predložaka poruka za komunikaciju

Definiciju predloška poruke možemo vezati uz određeno ponašanje. Kad stigne poruka koja sadrži podatke koji odgovaraju definiranom predlošku, pokreće se ponašanje. Korištenje predložaka olakšava korištenje poruka i omogućava specijalizirano korištenje poruka. Predlošci omogućavaju korištenje logičkih operatora čime je moguće kreirati napredne funkcionalnosti i definirati poruke koje jednoznačno određuju vrstu akcije koju je potrebno poduzeti.

5. Zaključak

Posljednjih nekoliko godina svjedoci smo brzog razvoja tehnologije što uključuje veće performanse i povezanost, a istovremeno smanjenje veličine uređaja i mogućnost njihove integracije u uređaje koje koristimo na dnevnoj bazi. Time nam je omogućeno povezivanje uređaja na internet s ciljem praćenja rada i kontrole, a sve u svrhu olakšavanja rada korisniku i stavljanja tehnologije na raspolaganje čovjeku u bilo kojem trenutku. Kao primjer možemo navesti klimatizacijske uređaje - korisnik ga je u mogućnosti uključiti preko interneta, definirati vremena uključivanja i isključivanja kako bi ga ured ili dom dočekao rashlađen kad stigne na odredište.

Način kontrole i povezivanja uređaja na internet pogrešno se u marketinške svrhe oglašava kao pametnim. Tako primjerice imamo pametne žarulje ili pametne klimatizacijske uređaje koji ni u jednom smislu nisu pametni, već su povezivanjem na internet postali dostupni korisniku preko mobilne aplikacije. Da bi ti uređaji postali pametni, potrebno im je dodati određenu vrstu autonomije kako bi neovisno o korisniku mogli osigurati ostvarenje cilja. Primjerice, pametni klimatizacijski uređaj koji prati temperaturu izvan kuće i u njoj, navike korisnika i trenutnu lokaciju korisnika. Na temelju tih podataka pametan klimatizacijski uređaj odredio bi udaljenost korisnika od kuće preko lokacije, odredio jačinu hlađenja na temelju unutarnje i vanjske temperature kao i optimalne temperature koja odgovara korisniku i nije štetna za zdravlje te ohladio prostor kako bi se korisnik pri povratku osjećao ugodno.

Primjere pametnih uređaja u različitim područjima industrije možemo vidjeti u upravljanju vozilima, upravljanju prtljagom, nadgledanju proizvodnje i proizvodnih procesa, pametnom navodnjavanju, praćenju životinja, upravljanju poljoprivrednim zemljištima, pametnim gradovima, pametnim kućama te u medicinskim svrhama.

Kako će primjena interneta stvari i višeagentnih sustava utjecati na razvoj tehnologije i društva, vidjet ćemo za nekoliko godina, a za sad je moguće predvidjeti kretanja na temelju prethodnih događanja.

S obzirom na broj uređaja koji se danas spaja na mrežu, još prije 20-ak godina predviđeno je pomanjkanje adresnog prostora, a 2017. godine kao internet standard uvedena je IPv6 adresa koja omogućuje spajanje 2^{128} uređaja. Uz povećanje adresnog prostora, dostupne su i mobilne mreže koje ubrzavaju promjene jer pružatelji usluga mogu pružiti uslugu većem broju korisnika neovisno o njihovoj lokaciji, a brzine koje pružaju su zadovoljavajuće velike. Interakcija će se također ubrzati uvođenjem 5G mreže koja će omogućiti upravljanje uređajima u realnom vremenu s obzirom na latenciju od 1 milisekunde. S obzirom na mogućnosti i brzine 5G mreže, danas u vijestima možemo vidjeti kako se komunikacijski divovi

bore oko istraživanja i postavljanja mreže za korisnike, a jedan od događaja koji je sugerirao važnost jest zabrana poslovanja Huaweijsa s tvrtkama u SAD-u.

Pojavom Arduina, a kao i njegovih klonova, programiranje mikrokontrolera se pojednostavljuje i približava općoj populaciji što omogućava da su znanja i uređaji dostupni većem broju korisnika koji se aktivno bave i uče o postojećim, ali i novim mogućnostima upotrebe u svakodnevnom životu. To će također utjecati i na obrazovne institucije koje će ponuditi obrazovne programe za nove naraštaje developera i inženjera koji će raditi u tom području. Već nekoliko godina možemo vidjeti da popularnost STEM područja naglo raste, iako je broj ljudi u Hrvatskoj koji se njime žele baviti na jako niskoj razini.

Razvoj većeg broja uređaja i njihove primjene u svakodnevnom životu dovodi u pitanje sigurnost i privatnost korisnika. Od pojave mreža i interneta hakeri iskorištavaju različite propuste kako bi ostvarili neku korist neovlaštenim pristupom dokumentima, osobnim podacima ili bilo kojim drugim resursima. Unatoč mogućnosti neovlaštenog preuzimanja digitalnih informacija, hakeri nisu bili u mogućnosti fizički utjecati na okruženje u kojem se nalazi korisnik. To će dovesti do potrebe korištenja kriptografije na svim komunikacijskim kanalima i protokolima, ali i potrebu da se ljudi educiraju u tom području. Mjere koje se danas koriste u zaštiti informacijskih sustava, web ili mobilnih aplikacija sigurne su, no najslabija karika najčešće je čovjek koji svojom nepažnjom, neznanjem ili nemarom hakerima ili malicioznom softveru omogući pristup.

Unatoč otvorenim tehnološkim i sigurnosnim pitanjima, razvoj interneta stvari i interneta stvari kao višeagentnog sustava omogućit će razvoj tehnoloških dodataka koji će omogućiti ljudima kvalitetniju i optimalniju potrošnju resursa, kvalitetnije usluge, optimiziranu proizvodnju te bolje praćenje zdravlja i poboljšanje kvalitete života općenito.

Popis literature

- [1] »Napierove kosti,« Wolfram Math World, [Mrežno]. Available: <https://mathworld.wolfram.com/NapiersBones.html>. [Pokušaj pristupa 2 1 2021].
- [2] »Oughtred, William,« Hrvatska enciklopedija, 2021. [Mrežno]. Available: <https://www.enciklopedija.hr/natuknica.aspx?id=45943>. [Pokušaj pristupa 2 1 2021].
- [3] »Wilhelm Schickard i rotirajući sat,« History Computer, [Mrežno]. Available: <https://history-computer.com/wilhelm-schickard-and-the-rotating-clock-complete-history/>. [Pokušaj pristupa 2 1 2021].
- [4] »Blaise Pascal,« *Osječki matematički list*, p. 119, 2006.
- [5] »Charles Babbage,« Computer History Museum, [Mrežno]. Available: <https://www.computerhistory.org/babbage/charlesbabbage/>. [Pokušaj pristupa 2 1 2021].
- [6] »Ada Lovelace,« Britannica, [Mrežno]. Available: <https://www.britannica.com/biography/Ada-Lovelace>. [Pokušaj pristupa 2 1 2021].
- [7] M. R. Swaine, »ENIAC,« Britannica, [Mrežno]. Available: <https://www.britannica.com/technology/ENIAC>. [Pokušaj pristupa 2 1 2021].
- [8] M. Riordan, »Tranzistor,« Britannica, [Mrežno]. Available: <https://www.britannica.com/technology/transistor>. [Pokušaj pristupa 2 1 2021].
- [9] M. A. Dennis, »Jack Kilby,« Britannica, [Mrežno]. Available: <https://www.britannica.com/biography/Jack-Kilby>. [Pokušaj pristupa 2 1 2021].
- [10] »Marcian Hoff,« Computer History Museum, [Mrežno]. Available: <https://computerhistory.org/profile/marcian-hoff/>. [Pokušaj pristupa 2 1 2021].
- [1] »Rođenje IBM PC-a,« IBM, [Mrežno]. Available: https://www.ibm.com/ibm/history/exhibits/pc25/pc25_birth.html. [Pokušaj pristupa 2 1 2021].
- [1] »Vremenska crta povijesti računala,« Computer History Museum, [Mrežno]. Available: <https://www.computerhistory.org/timeline/1989/>. [Pokušaj pristupa 2 1 2021].
- [1] »O nama,« Arduino, [Mrežno]. Available: <https://www.arduino.cc/en/Main/AboutUs>.
3] [Pokušaj pristupa 2 1 2021].
- [1] »Povijest Raspberry Pi,« Hitaltech, [Mrežno]. Available: <https://hitaltech.co.uk/the-history-of-the-raspberry-pi/>. [Pokušaj pristupa 2 1 2021].
4]

- [1] »Teensy,« Electronic Projects, [Mrežno]. Available: <https://www.pjrc.com/teensy/>. [Pokušaj pristupa 2 1 2021].
- [1] »Photon,« Particle Docs, [Mrežno]. Available: <https://docs.particle.io/quickstart/photon/>. [Pokušaj pristupa 2 1 2021].
- [1] »Sparkfun,« [Mrežno]. Available: <https://www.sparkfun.com/>. [Pokušaj pristupa 2 1 2021].
- [1] »FEATHER,« Adafruit, [Mrežno]. Available: <https://www.adafruit.com/category/943>. [Pokušaj pristupa 2 1 2021].
- [1] »Beagle Board,« [Mrežno]. Available: <http://beagleboard.org/>. [Pokušaj pristupa 2 1 9] 2021].
- [2] »Digispark,« DigiStump, [Mrežno]. Available: <http://digistump.com/products/1>. [Pokušaj pristupa 2 1 2021].
- [2] »Što je Arduino, a što Croduino,« e-radionica, [Mrežno]. Available: <https://e-radionica.com/hr/blog/2015/10/08/sto-je-arduino-i-croduino/>. [Pokušaj pristupa 2 1 2021].
- [2] »Banana Pi Open Source Project,« [Mrežno]. Available: <http://www.banana-pi.org/>. [Pokušaj pristupa 2 1 2021].
- [2] »Orange Pi,« [Mrežno]. Available: <http://www.orangepi.org/>. [Pokušaj pristupa 2 1 2021].
- [2] »La Frite: Open Source Fries,« Kickstarter, [Mrežno]. Available: <https://www.kickstarter.com/projects/librecomputer/la-frite-open-source-fries>. [Pokušaj pristupa 2 1 2021].
- [2] C. Miller, »Bill Gates: kako je čudak promijenio svijet,« BBC, [Mrežno]. Available: <http://news.bbc.co.uk/2/hi/business/7461783.stm>. [Pokušaj pristupa 2 1 2021].
- [2] S. Haller, S. Karnouskos i C. Schroth, »Internet stvari u kontekstu poduzeća,« u *Future Internet*, Beč, 2008.
- [2] »Ured za objave EU,« 3 2010. [Mrežno]. Available: <https://op.europa.eu/en/publication-detail/-/publication/ed079554-72c3-4b4e-98f3-34d2780c28fc>. [Pokušaj pristupa 9 1 2021].
- [2] D. Ueckelmann, M. Harrison i F. Michahelles, Projektiranje interneta stvari, New York: Springer, 2011.
- [2] »30+ primjena internet stvari u 2021,« AI Multiple, 7 1 2021. [Mrežno]. Available: <https://research.aimultiple.com/iot-applications/>. [Pokušaj pristupa 11 1 2021].
- [3] »Što je Arduino,« Arduino, [Mrežno]. Available: <https://www.arduino.cc/en/Guide/Introduction>. [Pokušaj pristupa 11 1 2021].

- [3 »Arduino UNO,« Arduino, [Mrežno]. Available: <https://store.arduino.cc/arduino-uno-rev3>.
1] [Pokušaj pristupa 3 1 2021].
- [3 »Arduino Leonardo,« Arduino, [Mrežno]. Available: <https://store.arduino.cc/arduino-leonardo-with-headers>. [Pokušaj pristupa 3 1 2021].
- [3 »Arduino NANO,« Arduino, [Mrežno]. Available: <https://store.arduino.cc/arduino-nano-every-with-headers>. [Pokušaj pristupa 3 1 2021].
- [3 »Arduino Micro,« Arduino, [Mrežno]. Available: <https://store.arduino.cc/arduino-micro>.
4] [Pokušaj pristupa 3 1 2021].
- [3 »Arduino Mega,« Arduino, [Mrežno]. Available: <https://store.arduino.cc/arduino-mega-2560-rev3>. [Pokušaj pristupa 3 1 2021].
- [3 »Arduino NANO IoT,« Arduino, [Mrežno]. Available: <https://store.arduino.cc/arduino-nano-33-iot>. [Pokušaj pristupa 3 1 2021].
- [3 »Arduino MKR WiFi,« Arduino, [Mrežno]. Available: <https://store.arduino.cc/arduino-mkr-wifi-1010>. [Pokušaj pristupa 3 1 2021].
- [3 »Arduino UNO WiFi Rev2,« Arduino, [Mrežno]. Available: <https://store.arduino.cc/arduino-uno-wifi-rev2>. [Pokušaj pristupa 3 1 2021].
- [3 »Arduino MKR GSM,« Arduino, [Mrežno]. Available: <https://store.arduino.cc/arduino-mkr-gsm-1400-1415>. [Pokušaj pristupa 3 1 2021].
- [4 »Produkti,« Raspberry Pi, [Mrežno]. Available: <https://www.raspberrypi.org/products/>.
0] [Pokušaj pristupa 4 1 2021].
- [4 S. J. Russel i P. Norvig, »Uvod, Okruženja,« u *Moderan pristup umjetnoj inteligenciji*, New
1] Jersey, Prentice Hall, 1995, pp. 31-46.
- [4 P. Maes, »Umjetni život susreće zabavu - život autonomnih agenata,« *Communications
2] of the ACM*, pp. 108-114, 11 1995.
- [4 P. Stone i M. Veloso, »A survey from a machine learning perspective, Autonomous
3] Robots,« *Multiagent systems*, pp. 345-383, 2000.
- [4 B. P. Gokulan i D. Sirinivasan, »Research Gate,« [Mrežno]. Available:
4] https://www.researchgate.net/publication/226165258_An_Introduction_to_Multi-Agent_Systems. [Pokušaj pristupa 21 1 2021].
- [4 T. Sandholm, u *Višeagentni sustavi - Moderan pristup distribuiranoj umjetnoj inteligenciji*,
5] 1999, p. 204.
- [4 B. Matic, u *Međunarodno poslovanje*, Zagreb, Sinergija-nakladništvo d.o.o., 2004, pp.
6] 140-146.

- [4 F. Sollish i J. Semanik, »Wiley,« [Mrežno]. Available: 7] <https://onlinelibrary.wiley.com/doi/pdf/10.1002/9781119205098.app4>. [Pokušaj pristupa 24 1 2021].
- [4 P. Pico-Valencia, J. A. Holgado-Terriza, D. Herrera-Sanchez i J. Sampietro, »Prema 8] internetu agenata: Analiza interneta stvari iz perspektive inteligencije i autonomije,« *Ingeniería e Investigación*, pp. 121-129, 4 2018.
- [4 »Raspberry Pi,« [Mrežno]. Available: <https://www.raspberrypi.org/>. [Pokušaj pristupa 18 1 9] 2021].
- [5 »SPADE,« [Mrežno]. Available: <https://spade-mas.readthedocs.io/en/latest/readme.html>. 0] [Pokušaj pristupa 18 1 2021].
- [5 »XMPP,« [Mrežno]. Available: <https://xmpp.org/>. [Pokušaj pristupa 18 1 2021]. 1]
- [5 »OpenFire,« Ignite realtime, [Mrežno]. Available: 2] <https://www.igniterealtime.org/projects/openfire/>. [Pokušaj pristupa 18 1 2021].
- [5 »Spring Boot,« Spring, [Mrežno]. Available: <https://spring.io/projects/spring-boot>. 3] [Pokušaj pristupa 18 1 2021].
- [5 »jQuery,« [Mrežno]. Available: <https://jquery.com/>. [Pokušaj pristupa 18 1 2021]. 4]
- [5 »Bootstrap,« [Mrežno]. Available: <https://getbootstrap.com/>. [Pokušaj pristupa 18 1 2021]. 5]
- [5 »ReactJS,« [Mrežno]. Available: <https://reactjs.org/>. [Pokušaj pristupa 18 1 2021]. 6]
- [5 »Docker,« [Mrežno]. Available: <https://www.docker.com/>. [Pokušaj pristupa 18 1 2021]. 7]
- [5 »Ubuntu,« Canonical, [Mrežno]. Available: <https://ubuntu.com/server>. [Pokušaj pristupa 8] 18 1 2021].
- [5 »Socket.io,« [Mrežno]. Available: <https://socket.io>. [Pokušaj pristupa 18 1 2021]. 9]
- [6 »AWS Lightsail,« Amazon, [Mrežno]. Available: <https://aws.amazon.com/lightsail>. 0] [Pokušaj pristupa 18 1 2021].
- [6 »Debian,« [Mrežno]. Available: <https://www.debian.org>. [Pokušaj pristupa 18 1 2021]. 1]
- [6 »Nginx,« [Mrežno]. Available: <https://www.nginx.com>. [Pokušaj pristupa 18 1 2021]. 2]

- [6] »Swagger,« [Mrežno]. Available: <https://swagger.io>. [Pokušaj pristupa 18 1 2021].
3]
- [6] »Raspberry Pi OS,« [Mrežno]. Available: <https://www.raspberrypi.org/software>. [Pokušaj
4] pristupa 24 1 2021].
- [6] »Git,« [Mrežno]. Available: <https://git-scm.com>. [Pokušaj pristupa 24 1 2021].
5]
- [6] »Bitbucket,« Atlassian, [Mrežno]. Available: <https://bitbucket.org>. [Pokušaj pristupa 24 1
6] 2021].
- [6] »SPADE WGI,« [Mrežno]. Available: [https://spade-
mas.readthedocs.io/en/latest/web.html](https://spade-
7] mas.readthedocs.io/en/latest/web.html). [Pokušaj pristupa 26 1 2021].
- [6] »SQLite,« [Mrežno]. Available: <https://www.sqlite.org/index.html>. [Pokušaj pristupa 24 1
8] 2021].
- [6] »Jinja2,« [Mrežno]. Available: <https://jinja.palletsprojects.com/en/2.11.x/>. [Pokušaj
9] pristupa 24 1 2021].
- [7] »ChartJS,« [Mrežno]. Available: <https://www.chartjs.org>. [Pokušaj pristupa 24 1 2021].
0]
- [7] »Google Maps API,« [Mrežno]. Available:
1] <https://developers.google.com/maps/documentation/distance-matrix/overview>. [Pokušaj
pristupa 24 1 2021].
- [7] N. Heath, »Inside the Raspberry Pi: The story of the \$35 computer that changed the
2] world,« Techrepublic, 19 12 2018. [Mrežno]. Available:
[https://www.techrepublic.com/article/inside-the-raspberry-pi-the-story-of-the-35-
computer-that-changed-the-world/](https://www.techrepublic.com/article/inside-the-raspberry-pi-the-story-of-the-35-
computer-that-changed-the-world/). [Pokušaj pristupa 2 1 2021].
- [7] »ENIAC – Povijest prvog računala,« Hyperaxion, [Mrežno]. Available:
3] <https://hyperaxion.com/technology/eniac-first-computer/>. [Pokušaj pristupa 2 1 2021].
- [7] »IBM PC-JR i PC-AT,« Università della Svizzera Italiana's, [Mrežno]. Available:
4] [http://piermarcobarbe.github.io/informatics_history_HCI_atelier_2015/html/hardware/ibm-
pcjr_pcat_hw.html](http://piermarcobarbe.github.io/informatics_history_HCI_atelier_2015/html/hardware/ibm-
pcjr_pcat_hw.html). [Pokušaj pristupa 2 1 2021].
- [7] O. Vermesan, P. Friess, P. Guillemin, S. Gusmeroli, H. Sundmaeker, A. Bassi, I. S. Jubert,
5] M. Mazura, M. Harrison, M. Eisenhauer i P. Doody, »IERC - European Research Cluster
on the Internet of Things,« 2011. [Mrežno]. Available: [http://www.internet-of-things-
research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2011.pdf](http://www.internet-of-things-
research.eu/pdf/IoT_Cluster_Strategic_Research_Agenda_2011.pdf). [Pokušaj pristupa 9
1 2021].

[7 M. Wooldrige, »Pametani agent,« u *Uvod u višeagentne sustave*, West Sussex, John Wiley & Sons, LTD, 2002, p. 16.

Popis slika

Slika 1. ENIAC [43].....	3
Slika 2. IBM PC [44]	3
Slika 3. Arduino NANO klon.....	3
Slika 4. Raspberry Pi Zero WiFi.....	3
Slika 5. Arduino UNO [29]	11
Slika 6. Arduino LEONARDO [30].....	11
Slika 7. Arduino NANO [34].....	11
Slika 8. Arduino MICRO [32].....	12
Slika 9. Arduino MEGA [33].....	12
Slika 10. Arduino NANO 33 IoT [33]	12
Slika 11. Arduino MKR WiFi 1010 [35].....	12
Slika 12. Arduino UNO WiFi REV2 [36]	12
Slika 13. Arduino MKR GSM 1400 [37].....	12
Slika 14. Raspberry Pi 1 Model A+	14
Slika 15. Raspberry Pi 1 Model B+	14
Slika 16. Raspberry Pi 2 Model B	14
Slika 17. Raspberry Pi 3 Model A+	14
Slika 18. Raspberry Pi 3 Model B	14
Slika 19. Raspberry Pi 3 Model B+	14
Slika 20. Raspberry Pi 4 Model B	14
Slika 21. Raspberry Pi Zero.....	14
Slika 22. Raspberry Pi Zero W	14
Slika 23. GPS.....	23
Slika 24. Magnetometar.....	23
Slika 25. Pasivni infracrveni senzor	23
Slika 26. Senzor udaljenosti (ultrazvučni).....	23
Slika 27. Senzor rotacije.....	23
Slika 28. Senzor nagiba.....	23
Slika 29. Žiroskop.....	23

Slika 30. Senzor ubrzanja.....	23
Slika 31. Senzor vibracija	23
Slika 32. Senzor sile	24
Slika 33. Senzor dodira	24
Slika 34. Zaslon na dodir	24
Slika 35. Senzor razine tekućina	24
Slika 36. Senzor protoka tekućina	24
Slika 37. Senzor tlaka tekućina.....	24
Slika 38. Senzor koncentracije plina	24
Slika 39. Senzor za mjerenje osvjetljenja sa fotodiodom	24
Slika 40. Senzor temperature	24
Slika 41. Mikrofon.....	25
Slika 42. Senzor struje.....	25
Slika 43. Senzor napona	25
Slika 44. Agent u svojoj okolini [61]	28
Slika 45. Opća klasifikacija višeagentnih sustava s obzirom na attribute [42]	30
Slika 46. Hijerarhijska arhitektura višeagentnog sustava	31
Slika 47. "Superholon"	32
Slika 48. Koalicija višeagentnog sustava	33
Slika 49. Organizacija višeagentnih sustava u timovima.....	33
Slika 50. Arhitektura sustava	41
Slika 51. Web aplikacija - upravljanje agentima.....	42
Slika 52. Web aplikacija - Upravljanje servisnim centrima i uslugama	42
Slika 53. OpenFire - XMPP poslužitelj.....	43
Slika 54. Klasa podataka agenata	44
Slika 55. Primjer podataka koji se šalju agentu.....	45
Slika 56. Pomoćna klasa za spremanje akcija agenta u dnevnik rada	45
Slika 57. Agent razvodne kutije - Ponašanja.....	46
Slika 58. Prikaz kretanja vrijednosti napona i struje koje prati agent razvodne kutije	47
Slika 59. Isječak koda - Ciklično ponašanje za čekanje poruka od strane agentata uređaja .	47
Slika 60. Stanja konačnog automata agenta uređaja	49
Slika 61. Početno i željeno stanje agenta uređaja.....	50
Slika 62. Definiranje stanja i prijelaza u konačnom automatu	50
Slika 63. Servisni agent - Ponašanja	51
Slika 64. Docker datoteka za kreiranje slike kontejnera	52
Slika 65. Servisni agent - Ponašanje za prihvaćanje poruka agenata uređaja	52
Slika 66. Agent aukcije - Koraci jednokratnog ponašanja	53

Slika 67. Komunikacija agenata i interakcija s okruženjem	54
Slika 68. Servisni agent - Definicija predložaka poruka za komunikaciju	55

Popis tablica

Tablica 1. Usporedba specifikacija Raspberry Pi uređaja	15
---	----

Prilozi

1. Github repozitorij agenata - <https://github.com/daspicko/iot-mas-agents>
2. Github repozitorij Web aplikacije admin - <https://github.com/daspicko/iot-mas-app-admin>
3. Github repozitorij Web aplikacije servisnih centara - <https://github.com/daspicko/iot-mas-app-services>