



UNIVERZITET SINGIDUNUM
BEOGRAD
DEPARTMAN ZA POSLEDIPLOMSKE STUDIJE

DOKTORSKA DISERTACIJA
**RAZVOJ ADAPTIVNOG ČETBOTA
ZASNOVANOG NA VEŠTAČKOJ
INTELIGENCIJI**

Mentor: Prof. dr Miodrag Živković

Student: Miloš Kabiljo
Broj indeksa: 466020/2016

Beograd, 2022. godina



SINGIDUNUM UNIVERSITY
BELGRADE
DEPARTMENT FOR POSTGRADUATE STUDIES

PHD THESIS

**DEVELOPMENT OF ADAPTIVE
CHATBOT BASED ON ARTIFICIAL
INTELLIGENCE**

Mentor: Professor Miodrag Živković

Student: Miloš Kabiljo
Index number: 466020/2016

Belgrade, 2022.

Mentor:

Prof. dr Miodrag Živković, redovni profesor
Univerzitet Singidunum

Članovi komisije:

Prof. dr Nebojša Bačanin Džakula, vanredni profesor
Univerzitet Singidunum

Prof. dr Boško Nikolić, redovni profesor
Univerzitet u Beogradu
Elektrotehnički fakultet, Beograd

Zahvalnica

Veliku zahvalnost dugujem svom mentoru, profesoru dr Miodragu Živkoviću koji me je od početka vodio kroz celokupni proces i pomagao tokom istraživanja, objave radova i izrade same doktorske teze.

Zahvaljujem se članovima komisije na podršci, korisnim sugestijama i primedbama tokom izrade i pregleda doktorske disertacije.

Posebnu zahvalnost dugujem svom kolegi, prijatelju i neformalnom mentoru, profesoru dr Radetu Matiću, koji je od početka verovao u mene i uvek bio tu da me sasluša, daje sugestije i nesebično pomogne deleći svoje znanje sa mnom. Uložio je puno truda i vremena tokom procesa istraživanja, kako bi ova doktorska teza bila što kvalitetnija. Velika je čast i privilegija raditi sa njim.

Želim da se zahvalim kolegi Milenu Janjiću i ostalim kolegama iz firme Intellya koji su mi dali veliku podršku tokom istraživanja. Pomogli su mi u eksperimentalnom radu, stručnim savetima i sugestijama doprineli da ova teza bude još bolja.

Kolegi dr Milanu Čabarkapi se zahvaljujem na stručnim savetima i sugestijama u izradi radova tokom istraživanja.

Zahvaljujem se svojim roditeljima i bratu koji su bili uz mene i pomagali mi tokom celog školovanja, a posebno u poslednjih par godina tokom doktorskih studija su mi davali nesebičnu podršku.

Želim da se zahvalim svojoj porodici, pre svega supruzi Mariji, koja me je bezuslovno podržavala, imala strpljenja i uvek bila tu da pomogne kako bih ostvario ne samo ovaj nego i sve ciljeve u životu. I na kraju, najvažnije, zahvaljujem se svojoj deci Hani, Sari i Luki, koji su mi bili najvažniji motiv i inspiracija da ostvarim svoj cilj – doktorsku disertaciju.

RAZVOJ ADAPTIVNOG ČETBOTA ZASNOVANOG NA VEŠTAČKOJ INTELIGENCIJI

Sažetak: Nakon blagog pada pojedinih društvenih mreža istraživanja ukazuju da su krajnji korisnici najveću količinu svoje pažnje usmerili na komunikacione platforme (*Viber, Whatsapp, FB messenger, Instagram*, itd). Pojam digitalna transformacija zajedno sa veštačkom inteligencijom, predstavlja ogroman iskorak u pravcu unapređenja poslovanja kompanija iz različitih industrija. U borbi protiv KOVID-19, a u cilju razvoja inovativnih rešenja koja obezbeđuju dodatnu vrednost, u digitalni svet je lansiran četbot. On predstavlja program koji imitira ljudski razgovor i pomaže korisniku da dobije prave informacije i pruži svakodnevne usluge. U poslednjoj deceniji, ogromno unapređenje komunikacionih platformi i mrežnih tehnologija pružile su nove tehničke mogućnosti za usvajanje četbota u različitim sektorima, kao što su: e-trgovina, korisnička podrška, marketing i obrazovanje. Četbot predstavlja softver koji simulira konverzaciju koristeći veštačku inteligenciju i učestvuje u dijalogu sa čovekom koristeći prirodni jezik i pri tome razume tekstualne i glasovne poruke. Četbot koristi veštačku inteligenciju u cilju pronalaženja odgovora i rešavanja korisničkih zahteva. U koraku sa novim trendovima u radu je definisan metodološki okvir za razvoj savremene arhitekture četbota zasnovanog na veštačkoj inteligenciji za sledeću generaciju Internet zajednice. U skladu sa tim najveći problem razvoja adaptivne platforme je rešavanje problema fleksibilnosti integrisanja, odnosno definisanje mehanizma nezavisnosti četbota od samo jednog servisa za razumevanje prirodnog jezika (engl. Natural Language Understanding - NLU) čime bi se obezbedila adekvatna adaptivnost.

Predložena metodologija daje opšti postupak i rezultat je kombinacije različitih metodoloških pristupa. Prateći metodologiju, razvoj adaptivnog četbota zasnovan na NLU se sastoji iz dve faze: projektni i aplikacioni inženjering. Svaka od ovih faza je zatim podeljena na aktivnosti koje treba da urade u okviru faza. Prednost ove podele je realizacija prilagodljive platforme za generisanje četbotova. Napredna arhitektura četbota koja podržava metodologiju je lako proširljiva, skalabilna i podržava različite servise za razumevanje prirodnog jezika i komunikacione platforme za razmenu poruka. Teza opisuje logičku i fizičku arhitekturu četbota. Kroz odgovarajuće metamodele i njihova mapiranje obezbeđena je nezavisnot NLU servisa. Kroz predloženi pristup daje se

sopstveni nezavisni okvir za razvoj četbota čiji je cilj smanjenje zavisnosti od proizvođača koji se bave servisima za razumevanje prirodnog jezika i kanalima komunikacija. Na ovaj način se stiče nezavisnost od bilo kog servisa za razumevanje prirodnog jezika, čime se rešenje neće vezati za jedan servis već će se predloženim mehanizmom obezbediti lako mapiranje i prebacivanje sa jednog na drugi novi i/ili stari servis.

Konačno, jedna implementacija predložene arhitekture četbota je ukratko prikazana kroz studiju slučaja. Akademska Digitalna Asistentkinja (ADA) je prva implementacija četbota u prosveti na Balkanu koja kroz prirodnu komunikaciju pruža efikasniju realizaciju usluga i uštedu vremena. ADA prima poruke sa različitih komunikacionih platformi za razmenu poruka (Viber, FB mesindžer, Instagram i vebčeta preko sajta BAPUSS). Četbot platform obavlja veliki broj važnih zadataka pre nego što pošalje poruku NLU. NLU zatim analizira tekst i na osnovu tehnika mašinskog učenja upoređuje, a zatim generiše nameru i njene entitete. Generisani rezultat se šalje nazad u četbot platformu, koja određenim algoritmom koristi nameru, entitete, kontekst i šalje odgovor korisniku u odgovarajućem formatu za prikazivanje poruke. ADA je dostupna 24 sata i daje odgovore na česta pitanja studenata i budućih studenata. ADA zna nešto više od 250 scenarija u vezi sa pitanjima na koje studenti i ostali korisnici žele da dobiju odgovor ili da izvrše odgovarajuću uslugu. Neki od scenarija su: prijava ispita, promena ispitivača, overa semestra, informacije o konsultacijama nastavnika, rokovima, rangiranju akademije, rasporedu predavanja, rejtingu studenata, događajima, dostupnosti usluga, akademskom okruženju, datumu predavanja i ispita, aktivnostima koje se dešavaju unutar akademije i druge korisne akademske informacije.

Ključne reči: četbot, četbot arhitektura, razvoj četbota, NLU, platforma za razmenu poruka, metamodeli, modelovanje konteksta.

Naučna oblast: Tehničko-tehnološke nauke

Uža naučna oblast: Elektrotehnika i računarstvo

DEVELOPMENT OF ADAPTIVE CHATBOT BASED ON ARTIFICIAL INTELLIGENCE

Abstract: *Research shows that end users have largely focused from social networking to communication platforms (Viber, WhatsApp, FB messenger, Instagram, etc.). Artificial intelligence, along with the notion of digital transformation, is a major step in modernizing the business company of each industry. In the battle against COVID 19, and in the development of innovative solutions that provide added value, chatbot was launched into the digital world. It is a software that imitate human conversation and help user to get the right information and facilitate day-to-day services. In the last decade, the huge improvement of communication platforms and networking technologies has provided new technical opportunities for the adoption of chatbots in various sectors, such as e-commerce, customer support, marketing and education. The Chatbot is a software program that imitates conversation using artificial intelligence and engages in dialogue with humans using natural language understanding through textual and voice messages. Chatbot uses artificial intelligence in order to find answers and solve the user's requests. In step with new trends, the methodological framework for the development of modern chatbot architecture based on artificial intelligence for the next generation of the Internet community is defined. Accordingly, the biggest problem in the development of the adaptive platform is solving the problem of flexibility of integration, defining the mechanism of independence of chatbots from only one service for natural language understanding (NLU) which would ensure adequate adaptability.*

The proposed methodology provides a general procedure and is the result of a combination of different methodological approaches. According to the defined methodology, the process of developing an adaptive chatbot based on NLU is divided into two basic phases: project engineering and application engineering. Each of these phases is then divided into activities. The advantage of this separation is the realization of an adaptable platform for generating chatbots. It is also proposed scalable and extensible chatbot architecture supporting different services for natural language understanding and communication channels. The paper describes the logical and physical architecture of chatbots. The independence of the NLU service is ensured through appropriate metamodels and their mapping rules. The thesis provides its own independent framework

for reducing dependence on suppliers that deal with the services of natural language understanding and communication channels. In this way, independence is gained from any service for understanding natural language, which will not tie the solution to one service, but the proposed mechanism will provide easy mapping and switching from one to another new NLU service.

Finally, one implementation of the proposed solution is summarized through a case study. The Academic Digital Assistant (ADA) is the first educational project in the Balkans that provides better communication, save time, and deliver services faster and more efficiently. The ADA receives messages from various messaging communication platforms (Viber, FB Messenger, Instagram, and the website). The chatbot platform performs a number of important tasks before sending a message to the NLU. The NLU analyzes the text based on machine learning techniques, compares, and then generates intent and its entities. The generated result is sent back to the chatbot platform, which uses a certain algorithm of intent, entities, and context and sends a response to the user in the appropriate format to display the message. ADA is available 24 hours a day and provides answers to frequently asked questions from students and prospective students. ADA knows over 250 scenarios regarding questions an appropriate service that students want to be performed. Some of the scenarios are exam registration, change of examiner, semester certification, information on teacher consultations, deadlines, academy ranking, lecture schedule, student rating, events, service availability, academic environment, date of lectures and exams, activities taking place within the academy and other useful academic information.

Keywords: chatbot, chatbot architecture, development of chatbot, NLU, messaging platform, metamodels, context modeling

Scientific field: Technical and technological sciences

Narrow scientific field: Electrical engineering and computing Science

Sadržaj

1. UVOD	1
1.1. PREDMET ISTRAŽIVANJA, HIPOTEZE I CILJ	3
1.2. DOPRINOSI DISERTACIJE	7
1.3. SADRŽAJ DISERTACIJE	9
2. ČETBOT	11
2.1. DEFINICIJA ČETBOTA	11
2.1.1. Osnovni pojmovi u razvoju četbota	13
2.1.2. Domen i oblasti primene četbota	16
2.1.3. Ciljevi integracije četbota u softverske sisteme	17
2.2. MOTIVACIJA.....	20
2.2.1. Prednosti uvođenja četbota	23
2.3. KRATKA ISTORIJA ČETBOTA	25
2.3.1. Ostali primeri četbot rešenja	28
2.4. KLASIFIKACIJE ČETBOTA	29
2.5. TEHNIČKE SPECIFIKACIJE IMPLEMENTACIJE	35
2.6. PLATFORMA – ARHITEKTURA ZA RAZVOJ ČETBOTA.....	37
3. OBRADA PRIRODNOG JEZIKA - NLP.....	39
3.1. OSNOVNE KOMPONENTE NLU	44
3.1.1. Interakcija - prosleđivanje poruke	50
3.2. POREĐENJE DVA POZNATA NLU SERVISA	53
4. RAZVOJ ČETBOTA	56
4.1. OPŠTA REFERENTNA ARHITEKTURA.....	57
4.2. METODOLOGIJA RAZVOJA ČETBOTA	63
5. PROJEKTNI INŽENJERING	65

5.1. ANALIZA ELEMENATA RAZVOJNE PLATFORME	65
5.2. DEFINISANJE ARHITEKTURA PLATFORME ZA RAZVOJ ČETBOTA	66
5.2.1. Struktura komponenti BOT <i>engine</i> -a	71
5.3. DEFINISANJE OPŠTEG KONTEKSTNOG MODELA.....	74
5.4. DEFINISANJE PRISTUPA ZA MODELOVANJE NLU PLATFORMI I KOMUNIKACIONIH KANALA	81
5.4.1. Opšti postupak kreiranja adaptivnog četbota nezavisnog od NLU platforme	83
5.4.2. NLU metamodeli i odgovarajuća pravila mapiranja.....	86
5.5. DEFINISANJE KONVERZACIONOG DIZAJNERA ZA MODELOVANJE KONVERZACIONIH TOKOVA.....	95
5.6. RAZVOJ ČETBOT KOMPONENTI.....	97
6. APLIKACIONI INŽENJERING	99
6.1. ANALIZA DOMENA KONKRETNOG ČETBOTA	100
6.1.1. Akademska Digitalna Asistentkinja - ADA.....	101
6.2. IZBOR I SETOVANJE NLU SERVISA I KOMUNIKACIONIH KANALA	118
6.3. MODELOVANJE SCENARIJA KROZ KONVERZACIONI DIZAJNER.....	118
6.4. MODELOVANJE KONTEKSTA	123
6.5. NLU UČENJE	125
6.6. TESTIRANJE KONKRETNOG ČETBOTA	132
6.7. PUŠTANJE ČETBOTA NA PRODUKCIJU I VALIDACIJA FRAZA.....	133
7. EVALUACIJA ČETBOTA ADA	137
7.1. TESTIRANJE ADA ČETBOTA	138
7.2. STATISTIKA KORIŠĆENJA ADA ČETBOTA	140
8. POREĐENJE SA OSTALIM PRISTUPIMA	149
9. ZAKLJUČAK	155

9.1. PRAVCI DALJEG ISTRAŽIVANJA.....	160
10. LITERATURA.....	163
11. INDEKS SLIKA	174
12. INDEKS TABELA	177

1. UVOD

Obrada velikog broja zahteva u okviru poslovnih procesa u svakodnevnom radu poslovnih sistema srednje i velike veličine je postao standard, a to je dovelo do ograničenja u ljudskim resursima jer je sve manja mogućnost obrade istih u kratko vreme. S druge strane, decenijama unazad istraživači pokušavaju da poboljšaju svoje veštine u dizajniranju korisničkog interfejsa u interakciji između čoveka i računara. Tehnologija je ponudila rešenje u obliku virtualnog saradnika (asistenta), koji se naslanja na obradu prirodnih jezika (engl. Natural Language Processing - NLP) kao deo oblasti veštačke inteligencije (engl. Artificial Intelligence – AI), a sve u svrhu izbegavanja ograničenja ljudskog faktora i poboljšanja interakcije između čoveka i računara. Ideja je da se korisnički interfejs zasniva na prirodnom jeziku i da se interakcija sa digitalnim sistemima ne odvija preko skrola, pretrage, pomeranja ili pritiskanja dugmića, već preko glasa i teksta napisanog prirodnim jezikom. Pravi primer platforme koja koristi obradu prirodnog jezika kako bi komunicirao sa klijentima, odgovarao na sva njihova pitanja i rešavao njihove probleme je četbot (engl. chatbot). Četbot je vrsta softverskog programa koji simulira ljudsku konverzaciju ili ćaskanje, koristeći ili ne koristeći veštačku inteligenciju. Četbot može razumeti pisani tekst, glasovne poruke, i u velikoj meri može interpretirati njihovo značenje. Jedan od osnovnih zadataka četbota je kako će razumeti korisnikov zahtev bez obzira na jezik ili rečenicu koju je napisao ili izgovorio. Na osnovu prirodne obrade rečenice uz pomoć veštačke inteligencije i odgovarajućeg algoritma, četbot dolazi do adekvatnog odgovora koji prosleđuje korisniku.

U mnogim komercijalnim i nekomercijalnim oblastima četbotovi su se pokazali korisnim u različitim kontekstima za automatizaciju zadataka i poboljšanje korisničkog iskustva. Međutim, četbotovi su obično ograničeni na razgovore za posebne svrhe, a ne za ceo spektar ljudske komunikacije. Problemi koji se javljaju u razvoju četbot platformi se najčešće ogledaju u tome što oni treba da razumeju tačnu dubinu i kontekst razgovora i ovo predstavlja jedan od najvećih izazova. Osim toga četbot treba da omogući jednostavne višestruke dijaloge, upravlja tokom konverzacije i pri tome treba uvek da uzima odgovarajući kontekst iz istorije konverzacije. Osim razvoja arhitekture četbota

jedan od najvećih problema kojim se ova teza bavi je nezavisnost četbota od svih proizvođača na tržištu koji se bave servisima (platformama) za razumevanje prirodnih jezika (engl. Natural Language Understanding - NLU) kao delu NLP-a. Još uvek je nerealno pobediti velike svetske IT kompanije i njihova rešenja u oblasti veštačke inteligencije kao što su: *Google asistenta*, *Amazon Alexa*, *Apple Siri*, *IBM Watson* i sl. Ali je zato moguće prevazići ovu zavisnost od njih koristeći ovde predloženo rešenje. Takođe, neminovan je napredak postojećih i/ili razvoj novih servisa za razumevanje prirodnog jezika. S tim u vezi, treba biti što fleksibilniji i ne zavisiti od samo jednog servisa, već na što lakši način obezbediti prelazak na novu i/ili više platformi koje su neophodne za uspešnu realizaciju procesa i/ili zahteva korisnika. Dodatni problem predstavlja nemogućnost komuniciranja sa postojećim eksternim poslovnim sistemima kao što su kontakt centri, internet stvari i razni drugi poslovni i bankarski sistemi itd.

S obzirom da su komunikacione platforme za slanje poruka (npr. *Viber*, *FB messenger*, *Whatsapp*, *Instagram*, itd.), najčešći način koji ljudi koriste u komunikaciji, svaki biznis bi trebalo da ima strategiju za angažovanje ljudi koji koriste komunikacione platforme za razmenu poruka, a četbot je jedan od skalabilnih načina za to. Iz ovoga proizilazi da će svaka kompanija na kraju imati svog četbota. Četbotovi i njihovi odgovori su brži od veb stranica i mobilnih aplikacija jer nema instalacija, pritiskanja dugmića i pretrage po četbotu kao na aplikacijama. A prednosti brzine i velikog broja opsluženih klijenata od strane četbota je još jedna prednost ove tehnologije u odnosu na ljudske resurse. Takođe, usluga je brža jer se klijenti rešavaju tradicionalne komunikacije (kanala) putem telefona i mejla. Zatim se obezbeđuje svakodnevna visoka dostupnost klijenata koji zahtevaju usluge van normalnog radnog vremena. Četbot radi neprekidno 24 sata tokom cele godine i odgovara u roku od maksimalno 3 sekunde.

Razvoj četbota u ovoj disertaciji podrazumeva izgradnju prilagodljive, skalabilne i proširljive arhitekture. Arhitektura treba da omogući jednostavno povezivanje sa postojećim i novim servisima za razumevanje prirodnog jezika kao i sa komunikacionim platformama (kanalima). Imajući u vidu da potreba za što boljim servisima za razumevanje prirodnog jezika raste, problem efikasne primene adaptacije fleksibilnog četbota postaje jedno od ključnih pitanja za potpunu realizaciju vizije četbota.

Zbog svega prethodno navedenog, oblast razvoja adaptivnog četbota zasnovanog na veštačkoj inteligenciji predstavlja adekvatnu tehnološku osnovu za primenu nove digitalne strategija razvoja poslovanja u budućnosti koristeći elemente veštačke inteligencije kroz uštedu, automatizaciju i optimizaciju.

Ova doktorska disertacija se bavi problemom razvoja adaptivnog četbota zasnovanog na veštačkoj inteligenciji koji poseduje mehanizam adaptacije za blagovremeno i efikasno prilagođavanje postojećim i novim servisima za razumevanje prirodnog jezika NLU. Postojeća rešenja su uglavnom bazirana na jednom NLU servisu i pravilima na osnovu kojih četbot funkcioniše u određenoj situaciji. Međutim, zbog kompleksnosti arhitekture za potrebe razvoja četbota, cene koštanja i preciznosti razumevanja različitih NLU servisa, kao i zbog različitih korisnika sa različitim jezicima i potrebama, veoma je teško i neefikasno da se koristi rešenje koje nije zavisno od samo jednog NLU servisa.

U uvodnom poglavlju se u nastavku teksta opisuje predmet istraživanja i postavljaju se hipoteze. Zatim se na osnovu opisa problema i nerešenih dilema definišu osnovni ciljevi i podciljevi. Pregled doprinosa ove disertacije, koji pokazuju ostvarenje postavljenih ciljeva, se daje pre same strukture disertacije.

1.1. PREDMET ISTRAŽIVANJA, HIPOTEZE I CILJ

Širi predmet istraživanja disertacije je upravo metodološki okvir za razvoj savremene arhitekture četbota zasnovanog na veštačkoj inteligenciji za sledeću generaciju Internet zajednice.

Uži predmet istraživanja ove disertacije je rešavanje problema fleksibilnosti integrisanja, odnosno definisanje mehanizma nezavisnosti četbota od NLU servisa čime bi se obezbedila adekvatna adaptivnost. U skladu sa tim predmet istraživanja je pregled postojećih referentnih arhitektura četbota, njihove arhitekture i NLU servisi kao i njihova komparativna analiza.

Opšta hipoteza u istraživanju je:

Hipoteza H: Svaki biznis će imati svog četbota koji treba da omogući jednostavne višestruke dijaloge, upravljaju tokom konverzacije i pri tome uvek uzima odgovarajući kontekst iz istorije konverzacije. Osim toga pretpostavka je da će se veliki broj zadataka moći realizovati u kratkom vremenskom intervalu i prikupiti veliki broj informacija radi unapređenja sistema i poslovanja koristeći četbot.

Posebna hipoteza je data u nastavku:

H 0.1: Razvojem adaptivnog četbota zasnovanog na veštačkoj inteligenciji se može postići automatizacija velikog broja poslovnih procesa i brža obrada velikog broja zahteva i na taj način se povećati produktivnost kompanija.

Specijalizacijom prethodno navedene posebne hipoteze, formulišu se pojedinačne hipoteze, koje se odnose na osnovne elemente predmeta istraživanja:

H 0.1.1: Moguće je definisati fleksibilnu generičku arhitekturu koja je proširljiva, skalabilna i koja podržava različite integracije sa eksternim servisima.

H 0.1.2: Sistemom analize servisa razumevanja prirodnih jezika svetskih IT kompanija moguće je utvrditi da oni imaju slične koncepte i svi koriste kombinaciju više koraka kako bi konvertovao korisnikov tekst ili govor u strukturiran tip podataka koji se koristi za selektovanje odgovarajućeg odgovora. S tim u vezi, pretpostavka je da je moguće predložiti nezavisan mehanizam koji će da podrži sve postojeće koncepte čime će se omogućiti adaptacija četbota svakom servisu za razumevanje prirodnih jezika i na taj način obezbediti nezavisnost od NLU servisa bilo kog proizvođača.

Osnovni cilj ove doktorske disertacije je definisanje originalnog pristupa razvoja adaptivnog četbota. Predmet istraživanja je dodatno podstakao definisanje sledećih podciljeva:

- Razvoj metodologije i njegovog okvira za adaptivi četbot zasnovan na veštačkoj inteligenciji.
- Definisane fleksibilne generičke arhitekture koja je proširljiva, skalabilna i koja podržava različite integracije sa eksternim servisima. Adaptivni četbot

treba da bude „labavo povezan“ (engl. loosely coupled), što znači da njegove komponente treba da budu nezavisne i lako zamenljive kada za to ima potrebe.

- Dinamička adaptacija četbota sa postojećim i/ili novim servisima za obradu prirodnih jezika koristeći njihove odgovarajuće metamodela i mapiranje sa originalnim opštim metamodelima aplikacionog okvira.
- U toku istraživanja implementirano je više instanci četbota, i s tim u vezi jedan od ciljeva je unapređenje (adaptacija) postojećih implementacija četbota.

Ciljevi su zahtevali različitu primenu znanja o vezi između metodoloških okvira, servisa za razumevanje prirodnih jezika i četbota, a sve u svrhu davanja doprinosa u različitim oblastima.

Metodologija istraživanja bazirala se na korišćenju pre svega stručne literature, ali i praktične primene pojedinih konverzionih platformi u pojedinim oblastima. Korišćena je teorijska analiza rezultata istraživanja iz stručne i naučne literature autora koji su se takođe bavili razvojem četbota. Istražena su naučno-teorijska saznanja, relevantna literatura i poslovna praksa prilikom koje su se koristile sledeće metode: generalizacija i specijalizacija, metamodelovanje i mapiranje, posmatranja i klasifikacija, studije slučaja korišćenja, kao i opšte poznate metode dedukcije, analize i sinteze. Praktično iskustvo je od izuzetne važnosti u uvođenju četbota u produkciju, jer postoji mnogo procesa koje treba obraditi prilikom same implementacije ovakvog sistema. Prikupljanje iskustava od četbotova koje se mogu naći na tržištu je veoma koristan metod u istraživačkom procesu jer se tako mogu sagledati različite platforme i uvideti dobre i loše stvari kako bi četbot unapredio. Dobijeni podaci potiču uglavnom iz doktorskih disertacija i korporativnih istraživanja. Tokom obrade rezultata pronađenih iz istraživanja sa drugih univerziteta i disertacija primenjena je takođe metoda posmatranja i analiza sadržaja. Svi rezultati su korišćeni u svrhu definisanja istraživanja problema razvoja adaptivnog četbota.

Da bi se definisala i pokazala fleksibilnost predloženog četbota rešenja sprovedene su metode studije slučaja. Eksperimentalna analiza je primenjena u fazi evaluacije primene predloženog pristupa u procesu razvoja četbota. Beogradska akademija poslovnih i umetničkih strukovnih studija je prva obrazovna ustanova na Balkanu koja je koristeći ovde predloženu metodologiju implementirala prvi četbot baziran na veštačkoj inteligenciji, čiji je zadatak da ubrza poslovne procese i obradi veliki

broj prostih upita od strane studenata i potencijalnih kandidata tj. srednjoškolaca. Izvršena je evaluacija i izbor njihovih komponenata. Prilikom sagledavanja ovog problema upotrebljen je projektni metod. U poređenju sa vrhunskim časopisima, međunarodnim konferencijama i postojećim platformama za razvoj četbota, dobijeni su rezultati koji potvrđuju primenu korišćenih metoda. U okviru istraživanja dodatno su se primenili:

- U fazi analize konkretnih poslovnih procesa, za identifikaciju elementarnih poslovnih funkcija koristiće se i principi dekompozicije funkcija odozgo na dole (engl. top-down) pristupom. Kao komplementarni pristup koristiće se i analiza odvijanja poslovnog procesa (engl. workflow analysis).
- Standard za modelovanje poslovnih procesa kao što je BPMN (engl. Business Process Management Notation) i UML (engl. Unified Modeling Language) kao standardni jezik za modelovanje u oblasti softverskog inženjerstva za potrebe velikih i kompleksnih sistema.
- Tehnika mapiranja modela u osnovi predstavlja model veza i korespondencije elemenata između metamodela. Ovo mapiranje između metamodela nije obavezna transformacija elemenata iz jednog u drugi metamodel. Kada su određena dva elementa povezana po nekom pravilu kaže se u stvari da oni dele neki semantički link.

1.2. DOPRINOSI DISEsRTACIJE

Upotrebom rezultata i saznanja razvijenih zemalja i iskustava uz istraživanja koja mogu biti nadogradnja postojećih, zemlje kao što je Srbija mogle bi ostvariti značajan pomak u oblasti primene veštačke inteligencije. Ovi rezultati su se koristili kao početna tačka za istraživanja i praktičnu primenu sprovedenu u ovoj disertaciji. Očekivanja su da će istraživanja ove disertacije dodatno potvrditi koliko je tema ove doktorske disertacije atraktivna i aktuelna. Opšti doprinos istraživanja je u uočavanju značaja razvoja i primene četbota za potrebe savremenog društva, bržeg i lakšeg poslovanja. U skladu sa predmetom i metodama istraživanja, ciljevima, opštim i posebnim pretpostavkama, rezultati ove teze dali su doprinos napretku razvoja i primene četbota u komercijalnim i nekomercijalnim oblastima. Ovaj rad najpre teži ka naučnom doprinosu u oblasti razvoja adaptivnog četbota. Osnovni naučni doprinosi su:

- Razvoj metodološkog pristupa za izgradnju konverzionog četbota zasnovanog na veštačkoj inteligenciji.
- Specifikacija jedinstvene logičke arhitekture za praktičnu primenu predloženog pristupa u implementaciji složenog konverzionog četbot sistema.
- Definisane aplikativnog okvira (engl. framework) koji treba da omogućiti specificiranja bilo koje definicije poslovnog procesa (engl. workflow) zajedno sa jasno definisanim modelom konteksta.
- Definisani opšti postupak i pravila mapiranja modela između elemenata metamodela opšteg (generičkog) i konkretnog metamodela servisa za razumevanje prirodnog jezika koji četbot u vreme njenog izvršavanja koristi, u cilju prilagođavanja četbota novim servisima, odnosno pružanju nezavisnosti od samo jednog servisa. Na ovaj način se stiče nezavisnost od bilo kog servisa za razumevanje prirodnog jezika, čime se rešenje neće vezati za jedan servis već će se predloženim mehanizmom obezbediti lako mapiranje i prebacivanje sa jednog na drugi novi i/ili stari servis.
- Pregled najpopularnijih servisa (platformi) za razumevanje prirodnog jezika, kao i opis detaljne specifikacije procesa obrade i komunikacije sa drugim sistemima.

- Nezavisna komunikacija sa eksternim poslovnim servisima kao što su npr.: poslovni ili bankarski sistemi, kontakt centri itd.
- Proces dinamičke adaptacije generisanjem modela mapiranja na osnovu metamodela i pravila mapiranja.
- U radu je definisana metodologija razvoja četbota ilustrovana kroz slučajeve korišćenja.

Nakon sprovedenog istraživanja dobio se rezultat koji ukazuje da su se postavljene hipoteze dokazale. Objavljivanjem naučnih radova u istaknutim međunarodnim i nacionalnim časopisima, kao i časopisu od nacionalnog značaja, potvrđeni su rezultati istraživanja od strane naučne zajednice:

- Rade Matić, Miloš Kabiljo, Miodrag Živković, Milan Čabarkapa: *Extensible Chatbot Architecture Using Metamodels of Natural Language Understanding*. Electronics 2021, 10,2300.
<https://doi.org/10.3390/electronics10182300> (kategorija M22) [81]
- Miloš Kabiljo, Marijana Vidas-Bubanja, Rade Matić, Miodrag Živković: *Education System in the Republic of Serbia under COVID-19 Conditions: Chatbot-Academic Digital Assistant of the Belgrade Business and Arts Academy for Applied Studies*, Knowledge International Journal, 43(1), 2020, pg. 25 - 30. Retrieved from <http://ikm.mk/ojs/index.php/KIJ/article/view/4855> ISSN: 187-923X (print) i ISSN 2545-4439 (kategorija M33) [21]
- Rade Matić, Miloš Kabiljo, Milen Janjić, Miodrag Živković, Milan Čabarkapa: *ČETBOT – Digitalni asistent zasnovan na WEAVER platformi*, Info M 70/2020, Serbia, Belgrade, (kategorija M53). [51]
- Latha, R. S., Balaji, B. S., Bacanin, N., Strumberger, I., Zivkovic, M., Kabiljo M. *Feature Selection Using Grey Wolf Optimization with Random Differential Grouping*. Computer Systems Science and Engineering, 43(1), 317–332. 2022 (kategorija M22)

1.3. SADRŽAJ DISERTACIJE

Uvažavajući osnovna pravila i principe pisanja doktorske teze, struktura rada je pratila plan istraživanja. Disertacija sadrži uvod, osam poglavlja i zaključak.

Predmet i ciljevi istraživanja doktorske teze su dati u uvodnom poglavlju. Odmah nakon toga su postavljene hipoteze koje su se ispitivale, očekivani rezultati i doprinosi teze. Na kraju prvog poglavlja se izlaže struktura rada.

U okviru drugog poglavlja daje se pregled relevantnih oblasti zajedno sa osnovnim definicijama četbota, njihova klasifikacija i implementacija razvoja istih. Ovo poglavlje takođe sadrži podatke o opštoj taksonomiju modela koji preovlađuju u razvoju četbota.

Treće poglavlje se detaljnije bavi obradom prirodnog jezika sa posebnim osvrtom na razumevanje prirodnog jezika.

Četvrto poglavlje daje osnovne smernice za razvoj četbota kroz opštu referentnu arhitekturu i predloženu metodologiju razvoja četbota. Predložena metodologija je rezultat kombinacije različitih metodoloških pristupa. Celokupan proces razvoja adaptivnog četbota zasnovan na NLU je predstavljen kroz dve faze: projektni i aplikacioni inženjering. Shodno prethodno navedenom, metodologija razvoja četbota je prikazana dualnim životnim ciklusom prikazanim.

Shodno prethodnom poglavlju u sledećem poglavlju se opisuje projektni inženjering. Ovo poglavlje daje predlog jednog novog aplikacionog okvira (engl. Framework) za razvoj četbota. Zatim se detaljno prikazuje logička arhitektura kao i konkretna platforma koja predstavlja konverzacionu platformu za razvoj četbota zasnovanu na veštačkoj inteligenciji. U ovom poglavlju se takođe detaljno opisuje struktura svih komponenti koji čine ovu arhitekturu. Takođe, ovde se opisuje originalni opšti metamodel servisa za razumevanje prirodnog jezika. Zatim se do detalja opisuju metamodeli najrelevantnijih proizvođača ovih servisa. I najzad, daje se opšti postupak i odgovarajuća pravila mapiranja između opšteg metamodela i metamodela specifičnih servisa za potrebe lakšeg prelaska na bilo koji NLU servis.

Šesto poglavlje je posvećeno opisu aplikacionog inženjeringa tj. primene definisanog metodološkog okvira u konkretnom domenu (obrazovanje), preko čega će se izvršiti validacija predložene metodologije i arhitekture. Ovde se takođe objašnjava način NLU učenja, testiranje i validacija svih modelovanih scenarija.

Sedmo poglavlje se nadovezuje na šesto sa studijom slučaja konkretne primene platforme za izradu četbota ADA (Akademska Digitalna Asistentkinja) koja je implementirana u Beogradskoj akademiji poslovnih i umetničkih strukovnih studija. Daće se evaluacija ADA četbota kroz različite analize i opise jednostavnih i složenih scenarija koja su realizovana tokom ove implementacije.

Poređenje metodološkog pristupa sa postojećim rešenjima razvoja četbota sa posebnim osvrtom na kontekst i zavisnost od NLU servisa je opisano u osmom poglavlju. Poređenje je urađeno prvenstveno imajući u vidu opšte karakteristike i komponente arhitekture za razvoj četbota.

Deveto poglavlje je posvećeno zaključcima, naučnim i stručnim doprinosima, i završava se predlogom potencijalno daljih pravaca istraživanja.

Poglavlje deset, jedanaest i dvanaest ukazuju na korišćenu literaturu, indeks slika i prikaz tabela.

2. ČETBOT

U ovom poglavlju se opisuje pojam, oblast i klasifikacija četbota, kao i prednosti uvođenja četbota u poslovne organizacije. Takođe se daje kratka istorija četbota i tehničke specifikacije implementacije četbota.

2.1. DEFINICIJA ČETBOTA

U naučnoj literaturi, akademskim člancima i zbornicima sa konferencije analizirani su različiti aspekti istorije, tehnologije i primene četbota [1, 2]. Decenijama unazad istraživači i praktičari pokušavaju da poboljšaju svoje veštine u dizajniranju korisničkog interfejsa u interakciji između čoveka i računara (engl. Human Computer Interaction - HCI). To znači da se teži ka interakciji zasnovanoj na prirodnom jeziku. U njima se interakcija sa digitalnim sistemima ne odvija preko skrola, pomeranja ili kliktanja dugmića, već preko tekstova napisanih i/ili izgovorenih prirodnim jezikom. Ovo je naročito vidljivo u razvoju četbotova (konverzionih agenata), odnosno, softverskog koda koji služe kao korisnički interfejs na prirodnom jeziku za potrebe dobijanja podataka ili obavljanja usluga. Četbot (engl. chatbot) predstavlja računarski program koji simulira ljudsku konverzaciju ili ćaskanje, koristeći ili ne koristeći veštačku inteligenciju (engl. Artificial Intelligence - AI), i učestvuje u dijalogu sa čovekom koristeći prirodni jezik [3]. Četbot može razumeti pisani tekst kao i glasovne poruke i u velikoj meri može interpretirati njihovo značenje.

Četbot je platforma koja koristi obradu prirodnog jezika (NLP) kao deo veštačke inteligencije i koja može da odgovori na sva pitanja i rešava probleme korisnika i tako poboljšava interakciju između korisnika i računara. Ovaj podskup veštačke inteligencije daje mogućnost analize unetih korisničkih poruka i fraza. Međutim, ova analiza se ne radi jednostavnom pretragom ključnih reči, već se radi sa potpunim razumevanjem prirodnog jezika i konteksta cele poruke. Zapravo, funkcija četbota se zasniva na razumevanju

zahteva bez obzira na jezik ili strukturu rečenice koja je napisana ili izgovorena [4]. Na osnovu obrade rečenice uz pomoć NLP-a i uz primenu odgovarajućeg algoritma, četbot dolazi do adekvatnog odgovora koji se prosleđuje korisniku.

Četbot je softver koji služi kao korisnički interfejs koji koristi (prepoznaje) prirodni jezik za potrebe dobijanja podataka ili obavljanja usluga. Osim toga četbot omogućava jednostavne višestruke dijaloge, upravlja tokom konverzacije i pri tome treba uvek da uzima odgovarajući kontekst iz istorije konverzacije. Danas četbot može da izvršava mnoge funkcije mobilnih aplikacija ili veb sajta, sve u okviru konverzacije putem aplikacija za razmene poruka, a bez potrebe da korisnik mora da instalira ili preuzme nove aplikacije.

Ako se fokusiramo na sve definicije četbota, videćemo da se one generalno opisuju kao računarski programi [5], alati ili softverski sistemi koji su sposobni da simuliraju ljudsku konverzaciju koristeći prirodni jezik kao i tehnike veštačke inteligencije [6]. Pored toga, četbot se povremeno opisuju i kao glasovni ili tekstualni interfejsi koji obezbeđuje pristup skupu usluga ili funkcionalnosti. Sve u svemu, zbog doslednosti i jasnoće, predlažemo da se izgradi koncizna, ali iscrpna zajednička definicija konverzacijskih agenata i četbota, koji će se koristiti kao sinonimi u ovom istraživanju. Jedna sveobuhvatna definicija bi mogla da bude: četbotovi (konverzacioni agenti) su sistemi za dijalog zasnovani na softveru, dizajnirani da simuliraju proces ljudske konverzacije obradom i generisanjem podataka prirodnog jezika putem tekstualnog ili glasovnog interfejsa za pomoć korisnicima u postizanju specifičnog cilja [6].

Četbotovi, konverzacioni agenti, sistemi za dijalog, digitalni asistenti i virtuelni asistenti su neki od termina koji se koriste u naučnoj literaturi [5] da opišu sisteme zasnovane na softveru koji su sposobni da obrađuju podatke na prirodnom jeziku kako bi simulirali pametan proces konverzacije sa ljudima [7]. Ovi četbotovi su izgrađeni i vođeni širokim spektrom tehnika različite složenosti, od tradicionalnih, unapred kodiranih algoritama, do novih algoritama adaptivnog mašinskog učenja [8]. Obično primenjivani kao sistemi orijentisani na korisničke usluge, dizajnirani su da pomognu korisnicima da postignu određeni cilj zasnovan na osnovu njihovih ličnih potreba [9]. U tom cilju, oni autonomno generišu poruke na prirodnom jeziku za interakciju i komunikaciju sa korisnicima oponašajući pravo ljudsko biće [10, 11].

Bez obzira da li ga nazivate četbot, digitalni asistent ili konverzacioni agent, osnovni koncept je isti: postići odgovarajući rezultat razgovarajući sa mašinom putem prirodnog jezika. Neki četbotovi su bez veštačke inteligencije, dok drugi imaju tako složenu veštačku inteligenciju da uspeju da dešifruju nameru korisnika. Dizajniranje efektivnog četbota znači razumevanje njihove osnovne arhitekture i koja vrsta inteligencije je potrebna.

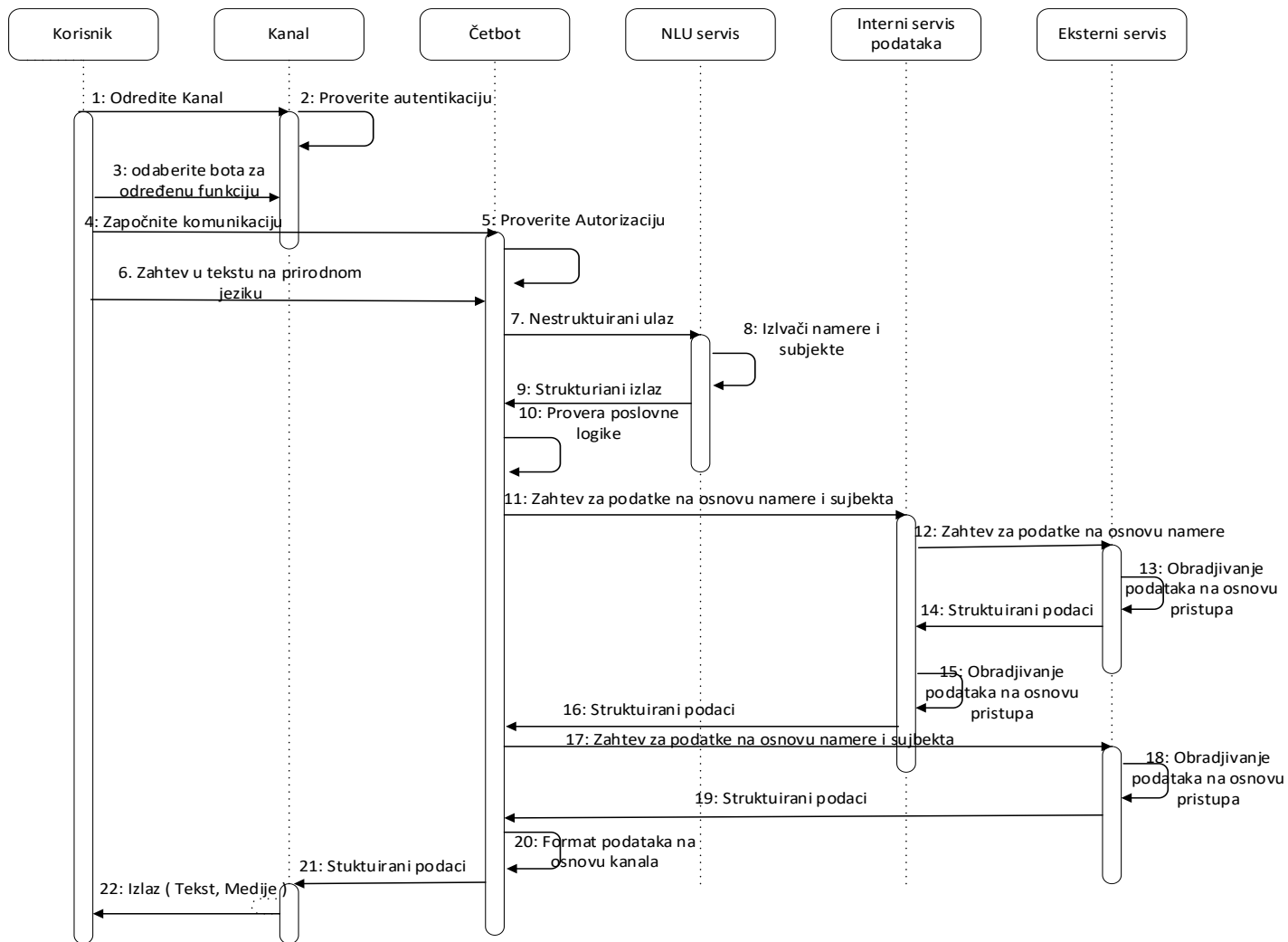
2.1.1. Osnovni pojmovi u razvoju četbota

U ovoj tezi dijagram slučajeva korišćenja se koristi da opiše interakcije korisnika sa četbot sistemom. Slučaj korišćenja je nešto što obezbeđuje neke merljive rezultate korisniku ili spoljašnjem sistemu. Slučaj korišćenja je jedan ili više scenario (situacija) gde se četbot koristi da ispuni jedan ili više korisničkih zahteva. On prikazuju samo deo funkcionalnosti koju obezbeđuje četbot. Slučajevi korišćenja utiču i vode sve ostale elemente u projektovanju. Slučaj korišćenja precizira ŠTA četbot treba da radi, a ne KAKO radi. On takođe predstavlja skup sekvenci događaja. Jedna sekvenca događaja se naziva SCENARIO (definicija scenarija). U razvoju četbotova čest naziv za slučaj korišćenja je priča (engl. story) ili scenario (engl. scenario), a slučaj korišćenja se veoma često poistovećuje sa jednim scenarijem. Međutim, slučaj korišćenja može da bude (veoma) kompleksan kao npr. neka vrsta transakcije za koju je obavezno autentifikovanje (prijava ispita, plaćanje računa, zakazivanje sastanka, izdavanje izvoda, itd.). Dakle, slučaj korišćenja može da se sastoji iz više scenarija. Od kojih je najčešće jedan normalan (osnovni) scenario, a ostali su alternativni (mogući izuzeci). KONVERZACIJA predstavlja konkretnu instancu jednog scenarija, odnosno jednu konkretnu konverzaciju između korisnika i četbota. Kao što smo rekli, u praksi slučaj korišćenja može da predstavlja jedan scenario koji se poistovećuje sa jednim pitanjem i jednim odgovorom. Kada korisnik npr. postavi jedno takvo pitanje smatra se da se okinula jedna konverzacija (instanca scenarija) između korisnika i četbota.

Čet sesija je vremenski interval koji korisnik provodi (komunicira) sa četbotom. U toku čet sesije može da se instancira jedan ili više slučajeva korišćenja između korisnika i četbota, čime se evidentira konverzacija i sve poruke između njega i četbota.

U toku čet sesije se takođe evidentiraju sve pokrenute, suspendovane i završene konverzacije (instance scenarija) koje prate svoju definiciju scenarija. Svaka dospela poruka se stavlja u red, posebno obrađuje i prolazi kroz niz faza obrade. Jednostavan dijagram sekvenci koji opisuje interakciju korisnika sa osnovnim komponentima četbota je dat na slici 1.

U nastavku su opisane predložene taksonomije ciljeva i domena četbota koje omogućavaju da se identifikuju sinergije između njih.



Slika 1. Dijagram sekvenci interakcije korisnika sa komponentama četbota

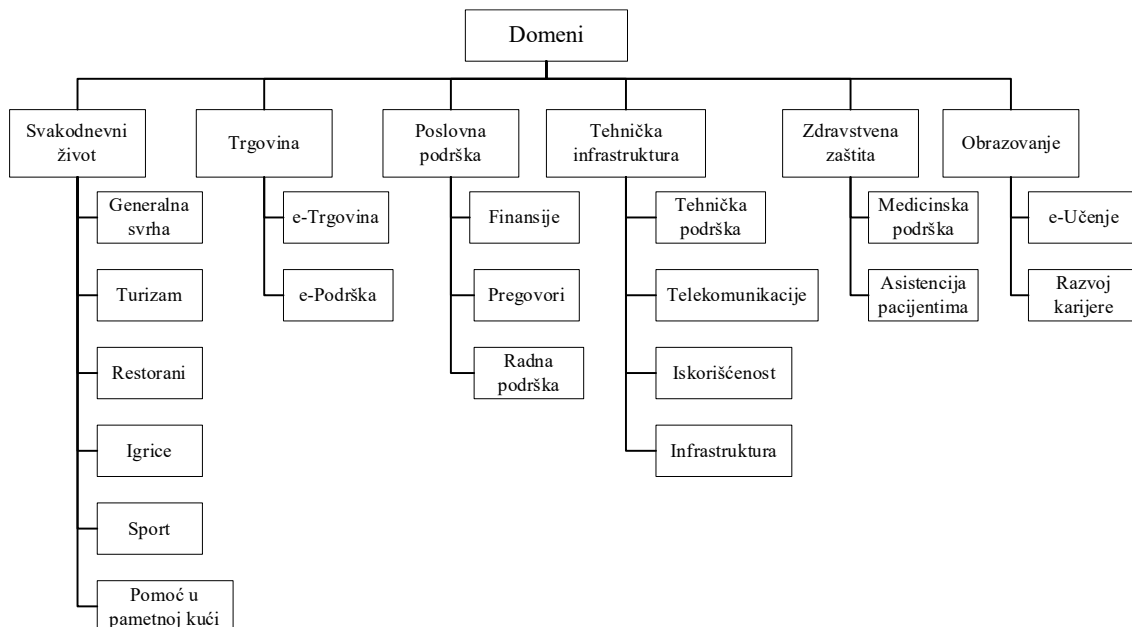
2.1.2. Domeni i oblasti primene četbota

Identifikacija i kategorizacija poslovnih domena i oblasti primene četbota je bitna da bismo razumeli uticaj četbota na okruženje. Diskusija o različitim domenima u kojima su četbotovi imaju relevantno prisustvo iz naučne perspektive je raznolika. Na osnovu istraživanja [6] predlaže se sledeća kategorizacija prikazana slikom 2. koja omogućava da se identifikuje 6 opštih domena ili oblasti istraživanja:

- Svakodnevni život - Najčešći domen među istraživanjima je podrška svakodnevnim životnim aktivnostima. Suština ove oblasti primene je upotreba i integracija četbota za postizanje ličnih ciljeva korisnika u svakodnevnim životnim aktivnostima. Pored opšte namene četbota kao što su pitanja i odgovori (engl. Q&A), ovaj domen uključuje različite oblasti primene specifične za domen kao što su turizam [12], restorani i hrana, igre, sport i pomoć u pametnoj kući.
- Trgovina - Aktivnosti koje se odnose na komercijalne aktivnosti, transakcije i podršku su istaknute kao druga najprisutnija oblast primene. Ovaj domen uključuje četbotove za korisničku podršku uslugama e-korisnika i pomoć u e-trgovini kao što je pomoć za onlajn kupovinu [13].
- Poslovna podrška - Četbotovi u ovom domenu su dizajnirani kao softverski alati za podršku zaposlenima u internim poslovnim procesima. Glavni cilj je da se obezbedi efikasan, intuitivan interfejs za tradicionalne poslovne procese za postizanje poluautomatskih obavljanje takvih procesa. Ovi procesi uključuju procese u IT (otključavanje naloga, resetovanje šifre i sl), finansijske zadatke, timski rad ili radna podrška [14].
- Tehnička infrastruktura – Ovde spadaju četbotovi za tehničko upravljanje i tehničku podršku u podskupu namenskih poddomena koji pružaju pristup i pomoć u upravljanju složenih tehnoloških infrastruktura i tehničkih procesa. Četbotovi o kojima je pisano u literaturama i koji se mogu svrstati u ovu kategoriju su obično vezani za upravljanje složenom tehničkom infrastrukturom i pomoć u softverskom inženjeringu. Ovaj domen pokriva od tehničke podrške za krajnje korisnike do naprednih, poluautomatskih upravljanja pametnim industrijskim infrastrukturama, uključujući telekomunikacione sisteme kao što su autonomni pozivni centri i alati za

premošćivanje jaza u pogledu veće upotrebljivosti softverskih proizvoda ili alata od strane korisnika [15].

- Zdravstvena zaštita - Domen zdravstvene zaštite je takođe značajno prisutan u literaturama [16]. Najčešća područja primene u ovom domenu uključuju alate za podršku zdravstvenim radnicima u medicinskim aktivnostima kao što su prepisivanje lekova ili kontrola i praćenje vitalnih znakova [17], i alati za podršku za pomoć pacijentima kao što je upravljanje terapijom [18].
- Obrazovanje - Slično domenu zdravstvene zaštite, posvećeno istraživanje u oblasti obrazovanja je takođe istaknuto kao relevantna oblast. Obrazovni četbotovi se mogu naći kao alati za e-učenje za pomoć studentima u procesu učenja [19] i kao automatizovani agenti za podučavanje [20, 21]



Slika 2. Domena primene četbota

2.1.3. Ciljevi integracije četbota u softverske sisteme

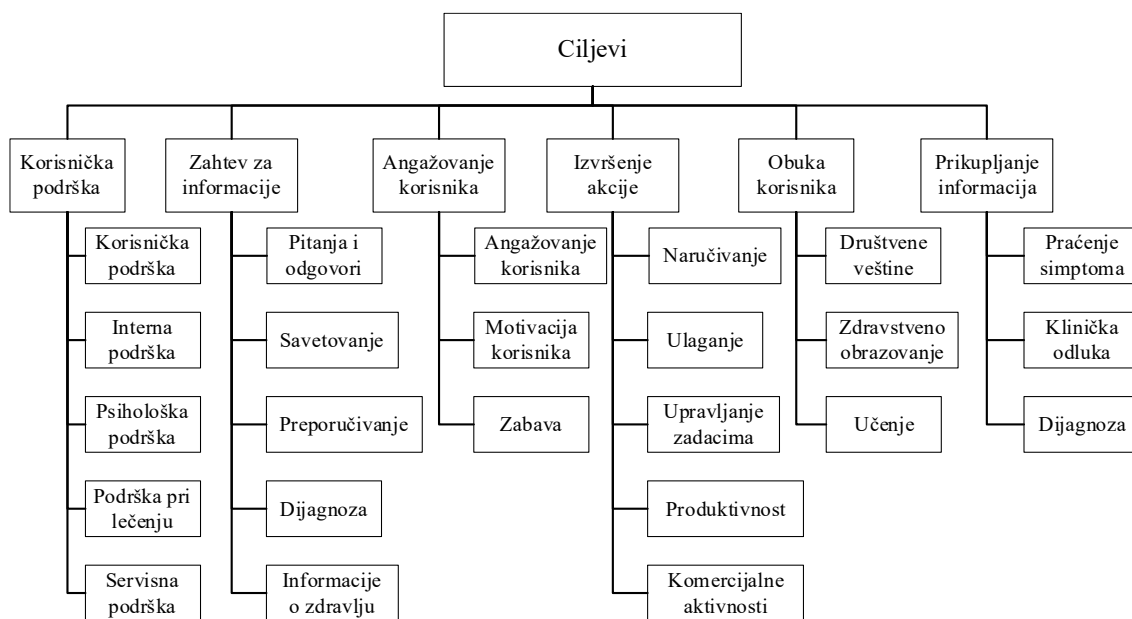
Slika 3. prikazuje opštu kategorizaciju šest glavnih ciljeva integracije četbota u softverske sisteme:

- Korisnička podrška - Najčešći cilj četbotova je proširenje softvera za podršku sa alatom za podršku integracije koja simulira ljudsku komunikaciju u postojećim aktivnostima i procesima, a sve sa ciljem poboljšanja korisničkog iskustva. Ovaj cilj je

prisutan u mnogim kontekstima. Na primer: u domenu trgovine, kao alate za korisničku i servisnu podršku. Zatim u poslovnom domenu, kao poluautomatizovani interfejsi uslužnih procesa i za internu podršku. U domenu zdravstvene zaštite, cilj je pomoći kroz terapijske pomoći (psihoterapija) i kroz podršku lečenja [22].

- Zahtev za informacijama – Najčešća kategorija ciljeva je da su četbotovi dizajnirani za upite o proizvodu (robi) koji jednostavno pristupaju bazama znanja (opštima ili specifičnim). Četbotovi za traženje informacija uključuju *Q&A* četbotove [23] i eksperte za složene domene kao što su autonomna dijagnoza ili pristup informacijama u vezi zdravlja. Oni takođe uključuju svakodnevnu upotrebu, specifičnu za domen kao što su preporuka restorana i savetnik za turizam. Informacije se direktno traže u skupu izvora podataka i dostavljaju krajnjem korisniku ili se obrađuju putem informacija agregiranih algoritama da bi se dali personalizovani odgovori na osnovu dodatnih podataka (npr. informacije o korisniku i ostali kontekstni podaci).
- Angažovanje korisnika - Ovaj angažman se odnosi na razne domene, uključujući angažovanje kupaca u e-trgovini [24], motivaciju korisnika za specifičnu aktivnost kao što je angažovanje pacijenata u fizičkoj aktivnosti i angažovanje korisnika za razgovore usmerene na zabavu.
- Izvršenje akcije - Četbot se takođe koriste kao tekstualni ili govorni interfejsi dajući pristup skupu funkcija iz integrisanih softverskih sistema ili softverskih sistema treće strane preko poluautonomnog komunikacionog protokola. Ovaj cilj se generalno posmatra kao međudomenski cilj, uključujući onlajn naručivanje robe [25], timsku kolaboracija upravljanja zadacima, podrška za produktivnost zadataka u zdravstvu. Zajednička karakteristika među ovim primerima je izvođenje akcija koristeći prirodni jezik (tekst ili govor) kao most interakcije između korisnika i sistema.
- Obuka korisnika - Široka zajednica istraživanja četbota se fokusira na evaluaciju, obuku i poboljšanje korisničkih veština u oblasti specifičnoj za domen. Ovaj cilj je posebno istaknut u obrazovnom polju [26], gde je učenje identifikovano kao primarni cilj. Zdravstvena zaštita i medicinska istraživanja fokusiraju se na obuku u smislu obuke socijalnih veština za poremećaje mentalnog zdravlja, obuke psihoterapije poput svesnosti, pa čak i korišćenje virtuelnih pacijenata kao alata za učenje.
- Prikupljanje informacija - Ovaj cilj je prvenstveno posvećen domenu zdravstvene zaštite, sakupljanju i obradi informacija od korisnika (bilo kroz interakciju prirodnog

jezika ili, u nekim slučajevima, putem kontekstnih podataka). Ovaj cilj se uglavnom primećuje u aktivnostima vezanim za zdravstvenu zaštitu kao što je praćenje simptoma [27], dijagnoza, napredna klinička odluka ili podrška trijaži. Implicitni i eksplicitni podaci se kombinuju za izgradnju i istraživanje složenog znanja o profilisanju korisnika kako bi poboljšali i podržali druge procese.



Slika 3. Ciljevi četbota

Zaključak je da je obuka korisnika isključivo pokrivena obrazovanjem i zdravstvenom zaštitom (npr. rehabilitacija pacijenata). Prikupljanje informacija se uglavnom odnosi na zdravstvenu zaštitu (npr. praćenje simptoma). Izvršenje akcija je dominantno u trgovini (npr. kupovina) i poslovnoj podršci (npr. zakazivanje). Korisnička podrška, zahtev za informacijama i angažovanje korisnika uglavnom se navode kao ciljevi za više domena. Korisnička podrška, angažovanje korisnika i obuka korisnika zahtevaju aktivno učešće korisnika u dizajnu i validaciji sistema. Shodno tome, potreba za posvećenim istraživanjem u smislu korisničkog iskustva, angažovanja korisnika i HCI je pojačana ovim rezultatima.

2.2. MOTIVACIJA

Svet se neprestano menja, a naročito se promenio način na koji ljudi komuniciraju. Potpuno smo opušteni kad komuniciramo putem kratko kućanih interakcija, i prilično smo opušteni s obzirom da istovremeno imamo nekoliko asinhronih razgovora. Mi smo već u svetu u kojem ne možemo da prepoznamo konverzacione partnere. Neki od njih su ljudi, neki su četbotovi. Međutim, sve više nas to možda neće ni interesovati, niti ćemo se brinuti oko toga, jer je bitno da smo zadovoljni sa konverzacijom ili akcijom koju smo hteli da izvedemo. Komunikacione platforme za razmenu poruka i broj njihovih korisnika veoma brzo raste. Samo u Srbiji postoji preko pet milion korisnika *Viber* platforme. *Facebook Messenger* je odavno prešao preko milijardu korisnika svakog meseca, a njihov broj raste brže od samog *Facebook*-a. Dakle, velika promena je dostupnost platformi za razmenu poruka koja se čini skoro savršenim okruženjem za četbot. Više nema potrebe za skidanjem, instaliranjem i otvaranjem aplikacije samo za naručivanje hrane ili za kupovine karte. To je samo još jedan aspekt današnjeg već povezanog multitasking sveta.

S druge strane svi imamo uglavnom loša iskustvo sa korisničkim servisom preko telefona. U dosta slučajeva to je veoma frustrirajuće koristiti jer moramo da prođemo kroz više opcija u meniju pre nego što se povežemo sa ljudskim agentom. I dalje vidimo da su svi provajderi usluga ovaj sistem primenili kao pristup za njihov korisnički servis, čak i ako ne postiže njihov osnovni cilj smanjivanja i skraćivanja poziva korisničkom servisu. Dodatna frustracija je činjenica da je agent dostupan samo maksimalno 8 sati, pet dana u nedelji. Tako da umesto dobijanja ograničenih informacija, uvek je bolje da smo upućeni na pravo mesto ili pravog agenta da reši naš zahtev i pruži sve informacije pored predloga za buduće akcije. Ovde četbot rešenje ima veliku ulogu. U osnovi, sve više ljudi koristi sve više vremena na platformama za razmenu poruka, što ih čini komunikacionim platformama budućnosti.

Na sve prethodno navedeno treba dodati i pandemiju KOVID-19 koja je imala i još uvek ima veliki uticaj na svet i globalnu ekonomiju. "Globalna ekonomska kretanja postala su veoma negativna 2020. godine, što je izazvalo najgore posledice od Velike

depresije 1930-ih. Projekcije ukazuju da je preko 170 zemalja 2020. godine doživela pad GDP-a po stanovniku, s tim da je kriza najviše pogodila zemlje u razvoju i one u tranziciji” [28]. Sa jedne strane, ona je ugrozila kontinuitet i kvalitet rada poslovnih sistema, a sa druge strane je otvorila vrata neophodne digitalne transformacije poslovanja i potrebu brže implementacije poslovanja i učenja na daljinu.

Gartner, koji je svetski konsultantski i istraživački lider u svetu, predviđa odličnu budućnost četbotova u vezi operacija vezanih za usluge i podršku klijentima. Oni takođe navode da: Četbot otvara nove kanale za pružanje javnih usluga (poboljšavajući ih). Zadovoljstvo klijenata se povećava tri puta, a troškovi se trostruko smanjuju kad se četbot koristi u organizaciji. Organizacije koje su uvele četbota izveštavaju o smanjenju poziva, ćaskanja i/ili upita putem mejla do 70 odsto [29]. Upotreba četbotova poslednjih godina brzo se razvila u brojnim oblastima, uključujući korisničku podršku, e-trgovinu, marketing, sisteme za pružanje podrške, obrazovanje, zdravstvenu zaštitu, kulturno nasleđe i zabavu. U svim ovim oblastima četbotovi su se pokazali veoma korisnim u različitim kontekstima za automatizaciju zadataka i poboljšanje korisničkog iskustva. Dodatna predviđanja govore da će do kraja 2022. godine 80% kompanija koristiti četbotove, a banke će moći automatizovati do 90% svojih interakcija s korisnicima [30]. Predviđa se da će globalno tržište četbotova do 2024. godine dostići 2 milijarde dolara [31]. Izvršni direktor kompanije *Microsoft* uporedio je predviđenu tranziciju na četbotove i korisničke interfejse prirodnog jezika na prethodne revolucije kao što je uvođenje grafičkog korisničkog interfejsa, Interneta i mobilnog Interneta. *Facebook*-ov kreator *Mark Zuckerberg* proglasio je četbotove kao rešenje za izazov preopterećenja aplikacija [9]. S obzirom da je ova vizija konverzionih interfejsa istinita, velike se promene se očekuju u oblasti interakcije čovek računar.

Četbot se koristi u različitim vrstama aplikacija kao što su usluge vezane za e-trgovinu, tehničku podršku, učenje itd. Iako je istorija razvoja veštačke inteligencije počela još 1950 godine sa poznatim Turingovim testom [32], koji se ogleda u sposobnosti mašine da ispolji ponašanje isto ili bolje od čoveka, industrija četbota je i dalje u ranoj fazi, ali njihov razvoj i primena u zadnje dve godine doživeo je veliku ekspanziju. Trenutno, četbotovi su najviše dizajnirani i razvijeni za potrebe četa na veb stranici i za mobilne aplikacije za slanje poruka.

S obzirom da aplikacije za razmenu poruka postaju najčešći način koji ljudi koriste u komuniciranju, svaki biznis će na kraju imati svog četbota. Četbotovi i njihovi odgovori će biti brži od veb stranica i mobilnih aplikacija jer neće biti instalacija, kliktanja i pretrage po četbotu kao na aplikacijama. Četbotovi mogu pružiti korisnicima brzu i praktičnu podršku koja odgovara posebno na njihova pitanja, i to je razlog zašto je najčešća motivacija za produktivnost korišćenja četbota, dok su drugi motivi zabava, društveni faktori, dobre i ažurne informacije. Umesto stvaranja aplikacije za pametnu mašinu slične čoveku, radi se o stvaranju efikasnih digitalnih pomoćnika koji mogu da pruže informacije, odgovore na pitanja, razgovaraju o određenoj temi ili izvrše zadatak [33]. Četbot je tipičan primer sistema veštačke inteligencije i jedan od najelementarnijih i najrasprostranjenijih primera inteligentne interakcije čoveka i računara [34].

Danas četbot može obavljati mnoge funkcije mobilnih aplikacija ili veb lokacija, sve u okviru razgovora putem komunikacionih aplikacija, bez potrebe da korisnik instalira ili preuzima nove aplikacije. Četbotovi su sada ugrađeni u popularne digitalne pomoćnike kao što su *Siri*, *Cortana*, *Aleka*, *Google Assistant* itd. Više nema potrebe za preuzimanjem, instaliranjem ili otvaranjem aplikacija samo za obavljanje operacija poput naručivanja proizvoda. To je samo još jedan aspekt već povezanog sveta sa više zadataka. Sadašnje veliko interesovanje za četbotovima je podstaknuto razvojem dve nove tehnologije:

- Tehnologija razmene poruka se brzo proširila u poslednjih nekoliko godina i postala jedna od najčešće korišćenih usluga pametnog telefona. Ove platforme servisa za razmenu poruka (*Viber*, *Facebook Messenger*, *Instagram*, *WhatsApp*, itd.) sada uključuju funkcionalnosti kao što su informisanje, podrška, plaćanje, naručivanje i rezervacija, što bi inače zahtevalo odvojenu aplikaciju ili veb sajt.
- Veštačka inteligencija je takođe napravila značajne korake napred u poslednjih nekoliko godina. Tehnika kao što je mašinsko učenje iskoristila je ogromne količine podataka i jeftinu procesorsku moć kako bi dramatično poboljšali kvalitet razumevanja i donošenja odluka.

Dodatni motiv za razvoj četbota, odnosno razvoj jedne adaptivne arhitekture koja je nezavisna od NLU proizvođača je bila potreba da se obezbedi podrška za Srpski jezik

koji ne pružaju svi NLU servisi i potreba za jednostavnim prebacivanjem sa jednog na drugi NLU servis. Zatim se takođe javila potreba smanjenja troškova jer su se pojavili neki proizvođači koji ne naplaćuju NLU servise, a oni su davali veoma dobre procenete preciznosti pronalaženja namere i ekstrakcije entiteta što je krucijalno u obradi prirodnog jezika za potrebe i izvršavanja akcija i zadataka koje krajnji korisnik daje četbotu.

2.2.1. Prednosti uvođenja četbota

Četbotovi i njihovi odgovori su brži od veb stranica i mobilnih aplikacija jer nema instalacija, pritiskanja dugmića i pretrage po četbotu kao na aplikacijama. Osim toga treba navesti sledeće bitne razloge zbog kojih je potrebno uključiti četbota u strategiju razvoja poslovanja u budućnosti:

- Brža usluga: svako smanjivanje vremena čekanja klijenata uvek poboljšava iskustvo korisnika. Klijentima se pomaže da se reše tradicionalne komunikacije putem telefona i mejla koje mogu biti dugotrajne i dosadne. Umesto da klijente ostavljaju na telefonskom čekanju, može se koristiti četbot kako bi se odmah bavio jednostavnim pozivima i zahtevima klijenata.
- Veća dostupnost klijentima: obezbeđuje se visoka dostupnost klijentima i to 7/24/365 dana. Klijenti žele usluge izvan normalnog radnog vremena od 9 do 17 sati. Četbot je dostupan 24 sata dnevno tokom cele godine. Gde god da se nalazimo internet je veoma dostupan samim tim i četbot. Internet je jeftiniji od telefonskog poziva ka npr. bankama ili akademijama da bi se dobila neka informacija, a pretrage na sajtu su veoma frustrirajuće.
- Veća efikasnost: Četbot efikasnije služi klijentima jer može da premosti jaz između velikog broja klijenata koji traže informacije i ograničenih resursa koje treba da pružaju takvu vrstu pomoći jer omogućava istovremeno opsluživanje više hiljada klijenata kao i manji broj grešaka.
- Veća vidljivost: zbog korišćenja više komunikacionih kanala odnosno platformi za slanje poruka veća je vidljivost i kvalitet usluga.
- Jednostavniji način komunikacije i lakše korišćenje: S obzirom da četbotovi koriste komunikacione kanale kao što su npr. *Viber*, *WhatsApp* i *FB messenger*, nema potrebe

za učenjem njihovog korišćenja. Ako znamo kako da ih koristimo, onda znamo kako da koristimo četbot. Dok u slučaju novih aplikacija učenje može da predstavlja problem i oduzima vreme.

- Povećanje baze mišljenja klijenata: lakše je privući korisnike na komunikacione platforme za razmenu poruka nego ih terati ili nagovarati da instaliraju neku aplikaciju na mobilni ili računar. Na ovaj način je lakše doći do mišljenja klijenata putem raznih anketa ili odlučivanja. Kao što smo već rekli samo u Srbiji ima preko pet miliona korisnika Vibera i broj stalno raste.
- Bolja interakcija sa klijentima: četbot izlaže interfejs preko kojih se mogu slati direktne poruke ka klijentima (vezane za razna obaveštenja, savete, ponude, targetirane reklame itd.), kao i interfejs za povlačenje podataka o korisniku i konverzacijama koje je imao sa četbotom. Samim tim četbot je koncipiran kao *in/outbound* platforma.
- Optimizacija: relaksiranje od zadataka koji se ponavljaju, a ljudski resursi mogu da se preusmere/prekvalifikuju na odgovornije i kreativnije poslove jer dolazi do rasterećenja kontakt centra. Ljudi često misle da će uvođenjem veštačke inteligencije izgubiti radna mesta, a u stvari dobijaju kreativnije poslove kao npr. poslove učenja četbota odnosno veštačke inteligencije. Oni su ti koji su domenski eksperti i preko administrativnog alata oni uče četbot šta i kakvo pitanje najčešće postavljaju korisnici.
- Integracija: lako se može izvršiti duboka integracija sa osnovnim sistemima (interni ili eksterni informacioni sistemi) u cilju da se njihove informacije lako izlože korisniku na komunikacioni kanal (npr. viber).
- Zamena većeg dela kontakt centra: vrši se integracija sa agentima kontakt centra i objedinjavanje sa njihovim servisima u skladu sa mogućnostima kontakt centra odnosno ako poseduju otvoren API. Ovo takođe znači da kada četbot dođe do poruke koju ne razume on je prosleđuje API kontakt centru. Konverzacija se prosleđuje od API kontakt centra ka operateru koji nastavlja razgovor. Moguće je da neki scenario zahteva agenta po definiciji scenarija. Ovaj zahtev zavisi od softvera u kontakt centru tj. da li ima otvoren API.
- Četbotovi su omni-kanali: klijenti mogu dobiti podršku kroz više kanala, a lako se prebacuju sa kanala na kanal i dobiju isti odgovor bez obzira na kanal (npr. sa *Viber* na *FB messenger* i obrnuto).

2.3. KRATKA ISTORIJA ČETBOTA

Alan Turing je 1950. godine predložio tzv. Tjuringov test („Mogu li mašine da misle?“), i od tada je krenula opšta ideja o četbotu [32]. Uvođenje četbot tehnologije započelo je 1966. godine računarskim programom poznatim pod imenom *ELIZA* [4], razvijen od strane grupe profesora sa MIT univerziteta. *ELIZA* je mogla da oponaša ljudski razgovor pokušavajući da odgovori na pitanje korisnika uparivanjem skriptovanih odgovora, tj. jednostavnim podudaranjem uzoraka. Razvio ga je Joseph Weizenbaum. Osmišljen je tako da oponaša psihoterapeuta i imao je bazu znanja u ovom domenu. Ona je za to vreme bila u mogućnosti da prođe Tjuringov test inteligencije.

PARRY [36] je predstavljao neobjašnjivo simuliranu osobu sa paranoidnom šizofrenijom razvijen 1972. godine. *PARRY* je bio mnogo ozbiljniji i napredniji od *ELIZA*-e, a bio je opisivan i kao *ELIZA* sa stavom.

JABBERWACKY je razvijen 1988. godine i predstavljao je jedan od prvih pokušaja da se kreira veštačka inteligencija kroz ljudsku interakciju. Uglavnom baziran na formi zabave, on je imao cilj da se pomeri sa tekstualno baziranog sistema na sistem koji je uglavnom zasnovan na govoru.

DR. SBAITSO je predstavljao program za govor zasnovan na veštačkoj inteligenciji koji je kreiran za MS DOS. Dizajniran je 1992. godine tako da predstavi digitalizovani glas. Međutim, *DR. SBAITSO* je bio daleko od stvarnog živog bića, uprkos tome što je imao ulogu psihologa u interakciji sa korisnicima.

A.L.I.C.E. (engl. Artificial Linguistic Internet Computer Entity) je razvijen 1995. godine [37]. To je bio bot za obradu prirodnog jezika. Ovaj bot je mogao da primeni pravila heuristika koja su se podudarala sa pravilima ljudskog unosa, odnosno mogla je imati konverzaciju. *ALICE* se oslanja na jednostavan algoritam za podudaranje obrazaca sa osnovnom inteligencijom zasnovan na jeziku za označavanje veštačke inteligencije (engl. Artificial Intelligence Markup Language - AIML) [38], koji programerima omogućava da definišu gradivne elemente znanja za četbot.

SMARTERCHILD [39] je inteligentan bot koji je masovno distribuiran kroz SMS mreže 2001. godine. Sa mogućnostima kao što su brzi pristup podacima i zanimljivim

personalizovanim konverzacijama, bio je uzet kao predhodnik za *Apple-ov SIRI* i *Samsung-ov S Voice*.

BOTS FOR MESSENGER je platforma koja je razvijena 2016. godine od strane kompanije *Facebook*, koja omogućava developerima da kreiraju botove koji mogu komunicirati sa *Facebook* korisnicima. Trenutno postoji preko 300.000 aktivnih botova na ovoj platformi.

TAY je takođe dizajniran 2016. godine tako da imitira govor i navike tinejdžerke iz amerike. Ovaj četbot koji je razvijen od strane *Microsoft-a* je brzo razvio groznu paranoju. *TAY* je ugašen nakon 16 sati postojanja i nikad se više nije moglo ništa čuti o njemu.

Tokom godina, drugi četbotovi su uključili savremene i sofisticiranije tehnike kako bi bolje razumeli korisnička pitanja i pružili relevantnije i korisnije odgovore. Danas četbot može da izvršava mnoge funkcije mobilnih aplikacija ili veb sajta, sve u okviru konverzacije putem aplikacija za razmene poruka, a bez potrebe da korisnik mora da instalira ili preuzme nove aplikacije. Veći broj četbotova se pojavilo početkom dvadesetog veka, dok je zadnjih šest godina došlo do naglog povećanja broja četbotova zbog povećanog tržišta mobilnih telefona i ostalih pametnih uređaja i naravno napretka mašinskog učenja. Četbotovi su sada ugrađeni u popularne u digitalne asistente kao što su: *Siri*, *Cortana*, *Alexa*, *Google Assitent*, *IBM Watson* itd.

SIRI [40] je inteligentni lični asistent, koji je deo *Apple-ovog IOS* sistema čije su karakteristike prirodni jezik, koje koristi kako bi odgovarao na pitanja i izvršavao veb servis zahteve. *SIRI* obuhvata širok spektar komandi, koje uključuju standardne aktivnosti na telefonu, proveravanje informacija, podešavanje rasporeda i alarma, pretragu interneta, navigaciju, itd. Poslednje verzije uključuju glas koji je veoma sličan ljudskom, podržava follow-up pitanja, prevođenje, i mnoge dodatne akcije. *SIRI* je 2011. godine otvorila put za sve ostale četbotove i personalne asistente koji su kasnije nastali.

GOOGLE ASISTENT je razvijen 2012. godine od strane *Google* i podržava tekstualne i glasovne komande radi pretrage mobilnih aplikacija [41]. On koristi NLU kako bi odgovorio na pitanja, davao preporuke i izvršavao akcije koje su mu postavljene i prosleđivao ih veb servisima.

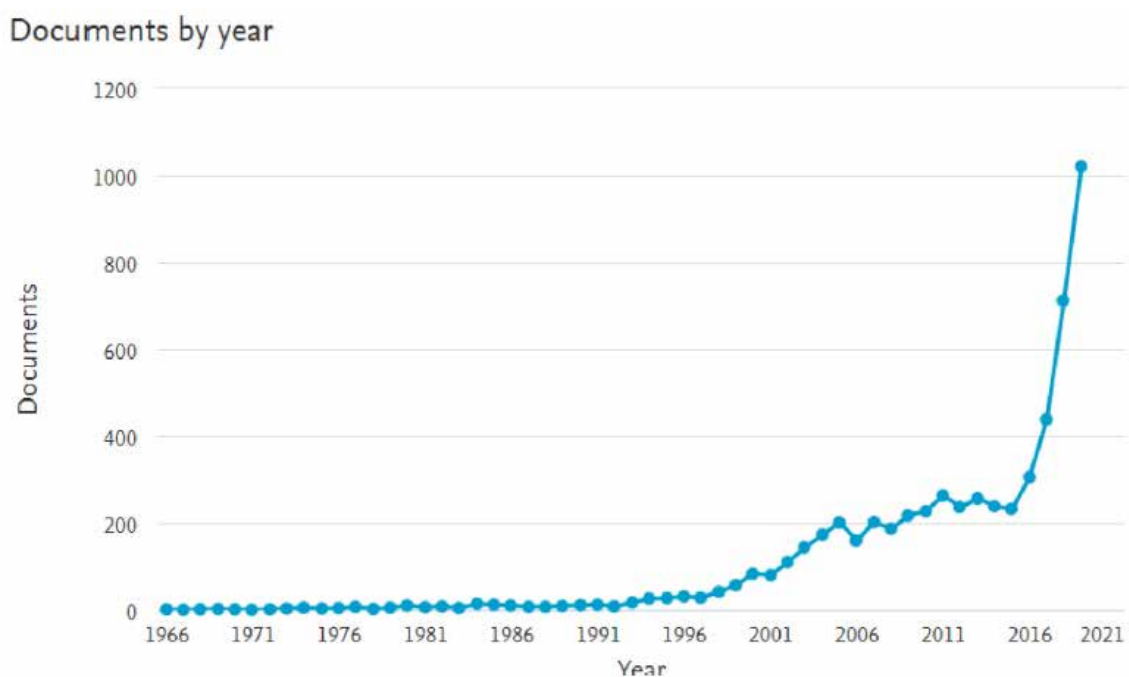
ALEXA [42] je servis koji se koristi od 2015. godine u *Amazon Echo* uređaju i predstavlja do sada finansijski najuspešniji četbot. On je sposoban za glasovnu

interakciju, koristi NLP algoritme kako bi primao, prepoznao i odgovorio na glasovne komande. Korisniku može pomoći pri kupovini, slušanju muzike, kreiranju ankete i mnogim drugim aspektima.

CORTANA [43] je inteligentni lični asistent razvijen 2015. godine od strane kompanije *Microsoft*. Ona može praviti podsetnike, prepoznavati glasovne poruke, i odgovarati na pitanja korišćenjem *Bing* pretraživača.

IBM WATSON [44] je bio specijalno dizajniran 2006. godine da se takmiči u *Jeopardy*-u. Ovo takmičenje je osvojio 2011. godine protiv dva predhodna šampiona. Trenutno *IBM WATSON* koristi NLP i mašinsko učenje kako bi imao uvid u veliku količinu podataka i smatra se veoma preciznim u razumevanju namera i izvršavanja zadataka.

Kao što je prikazano na slici 4 prema *Scopus*-u [45], došlo je do brzog rasta interesovanja za četbotove, posebno nakon 2016. godine. Mnogi četbotovi su razvijeni za industrijska rešenja, dok postoji širok spektar manje poznatih četbotova relevantnih za istraživanje i njihove aplikacije [46].



Slika 4. Rezultati pretrage reči *chatbot* i *conversation agent* po godinama

2.3.1. Ostali primeri četbot rešenja

U današnje vreme, četbotovi su svepristuniji. Ljudi često komuniciraju sa ovim ličnim asistentima na dnevnoj bazi, njihovo korišćenje kao i njihova pomoć je duboko ukorenjena u svakodnevnom životu pojedinaca. Pored ovih opštih asistenata, postoje takođe specijalizovane manje poznate verzije četbotova koje su programirane da reše zadatke koji su vezani za neku specifičnu oblast. Velika većina su uglavnom društveni četbotovi koji su fokusirani na neku oblast. Ove tehnologije su proširene, kao na primer na *Facebook* koji ima svoj API koji može da kreira svog četbota, koje uglavnom koriste poznate ličnosti ili kompanije, koji mogu da odgovore automatski na neka opšte poznata pitanja [47].

Swelly je četbot koji omogućava dobijanje prijateljskih mišljenja na različite teme, čime se pomaže pri donošenju određenih odluka [48]. Korisnici pitaju svoje prijatelje šta da obuku, kako da odgovore na poruku i mnoga druga pitanja. Korisnik može pitati svog prijatelja za mišljenje o nečemu. Međutim, zahvaljući svom viralnom aspektu, aplikacija je postigla visok rast i dostigla preko 2 miliona korisnika.

Švedska banka SEB kreirala je Aidu, kao pratnju postojećem internom četbotu Ameliji, namenjenog zaposlenima. U toku prve tri nedelje korišćenja 700 zaposlenih razgovaralo je preko 4000 puta, tokom kojih je ona rešila veliki broj ukazanih problema bez odlaganja. Njeni glavni zadaci su rad sa zaposlenima i pomoć internoj IT podršci. Aida trenutno rešava oko 13% svih pitanja IT podrške. Takođe, pomaže korisnicima sa problemima oko kartica, nalozima i rezervaciji sastanaka u banci.

XiaoIce je *Microsof*-ov najuspešniji četbot i jedan od tehnički najsofisticiranijih botova [49]. U julu 2014. godine počela je da se koristi na kineskoj društvenoj mreži *Weibo*. Za tri meseca dostigla je pola milijardi razgovora. Rangirana je kao Weibov top influencer sa preko 850 hiljada pratilaca. U proseku, svaka osoba koja je dodala *XiaoIce* pričala je sa njom preko 60 puta mesečno. Ovaj četbot je prošao Tjuringov test za 10 minuta, što znači da ljudi koji pričaju u prvih 10 minuta nisu znali da je u pitanju četbot.

Instalocate je četbot koji pomaže prilikom odlaganja letova i rešavanja pitanja kompenzacije. Uzima 25% udela kada putnik dobije kompenzaciju za odložen, poništen

ili prebukiran let. Za 6 meseci ostvareno je preko 1.1 miliona razmenjenih poruka i ovaj četbot predstavlja jedan od finansijski uspešnih četbotova.

Sa preko 110 hotela otvorenih u skoro 20 zemalja širom sveta Aloft hoteli uvode tehnološki naprednog četbota modernog dizajna, koji pruža jedinstveno iskustvo pod nazivom *Chatbotlr*. *Chatbotlr* je u prvom period bio dostupan preko tekstualnih poruka, kao dodatan način rešavanja pitanja korisničkih zahteva. Sa sledećom generacijom, gosti su mogli koristiti svoj telefon da zahtevaju bilo koju informaciju u bilo koje vreme. *Chatbotlr* postaje sve inteligentniji kroz interakciju sa gostima.

Još jedno rešenje za čekiranje i rešavanje zahteva vezanih za aerodrome, je Carla. Ovaj četbot omogućava korisnicima da se čekiraju na aerodromima i dobiju svoje karte. Razgovor sa njom je poput razgovora sa čovekom, zahtevi se obavljaju brzo i bez potrebe za instalacijom određene aplikacije. Na ovaj način smanjili su svoje troškove i pružili usluge većem broju putnika uz minimalan trud. Sve se obavlja preko *FB Messenger*.

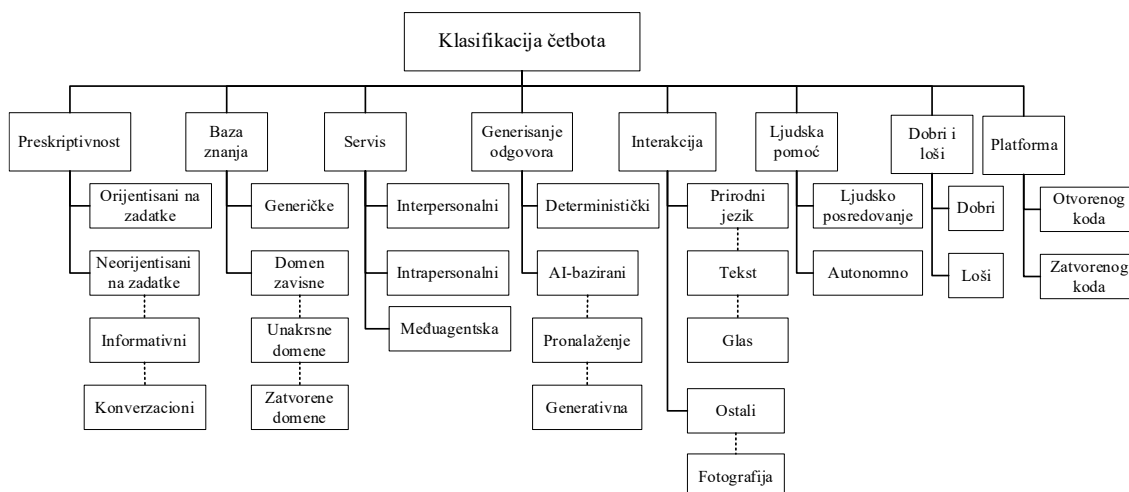
Odličan primer implementacije četbota u Srbiji je Rajfajzenova elektronska asistentkinja – REA [50], digitalna asistentkinja *Raiffeisen* banke zasnovana na platformi *Weaver* [51] kompanije SAGA [52]. Preko ovog četbota klijent može da postavi bilo koje pitanje u vezi banke, proveriti stanja na računu, da zakaže sastanak, informacije o najbližem bankomatu, izvrši određene finansijske transakcije i sl. REA spada u zatvorene domene jer pruža trenutnim i budućim klijentima banke lakše i brže informacije vezane za usluge banke.

2.4. KLASIFIKACIJE ČETBOTA

Tokom poslednjih nekoliko godina, oblast četbota postala je toliko dinamična sa dolaskom novih tehnologija da je precizna klasifikacija četbota postala subjektivna u odnosu na obim njihove upotrebe. Jedan predlog klasifikacije četbota je dat na slici 5. Četbotovi se mogu klasifikovati u različite kategorije na osnovu nekoliko kriterijuma, na primer. način interakcije, domen znanja, njihova upotreba i tehnike dizajna (metoda generisanja odgovora) koje se obično koriste u izgradnji četbota. Ovi kriterijumi mogu

uključivati osnovnu filozofiju dizajna četbota ili stepen u kojem kontekst treba da se čuva i razmatra u razumevanju razgovora ili vrste i svrhe razgovora za koji treba da bude dizajniran četbot [6, 53]. Široka klasifikacija se može uraditi na osnovu sledećih kriterijuma:

- Orijentisan ili neorijentisan na zadatak;
- Način interakcije (tekstualni, glasovni);
- Generisanje odgovora;
- Otvorene i zatvorene domene, odnosno opšte (generičke) i domen-zavisne baze znanja;
- Tip odnosa četbota sa korisnikom;
- Potrebna ljudska pomoć ili ne;
- Dobri i loši četbotovi;
- Četbotovi razvijeni na otvorenim ili zatvorenim platformama.



Slika 5. Klasifikacija četbotova

Četbotovi orijentisani na zadatak su definisani kao četbotovi kratkog razgovora dizajnirani da izvrše određeni zadatak iz poznatog podskupa unapred konfigurisanih zadataka koje pokreće konverzacioni proces. Na primer, onlajn četbot za kupovinu je dizajniran da pomogne korisnicima u njihovom procesu kupovine u potrazi za proizvodima i rešavanje pitanja vezanih za nalog. Neorijentisani na zadatke imaju za cilj da simuliraju ljudski konverzacioni proces bez određenog zadatka ili radnje kao glavni cilj njihove interakcije. Četbotovi za slobodno vreme ili zabavu spadaju u ovu kategoriju.

Što se tiče četbotova koji nisu orijentisani na zadatke, neki prave razliku između konverzionih i informativnih četbotova [54]. Dok se konverzionih četbotovi poklapaju sa definicijom neorijentisanih na zadatak, informativni se definišu kao tip četbotova koji nisu orijentisani na zadatak, koji ne slede specifičnu aktivnost ili zadatak koji treba da izvrše, već interakciju i proces razgovora ima svrhu prikupljanja informacija. Četbotovi za pružanje najčešćih pitanja i odgovora za servisnu podršku spadaju u ovu kategoriju. Četbotovi orijentisani na zadatak obično podrazumevaju da je četbot integrisan kao alat ili podmodul drugog softverskog sistema koji pruža skup funkcija korisniku, ili četbot zahteva integraciju sa softverskim uslugama treće strane da bi izvršio ove zadatke. Često ove četbotove orijentisane na zadatak zovu i transakcionim četbotovima.

Dimenzija interakcije definiše komunikacioni mehanizam koji koristi četbot za obradu korisničkih informacija i za generisanje odgovora korisniku. U tu svrhu, četbotovi koriste tekstualne interfejsse (poruke), prepoznavanje glasa, slike ili kombinaciju svih navedenih. Tipično, četbotovi koji integrišu glasovnu interakciju sa njihovim korisnicima uvode neku vrstu prevođenja govora u tekst (engl. speech-to-text) i teksta u govor (engl. text-to-speech) ili automatsko prepoznavanje govora kao gornji sloj konverzionog procesa koji podržavaju govor i tekst, i integrišu tehnike obrade podataka prirodnog jezika. Ovo je slučaj poznatih komercijalnih glasovnih asistenata kao što su *Alexa* ili *Cortana*. Neki četbotovi uvode prepoznavanje slika kao funkciju za podršku i proširenje ograničenja komunikacije na prirodnom jeziku, podržan od strane drugih komercijalnih, popularnih četbotova kao što su *Siri* ili *XiaoIce*.

Klasifikacija bazirana na obradi ulaza (frazu, poruke, rečenice, pitanja) i generisanju odgovora se odnosi na mehanizam koji koristi četbot da procesira ulaze na prirodnom jeziku i da generiše adekvatnu poruku (odgovor) i/ili akciju (ako postoji) takođe na prirodnom jeziku korisnika. Prvenstveno se razlikuju dve glavne kategorije rešenja: deterministički modeli - zasnovani na pravilima (engl. Rule-based) i modeli zasnovani na veštačkoj inteligenciji (engl. AI-based). Deterministički algoritmi obrađuju ulazne poruke korisnika da bi izdvojili neku vrstu strukture, interpretiranog znanja, i primenili neku vrstu determinističke strategije za povezivanje ovih strukturiranih podataka sa specifičnom izlaznom porukom ili akcijom. Oni biraju sistemski odgovor na osnovu fiksnog unapred definisanog skupa pravila, zasnovanih na prepoznavanju leksičke

forme ulaznog teksta bez kreiranja novih tekstualnih odgovora. Znanje koje se koristi u četbotu je često ljudski ručno kodirano i organizovano i predstavljeno obrascima (paternima) konverzacije. Većina postojećih istraživanja o četbotovima zasnovanim na pravilima proučava izbor odgovora za razgovor u jednoj konverzaciji, koja uzima u obzir samo poslednju ulaznu poruku. U bolje humanoidnim četbotovima, izbor odgovora u više koraka uzima u obzir prethodne delove konverzacije da bi se izabrao odgovor relevantan za ceo kontekst razgovora. S druge strane, novije strategije koriste potencijal strategije zasnovane na veštačkoj inteligenciji, koje integrišu upotrebu mašinskog učenja i modela dubokog učenja za obradu korisničkih unosa i izgradnju izlaznih poruka na osnovu izvora znanja i podataka o obuci. Uglavnom, postoje dve vrste strategija zasnovanih na veštačkoj inteligenciji: modeli zasnovani na pronalaženju (engl. Retrieval-based models) i generisani modeli (engl. Generative models). Malo drugačiji od modela zasnovanog na pravilima je model zasnovan na pronalaženju, koji nudi veću fleksibilnost dok ispituje i analizira dostupne resurse koristeći API [55]. Modeli zasnovani na pronalaženju koriste modele i tehnike mašinskog učenja i dubokog učenja da bi razumeli ulaze i predvideli najtačniji odgovor iz zatvorenog skupa odgovora koristeći izlaznu rangiranu listu mogućih odgovora. Ovi modeli mogu pružiti pouzdanije i gramatički ispravnije odgovore. Lakše ih je učiti jer traže manje podataka, ali nisu u mogućnosti da odgovore na pitanja izvan svoje baze znanja i ne mogu kreirati novi tekst i učiti sami na osnovu predhodnih konverzacija, već biraju odgovor iz fiksnog skupa. Ova vrsta četbota zahteva ručnu pomoć od operatora za netipična pitanja. S druge strane, na generisanoj osnovi sistemi se fokusiraju na korišćenju modela dubokog učenja za sintezu i izgradnju odgovora na određeni korisnički unos, umesto da ga biraju iz zatvorenog skupa podataka odgovora. Oni generišu nove odgovore od starta jer mogu da odgovore na dvosmislena pitanja. Ovi modeli koji se baziraju na tehnikama mašinskog učenja i dubokog učenja mogu da vode duge i inteligentne razgovore. Ovakvi četbotovi kreiraju odgovore od početka koristeći generisanje prirodnog jezika (engl. Natural Language Generation - NLG). Ovi četbotovi postaju pametniji sa vremenom i sposobni su da samostalno uče iz prethodnih pitanja, odgovora, konverzacija i na taj način postaju „pametniji“. Međutim, teško ih je trenirati (učiti) jer zahtevaju veliku količinu podataka. Postoje i takozvani hibridni modeli gde se analiza fraza realizuje korišćenjem dela NLP, odnosno NLU, ali za potrebe izvršavanja i odgovaranja se koristi model zasnovan na pronalaženju gde se

pokušava naći odgovarajuća definisana akcija ili odgovor iz svoje baze znanja. Takođe, hibridni četbotovi mogu da koriste pristup zasnovan na pronalaženju i generisanim modelima kako bi odgovorili na unos korisnika ako se ne podudara ni sa jednim od pravila [56].

Glavni ključni faktor dizajna je obim baze znanja koju četbotovi podržavaju u svojim razgovorima, kao i nivo granularnosti. Ova karakteristika ima uticaj na izvore podataka (u smislu broja, raznolikosti i složenosti) potrebne za obuku i izgradnju baze znanja. Razlikuju se opšte (generički) i domen-zavisne baze znanja. Što se tiče prvog, opšte baze znanja nude informacije i teme razgovora iz bilo kog domena. Primer za ovu kategoriju je adaptivni četbot dizajniran da uči i prilagođava se novim kontekstima i temama na osnovu interakcije korisnika (npr. Google asistent). S druge strane, četbotovi zavisni od domena uključuju otvorene ili unakrsne domene (tj. integraciju više baza znanja i izvora podataka u domenu) i zatvorene domene (tj. fokusiranje na jednu, ekspertsku bazu znanja). Opšti i unakrsno domenski četbotovi obično zahtevaju pomoćne NLP tehnike da obrade i kontekstualizuju unos korisnika u određenu temu ili domen. To je posledica šireg obima razgovora, namera i entiteta korisnika. S druge strane, veća je verovatnoća da će četbotovi zatvorenog domena biti ograničeni na specifičan, dobro poznat podskup namera i entiteta kao što je npr. REA.

Odnos korisnika sa četbotom je definisana kao tip odnosa koji se uspostavlja između četbota i njegovog korisnika, na osnovu potreba koje četbot ispunjava. Razlikuju se interpersonalni, intrapersonalni i međuagent četbotovi. Interpersonalni četbotovi ne grade lični odnos sa korisnikom i usmereni su na pružanje konkretnih usluga zasnovanih na opštoj kategorizaciji korisnika. Oni mogu imati ličnost, mogu biti prijateljski nastrojeni i verovatno će zapamtiti informacije o korisniku, ali nisu obavezni niti se od njih očekuje da to urade. Primer ove kategorije je razgovor četbota za preporuke i rezervacije restorana ili avionskog leta. S druge strane, intrapersonalni četbotovi su fokusirani na lične scenarije gde četbot pomaže korisnicima da ispune lične zadatke u njihovom ličnom životu, pa zato postoji personifikacija usluge na osnovu korisničkih potreba. Oni su prijatelji korisnika i razumeju ga kao što čovek može da ga razume. Na primer, pametni kućni asistenti za starije osobe spadaju u ovu kategoriju. Dok su interpersonalni četbotovi tipično nezavisni od konteksta i nezavisni od korisnika,

intrapersonalni četbotovi su dizajnirani sa tehnikama profilisanja korisnika i mehanizmima konfiguracije korisnika kako bi se uklopili i prilagodili potrebama korisnika i ličnim ciljevima. S obzirom da se na tržištu pojavljuje sve veći broj četbotova, javila se i potreba međusobne komunikacije između njih. Upravo međuagent četbotovi omogućavaju takvu komunikaciju. Iako četbot ne može biti potpuno međuagentski, svakako može biti servis koji upravlja ili komunicira sa drugim četbotom ili upravlja komunikacijom, što olakšava programerima i korisnicima da integrišu različite usluge u konverzacioni ekosistem. Integracija *Alexa-Cortana* su dva četbota koji su primer međuagenta [57].

Ljudska pomoć oslikava stepen autonomije u kojoj se četbotom može upravljati, bilo da je dizajniran kao autonomni četbot ili je potrebna ljudska intervencija. Ljudsko posredovanje odnosi se na četbotove koji zahtevaju ljudsku interakciju u nekom trenutku konverzacionog procesa. Neki konteksti scenarija zahtevaju od ljudi da se prilagode i prošire četbot bazu znanja u vreme izvršavanja ili bar da preuzmu komunikaciju sa korisnikom kad prepoznaju da četbot nije znao odgovor ili kad je pogrešio (npr. kontakt centar). Sa druge strane, autonomni četbotovi potpuno autonomno komuniciraju sa korisnicima bez ljudske pomoći u petlji.

Postoji podela na dobre i loše četbotove. Dok su ovi prvi nadamo se jasni, ovi drugi imaju loše namere jer žele da iskoriste porast onlajn saobraćaja zbog globalne pandemije. Oni su sada sofisticiraniji nego ikad i simuliraju ljudsko ponašanje na načine koji ih teže otkriti i sprečiti. Oni omogućavaju bot operaterima, napadačima, neugodnim konkurentima i prevarantima da izvrše širok spektar zlonamernih aktivnosti. Skraping veba, konkurentno rudarenje podataka, prikupljanje ličnih i finansijskih podataka, logovanje grubom silom, prevare sa digitalnim oglasima, neželjena pošta i transakcione prevare su samo neke od ovih aktivnosti. Ovo samo ukazuje na neke negativne efekte koje četbot može da donese.

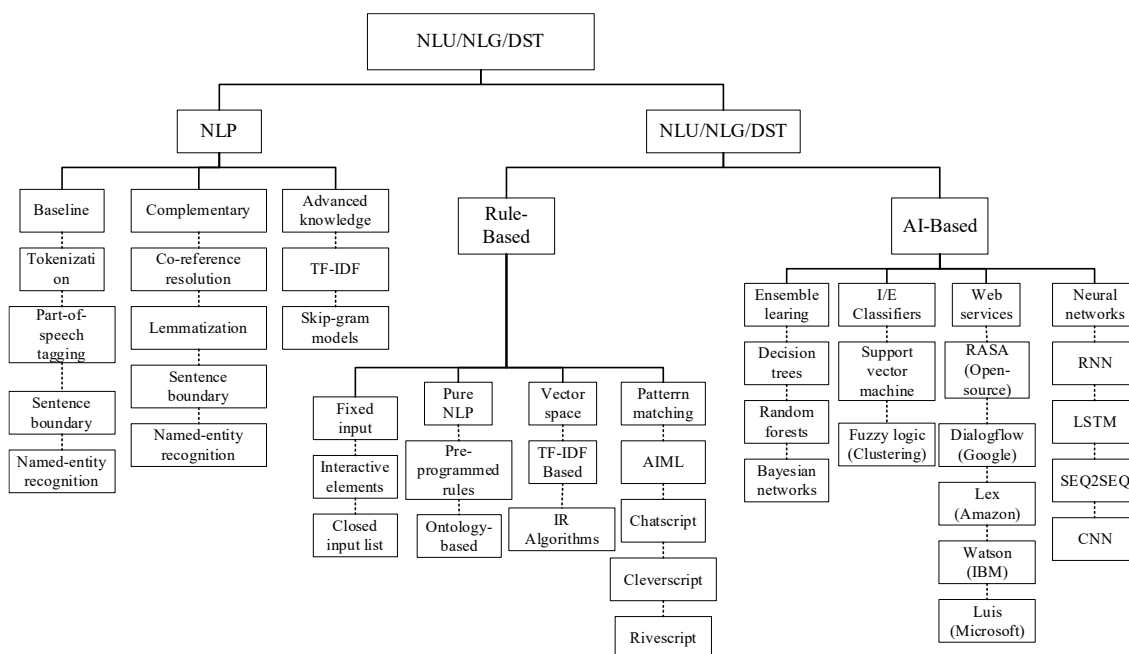
Četbotovi se takođe mogu klasifikovati prema dozvolama koje im daje njihova razvojna platforma. Razvojne platforme mogu biti otvorenog koda, kao što je *RASA Open source*, ili mogu biti vlasničkog koda kao što su razvojne platforme koje obično nude velike kompanije kao što su *Google* ili *IBM*. Platforme otvorenog koda pružaju dizajneru četbota sposobnost da se interveniše u većini aspekata implementacije. Zatvorene

platforme obično deluju kao crne kutije, što može biti značajan nedostatak u zavisnosti od projekta.

Naravno, četbotovi ne pripadaju isključivo jednoj ili drugoj kategoriji, ali ove kategorije postoje u svakom četbotu u različitim razmerama.

2.5. TEHNIČKE SPECIFIKACIJE IMPLEMENTACIJE

Analiza metoda, tehnika i tehnologija koje se koriste za realizaciju četbota je jedan od najčešćih predmeta istraživanja. Fokus tehničkog istraživanja je na obradi prirodnog jezika i znanju interpretacije zadataka u pogledu pametne komunikacije sa korisnikom. Identifikovane su četiri glavne teme u implementaciji četbota [6]: NLP, NLU, NLG i praćenje faze dijaloga (engl. dialogue stage tracking - DST). Slika 6. sumira metode, tehnike i tehnologije o kojima se piše u istraživačkoj literaturi [6].



Slika 6. Metode, tehnike i tehnologije razvoja četbota

Što se tiče NLP tehnika, s obzirom da je prirodni jezik osnova konverzacije većine interfejsa četbota, oni su obično uključeni kao gornji sloj između korisničkog interfejsa

za unos (npr. interfejs za ćaskanje, prepoznavanje glasa) i modula za interpretaciju prirodnog jezika odgovornog za tumaćenje ovih poruka. Postoje različite NLP tehnike i one se ovde klasifikuju u tri kategorije: osnovne tehnike, komplementarne tehnike i napredno tumaćenje znanja.

- Osnovne tehnike se obično koriste kao preliminarni zadaci za bilo koji NLP plan izvršavanja (engl. Pipeline). Ove tehnike se se koriste za strukturiranje, ćišćenje i dodeljivanje osnovnih gramatićkih napomena korisnikovim ulaznim frazama (porukama). Oni su generalno integrisani sa nezavisnošću integracije NLU modula. Osnovne tehnike uključuju tokenizaciju, oznaćavanje dela govora (engl. part-of-speech tagging), razgranićenje granice rećenice i prepoznavanje imenovanih entiteta.
- Komplementarne tehnike predstavljaju proširenje osnovnih tehnika, i obično se primenjuju pre naprednih NLU tehnika da bi se dobilo prošireno gramatićko i semantićko znanje. Komplementarne tehnike uključuju rezoluciju koreferenci, lematizaciju, rašćlanjivanje zavisnosti i oznaćavanje semantićkih uloga.
- Napredna interpretacija znanja kao što su strukture modela vektorskog prostora predstavljaju most između ćistih NLP tehnika i integracije NLU modula. Ova napredna interpretacija omogućava transformaciju podataka korpusa teksta u osnovu znanja za komunikacioni proces ćetbota. Ove tehnike uključuju vektorske reprezentacije kao što su TF-IDF i modeli sa skip-gramom kao što je Word2Vec.

Što se tiće NLU, NLG i DST tehnika, postoji opšta kategorizacija dva glavna pristupa: pristupi zasnovani na pravilima i pristupi zasnovani na veštaćkoj inteligenciji. Identifikovane su ćetiri kategorije tehnika bazirane na osnovu pravila: **fićsni unos** (engl. Fixed input), **ćisti NLP** (engl. Pure NLP), **vektorski prostor** (engl. Vector-space) i **podudaranje šablona** (engl. Pattern-matching). Pristup zasnovan na veštaćkoj inteligenciji takode ima podelu na ćetiri kategorije tehnika: **ansambl ućenje** (engl. ensemble learning) kao npr. stablo odlućivanja, **I/E klasifikatori** (engl. I/E classifiers) kao npr. Intent/Entity klasifikatori, **Veb servisi** (rešenja koja obićno dozvoljavaju kombinaciju mašinskog ućenja (I/E klasifikatora) i pristupa zasnovanih na pravilima za NLU zadatke. Primer veb servisa su: *RASA*, *DialogFlow*, *Luis* itd.) i **neuronske mreže** (engl. Neural networks).

Na kraju ove klasifikacije nikako ne treba zaboraviti da se četbotovi naslanjaju na specifične konverzacione kontekste. Ovi konteksti konverzacione se mogu podeliti na otvorene i zatvorene domene. U otvorenom domenu kontekst konverzacione nije definisan. Ne postoji dobro definisan cilj ili namera. Primer ovoga može biti razgovor sa *Google* asistentom. Možemo ga pitati šta poželimo, a on će pokušati da pronađe odgovor i da ga prosledi. Većina aplikacija za četbot spadaju u zatvorene domene. Svaki četbot ima poseban kontekst (kontekstne informacije) i specijalizovan je za odgovore na pitanja vezana za taj domen. Tehnička podrška korisnicima ili asistenti za kupovinu su primeri problema zatvorenog domena.

2.6. PLATFORMA – ARHITEKTURA ZA RAZVOJ ČETBOTA

Savremena platforma za razvoj četbota mora da ima sledeća tri dela koja zaista pruža konverzaciono iskustvo:

- Obrada prirodnog jezika (NLP): Razumevanje unosa korisnika i izvlačenje relevantnih informacija.
- Tok razgovora: Uključuje upravljanje kontekstom razgovora.
- Izvršavanje akcija (zadataka): koriste se za pružanje jednostavnih odgovora, kao i naprednih funkcija kao što su upiti u bazu podataka, zahtev ka interfejsu aplikativnog programiranja (API) ili logički prilagođeni okidač.

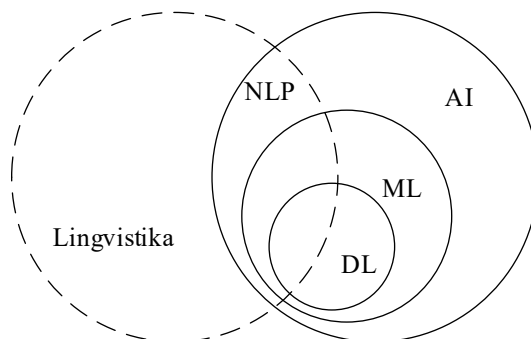
NLU je u ovom radu naglašen kao srce svakog četbota. Proširivost je sposobnost softverskih arhitektura da poveže eksterne arhitekture (listove) na njihovu osnovnu strukturu, stvarajući sinergiju između ovih različitih arhitektura [58]. U ovom radu predložena je napredna proširiva arhitektura za mikroservisne četbotove koja omogućava lakše prebacivanje sa jednog NLU provajdera na drugi (npr. zbog bolje podrške za srpski jezik ili zbog smanjenja troškova). Ovde predloženi objektno orijentisani okvir je dizajniran da bude agiln, da se izbori i sa neverovatno promenljivim okruženjem savremenih komunikacionih kanala za interakciju korisnika. Rešenje je primenjeno kroz dve studije slučaja na srpskom jeziku. I do sada, koliko je autoru poznato, niko se nije

bavio srpskim jezikom koristeći različite NLU. Zbog svega gore navedenog, nezavisnost od proizvođača rešenja servisa za razumevanja prirodnog jezika je neophodan uslov za proširljivu i skalabilnu arhitekturu koja će kasnije detaljnije biti opisana.

3. OBRADA PRIRODNOG JEZIKA - NLP

Počeci moderne veštačke inteligencije mogu se pratiti od pokušaja klasičnih filozofa da opišu ljudsko mišljenje kao simbolički sistem. Međutim, polje veštačke inteligencije nije formalno osnovano sve do 1956. godine, na konferenciji na koledžu Dartmouth u Nju Hempširu, gde je prihvaćen izraz veštačka inteligencija (Artificial Intelligence - AI).

Oblast studija koja se fokusira na interakcije između ljudskog jezika i računara naziva se obrada prirodnog jezika ili skraćeno NLP. NLP predstavlja multidisciplinarnu oblast obuhvatajući informatiku, veštačku inteligenciju i računarsku lingvistiku. Veštačka inteligencija omogućava računarima da imitiraju ljudsko ponašanje. Mašinsko učenje (engl. machine learning - ML) je podskup veštačke inteligencije koje koristi statističke metode za unapređenje učenja računara odnosno davanja preciznijih predviđanja. ML je u stvari tehnika za realizaciju AI. Duboko učenje (engl. Deep Learning – DL) je sledeća generacija ML i predstavlja podskup mašinskog učenja. Kao što ljudski mozak koristi paterne i klasifikuje različite vrste informacija za određene probleme, isto tako DL koristi algoritme za izvršavanje tih problema na računarima. Inspiracija je nastala funkcijom ljudskog mozga. Reči AI, NLP i ML se ponekad koriste gotovo naizmenično. Hijerarhijski, NLP se smatra podskupom mašinskog učenja, dok NLP i ML spadaju u veću kategoriju veštačke inteligencije. NLP kombinuje AI i računarsku lingvistiku tako da računari i ljudi mogu “lako” da razgovaraju.

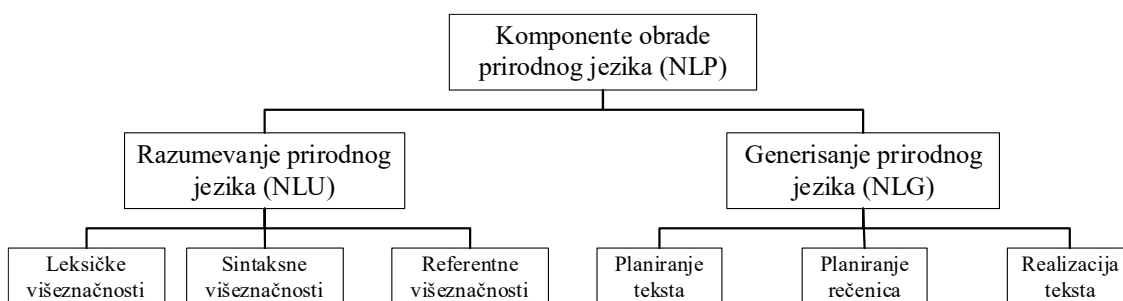


Slika 7. Odnos AI, ML, DL i NLP

NLP nastoji da premosti jaz između mašina i ljudi tako što omogućava računaru da analizira ono što je korisnik rekao (unos proznatog govora) i obradi ono što je korisnik mislio. Ovaj zadatak se pokazao prilično složenim. Da bi razgovarao sa ljudima, program mora razumeti sintaksu (gramatiku), semantiku (značenje reči), morfologiju (vreme) i pragmatiku (konverzaciju). Broj pravila koja treba pratiti može izgledati ogromno i to objašnjava zašto su raniji pokušaji NLP -a u početku doveli do razočaravajućih rezultata. Sa drugačijim sistemom, NLP je polako poboljšavao prelazak sa glomaznih pravila na učenje zasnovano na uzorcima. Tokom proteklih nekoliko godina, arhitektura i algoritmi dubokog učenja postigli su impresivan napredak u oblastima kao što su prepoznavanje slika i obrada govora. Prepoznavanje imenovanih entiteta, označavanje dela govora ili analiza osećanja neki su od problema u kojima su modeli neuronskih mreža nadmašili tradicionalne pristupe. Napredak u mašinskom prevodenju je možda najznačajniji među svima. Osim velikog doprinosa u razvoju čebota, algoritmi za obradu prirodnog jezika se koriste u pretraživanju, onlajn prevodenju, filterima za neželjenu poštu i proveru pravopisa. Dakle, koristeći NLP, programeri mogu organizovati i strukturirati masu nestrukturiranih podataka za obavljanje određenih zadataka. NLP se može podeliti na dve osnovne komponente:

- Razumevanje prirodnog jezika (engl. Natural Language Understanding - NLU)
- Generisanje prirodnog jezika (engl. Natural Language Generation - NLG)

Ove dve komponente se dalje dele na podkomponente. Slika 8. prikazuje podelu ove dve osnovne komponente.



Slika 8. Komponente obrade prirodnog jezika (NLP)

Mnogo je nejasnoća tokom učenja ili pokušaja tumačenja jezika i to je razlog zašto NLU ima teže zadatke od NLG.

- Leksička višeznačnost (engl. Lexical Ambiguity) može nastati kada reč ima različit smisao, odnosno ima više od jednog značenja, a rečenica u kojoj je sadržana može se različito tumačiti u zavisnosti od njenog ispravnog značenja. Leksička nejasnoća može se donekle rešiti upotrebom tehnika označavanja delova govora (engl. parts-of-speech tagging).
- Sintaksna višeznačnost (engl. Syntactical Ambiguity) se dešava kada vidimo više od jednog značenja u nizu reči. Takođe se naziva gramatička nejasnoća.
- Referentna višeznačnost (engl. Referential Ambiguity) podrazumeva kad se tekst pominje kao entitet (nešto/neko), a zatim se na njega ponovo poziva, moguće u drugoj rečenici, koristeći drugu reč.

Generisanje prirodnog jezika (NLG) predstavlja proces generisanja smislenih fraza i rečenica u obliku prirodnog jezika iz nekog unutrašnjeg prikaza. To uključuje planiranje teksta i rečenice i realizaciju teksta. Planiranje teksta uključuje preuzimanje relevantnog sadržaja iz baze znanja. Planiranje rečenica uključuje izbor potrebnih reči, formiranje smislenih fraza, setovanje mnogo rečenica. Realizacija teksta je preslikavanje plana rečenice u strukturu rečenice.

Redosled koraka izvršavanja NLP zadataka zavisi od svakog NLP proizvođača. Treba imati na umu da redosled opisuje neke standardne tokove rada, međutim on se može drastično razlikovati u toku implementacije u stvarnom životu na osnovu analize problema ili zahteva. Na tržištu postoji mnogo NLP biblioteka, paketa i alata. Svaki od njih ima svoje prednosti i nedostatke. Kao tržišni trend, Python je jezik koji ima najviše kompatibilnih biblioteka. Većina njih pruža osnovne karakteristike NLP-a. Svaka NLP biblioteka je izgrađena sa određenim ciljevima, pa je sasvim očigledno da jedna biblioteka možda neće pružiti rešenja za sve. Međutim, programer je taj koji treba na osnovu znanja i iskustva da odluči šta, gde i kad da ih koristi. Neke od najpoznatijih NLP alata su: SpaCy, NLTK, Gensim.

Znanje o razumevanju i upotrebi ljudskog jezika se prikuplja da bi se razvile tehnike koje će omogućiti računarima da razumeju i manipulišu prirodnim izrazima kako bi izvršili željene zadatke [59]. Većina NLP tehnika je zasnovana na mašinskom učenju.

NLU je u osnovi svakog NLP zadatka i on je neophodan da obezbedi četbotu da razume i deluje na osnovu korisničkog unosa. NLU pomaže računarima da razumeju ljudski jezik razumevanjem, analizom i tumačenjem osnovnih delova poruke ili govora. On to rešava razumevanjem konteksta, semantike, sintakse, namere i sentimenta teksta. U tu svrhu se koriste različita pravila, tehnike i modeli. Njegov osnovni zadatak je da pronade nameru iz tog teksta. NLU je treniran (obučen) prirodnim korisničkim frazama (iskazima) označenim entitetima i proširenom upotrebom sinonima. NLU koristi tehnike mašinskog učenja i obrade prirodnog jezika (NLP) za izdvajanje strukturiranih informacija (namera i povezani entiteti) iz nestruktuiranog korisničkog unosa (tekstualne poruke). NLU ima za cilj da izdvoji kontekst (kontekstne informacije) i značenja iz korisničkih unosa na prirodnom jeziku, koji mogu biti nestrukturirani i kao takvi prihvatljivo odgovore u skladu sa namerama korisnika [59]. On identifikuje nameru korisnika i izdvaja entitete specifične za domen u različitim kontekstima.

Namera predstavlja mapiranje između onoga što korisnik kaže i akcije koju treba četbot da uradi. Akcije predstavljaju korake koje će četbot preduzeti kada se određene namere okinu korisnikovim unosima i one mogu imati parametre za određivanje detaljnih informacija o tome. Otkrivanje namere se obično formuliše kao klasifikacija rečenica u kojoj je jedna ili više labela namere predviđena za svaku rečenicu. Entitet je alat za izdvajanje vrednosti parametra iz unosa prirodnog jezika. Na primer, rečenica „Da li je hladno u Beogradu?“. Namera korisnika je da sazna „vremensku prognozu“. Vrednost entiteta je mesto „Beograd“. Korisnik traži vremensku prognozu u Beogradu. Entiteti mogu biti sistemski ili programerski definisani. Na primer sistemski entitet „@sis.date“ odgovara standardnim referencama datuma kao što je 11. avgust 2021. ili 11. avgust. Ekstrakcija domenskih entiteta koja se obično naziva problemom popunjavanja mesta (slot), formulisana je kao problem sekvencijalnog označavanja gde se delovi rečenice izdvajaju i označavaju entitetima domena. Neki proizvođači NLU platformi koriste koncept slot-a da bi rešili kontekst fraze. Kontekst je veoma bitan termin u NLP svetu. Konteksti su stringovi koji čuvaju kontekst objekta na koji se korisnik poziva ili o kome govori. Na primer, korisnik se može pozvati na prethodno definisan objekat u sledećoj rečenici: „Pada li kiša?“. Ako je u prošlom primeru bilo reč o Beogradu, jasno je da je kontekst koji treba sačuvati Beograd. Tako da kada korisnik kao sledeći unos kaže „Pada li kiša“, namera „Pada li kiša“ se može se pozvati u kontekstu mesta „Beograda“.

Koncept korišćenja veštačke inteligencije i ljudskog jezika za obavljenje komunikacije sa mašinama nisu novina. Međutim, istraživači su godinama, čak i decenijama, potcenjivali složenost ljudskog jezika. Čak i danas, nakon velikog napretka u AI i NLP-u, sprovedeno je nekoliko studija koje pokazuju da se korisnici ne mogu tako lako prevariti i da skoro uvek znaju da komuniciraju sa računaraom, a ne sa drugim ljudskim bićem. Zato je za razvoj četbota izbor NLU platforme veoma bitan.

Na koje stvari treba obratiti pažnju prilikom odabira NLU platforme?

1. **Jezička podrška** (engl. Language support): NLU platforma treba da podržava jezik ulaznih podataka. Trenutno je kvalitet NLU-a niži na svim jezicima osim engleskog, zbog manjeg komercijalnog potencijala jezika. Međutim, sa povećanim istraživačkim interesovanjem, ovo se menja.
2. **Kvalitet rezultata** (engl. Result quality): Dobro NLU rešenje treba da bude u stanju da prepozna jezičke entitete, izdvoji veze između njih i koristi sematički softver za razumevanje sadržaja, bez obzira na to kako je izražen. Kontinuirano učenje zasnovano na mašinskom učenju tokom vremena može poboljšati kvalitet rezultata.
3. **Brzina** (engl. Speed): Razumevanje jezika je deo procesa u konverzionim AI aplikacijama. Ostali delovi uključuju generisanje odgovora ili delovanje na osnovu upita. Zbog toga proces percepcije i tumačenja jezika treba da se odvija brzo. Međutim, može da postoji kompromis između kvaliteta rezultata i brzine izračunavanja. Izbor treba uraditi u zavisnosti od oblasti primene.
4. **Fleksibilnost** (engl. Flexibility): Prilagodljivost ka različitim oblastima rešenja je važna. Ovo se postiže treningom i mogućnostima kontinuiranog učenja NLU rešenja.
5. **Upotrebljivost** (engl. Usability): Rešenje bi trebalo da bude lako za korišćenje tehnički obučenih i neobučenih zaposlenih. Može se razmotriti rešenje sa različitim interfejsima za netehničke zaposlene (na primer, zaposleni u korisničkoj službi) koji aktivno razvijaju ovaj sistem sa povratnim informacijama. S obzirom da četbotovi mogu da razvijaju zaposleni koji nisu tehnički obučeni, upotrebljivost softvera i lakoća korisničkog interfejsa je veoma važna.

Kao i u mnogim novim oblastima, tehnološki giganti zauzimaju veliko mesto u razvoju NLU. Slede neka od njihovih rešenja:

1. Microsoft

- Language Understanding Intelligente service - LUIS;
- Knowledge Exploration Service;
- Azure Translator APIs.

2. Google

- Dialogflow;
- Translate API;
- Cloud Natural Language API.

3. IBM

- Watson Conversation Service;
- Watson Tone Analyzer.

4. Amazon

- Lex;
- Comprehend.

Neke firme koje nude otvoreni kod koristeći API su takođe deo ovog ekosistema. Među najpoznatijim i najkorišćenijim su:

- Facebook - Wit.ai;
- Rasa NLU;
- FuzzyWuzzy;
- PyNLP;
- Stanford CoreNLP;
- NLTK.

3.1. OSNOVNE KOMPONENTE NLU

S obzirom da je *RASA Open source* otvorenog koda cilj ovog dela teksta je da objasni kako generalno radi NLU posmatrajući neke od osnovnih komponenti u RASA

NLU planu izvršavanja, kao i da objasni kako one međusobno deluju. Po sličnom principu rade i ostale poznate NLU platforme. U RASA projektu, NLU plan izvršavanja predstavlja niz komponenti koji definiše korake obrade koji pretvaraju nestrukturirane korisničke poruke u namere i entitete. Ove komponente programeri mogu konfigurisati i prilagoditi. NLU plan izvršavanja se obično definiše u nekom fajlu kao što se u RASA definiše u „config.yml“ fajlu. Ova datoteka opisuje sve korake u planu izvršavanja koje će Rasa koristiti za otkrivanje namera i entiteta. Počinje sa tekstom kao ulazom i nastavlja da analizira sve dok ne dobije entitete i namere kao izlaz, slika 9.



Slika 9. Koraci u NLU planu izvršavanja

Postoje različite vrste komponenti koje možemo očekivati u NLU planu izvršavanja. Glavne su:

1. Tokenizer (engl. Tokenizers),
2. Karakterizatori (engl. Featurizers),
3. Klasifikatori namere (engl. Intent Classifiers),
4. Ekstratori entiteta (engl. Entity Extractors).

1. Tokenizer (engl. Tokenizers)

Prvi korak je da se fraza (izraz) podeli na manje delove teksta, poznate kao tokeni. Ovo se mora uraditi pre nego što se tekst karakterizuje (engl. featurized) za mašinsko učenje kao na sledećoj slici, što je razlog zašto je tokenizer naveden kao prva komponenta.

"Hi, my name is Vincent." → ["Hi", "my", "name", "is", "Vincent"]

Details on Tokenizers.

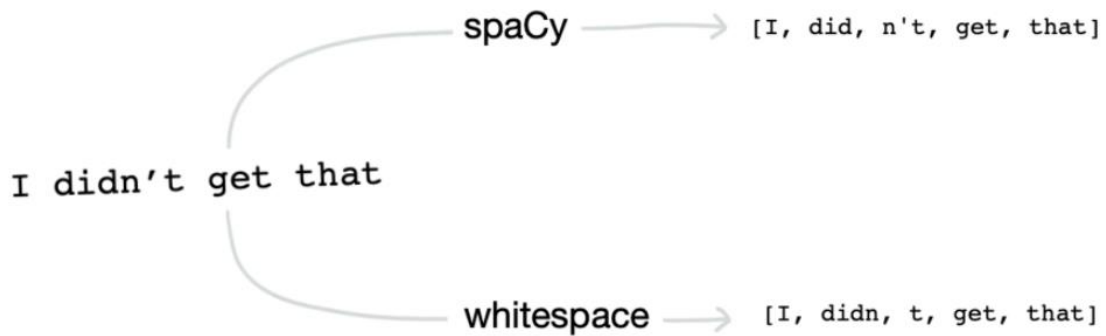
Just Tokenisation
"He likes dogs" → ["He", "likes", "dogs"]

Tokenisation and Lemmatisation
"He likes dogs" → ["He", "like", "dog"]

Slika 10. Fraza podeljena na manje delove teksta - tokene

Neki tokenizeri takođe dodaju posebne informacije tokenima. Na primer, *spaCy* (*referenca*) takođe može da generiše osnovu reči (engl. lemma) tokena koje kasnije može da koristi *CountVectorizer*. *CountVectorizer* je odličan alat koji obezbeđuje biblioteka u *Python*-u. Koristi se za transformaciju datog teksta u vektor na osnovu učestalosti (broja) svake reči koja se pojavljuje u celom tekstu.

Tokenizer deli svaku pojedinačnu reč u frazi u poseban token, a obično je izlaz tokenizera lista reči. Takođe možemo dobiti odvojene tokene za interpunkciju u zavisnosti od tokenizera i podešavanja kroz koja prolazimo. Za engleski jezik se obično koristi *WhiteSpaceTokenizer*, ali za ostale jezike može biti uobičajno da se koriste neki drugi. *WhiteSpaceTokenizer* unutar RASE deli fraze na razmake, ali i na nealfanumeričke znakove. *SpaCy* tokenizer koji dolazi uz RASU dolazi sa skupom unapred definisanih pravila za obradu razdvajanja karaktera. Na engleskom to može izazvati suptilne razlike u tokenima, ali u drugim jezicima ove razlike mogu svakako biti dublje.



Slika 11. Razlika razdvajanja karaktera u zavisnosti od teknizera

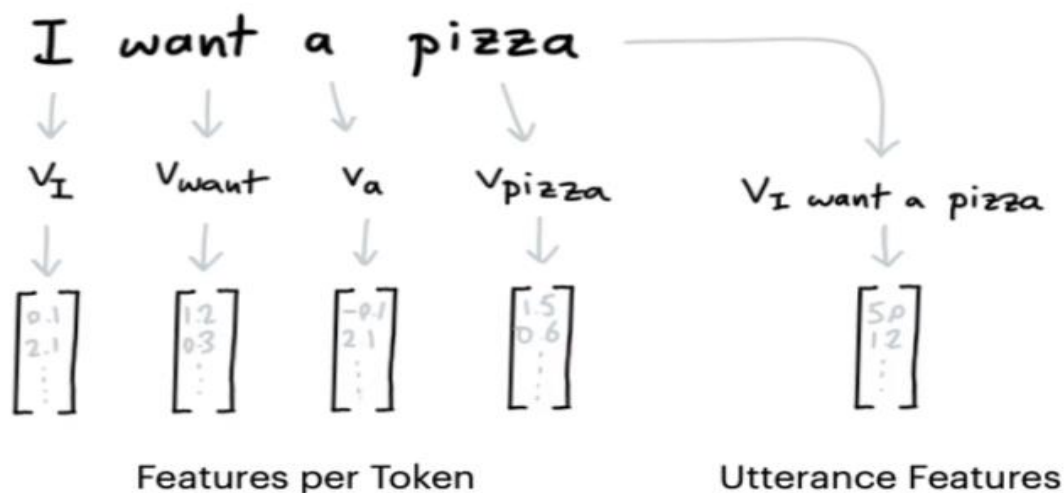
SpaCy modeli koje RASA direktno podržava u vreme pisanja uključuju Kineski, Danski, Holandski, Engleski, Francuski, Nemački, Grčki, Italijanski, Japanski, Litvanski, Makedonski, Norveški, Poljski, Portugalski, Rumunski, Portugalski, Rumunski, Ruski i Španski.

SpaCy je dobar izbor za Evropske jezike koji nisu engleski i on podržava više tokenizera. Treba napomenuti da tokenizeri ne menjaju osnovni tekst, oni samo razdvajaju tekst na tokene. To znači na primer, da velika slova ostaju netaknuta. Ako želimo da šifriramo samo mala slova za svoj NLU plan izvršavanja potreban je karakterizator (engl. featurizers).

2. Karakterizator (engl. Featurizers)

Mašinsko učenje se odnosi na statistiku, statistika radi sa brojevima, a ne rečima. Karakterizator transformiše tokene kao i neka njihova svojstva u karakteristike koje kasnije mogu da koriste algoritmi mašinskog učenja (ML modeli). Jednom kada se fraza tokenizira, RASA nastavlja dodavanjem numeričkih karakteristika. Ove karakteristike će kasnije koristiti algoritmima mašinskog učenja (ML modeli) da bi izabrali ispravnu nameru i entitet. Karakteristike se dodeljuju svakom tokenu, ali postoje i karakteristike koje pripadaju celoj frazi (iskazu, rečenici). Različiti algoritmi mašinskog učenja očekuju različite unose karakteristika tako da je potrebno upariti karakterizator sa odgovarajućim algoritmom za mašinsko učenje. Dakle, karakterizatori konvertuju ulazne tokene u mašinski čitljive (numeričke) oblike. NLU plan izvršavanja može da koristi više karakterizatora koji se mogu izvršavati paralelno ili sekvencijalno. Korišćenje već treniranih karakterizatora štedi na vremenu i računanju, ali može da se desi da su takvi

karakterizatori domenski specifični i nisu od velike pomoći. Zato je veoma bitno biti obazriv prilikom izbora karakterizatora. Na slici 12. je prikazan dijagram koji pokazuje kako se može šifrovati rečenica „I want a pizza“.



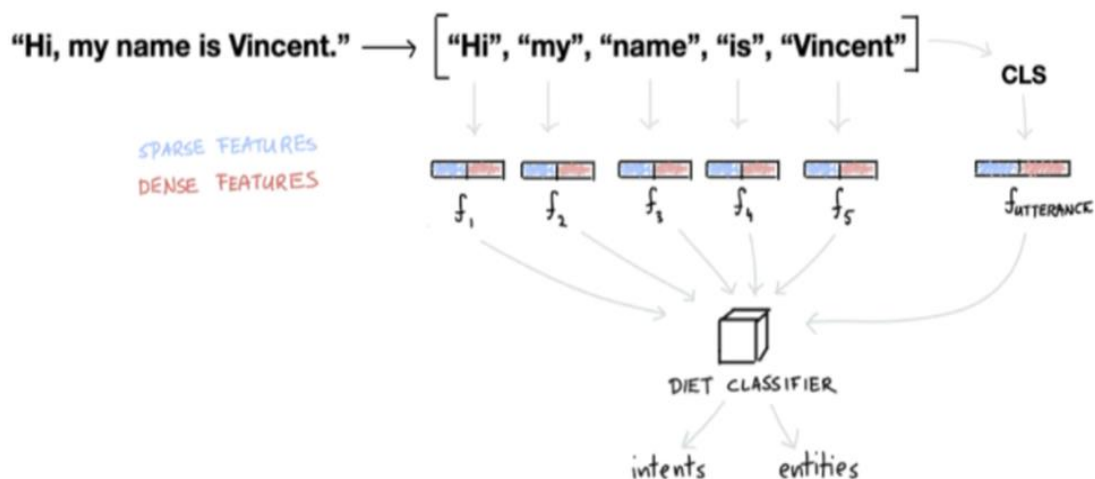
Slika 12. Dijagram koji prikazuje kodiranje rečenice

Postoje dve vrste karakteristika: retke (engl. sparse) i guste matrice (engl. dense). U matematici, retke i guste matrice se često odnose na broj nultih naspram nenultih elemenata u nizu (npr. vektor ili matrica). Ove prve matrice su one koje sadrže veliki broj nula. Ove druge, tzv. guste matrice su one koje sadrže malo nula ili uopšte ne sadrže nule. S obzirom da retke matrice zauzimaju manje memorije, one se pogodnije zapisuju kao niz vrednosti uz pamćenje njima odgovarajućih indeksa.

Treba imati na umu da postoji sloboda da se dodaju sopstvene komponente pomoću prilagođenih alata za karakterisanje. Na primer, postoji projekat koji održava zajednica pod nazivom *Rasa-nlu-examples* koji ima mnogo eksperimentalnih dodataka za neengleske jezike. RASA ga zvanično ne podržava, ali može biti od pomoći mnogim korisnicima jer je zastupljeno preko 275 jezika.

3. Klasifikatori namere (engl. Intent Classifiers)

Kada se generišu karakteristike za sve tokene i za celu frazu, one se prenose ka model klasifikacije namera. Preporuka je da se koristi RASIN DIET (Dual Intent and Entity Transformer - DIET) model koji može da obradi klasifikaciju namera kao i ekstrakciju entiteta što je prikazano na slici 13. Model je takođe u stanju da uči iz karakteristika tokena i fraze.

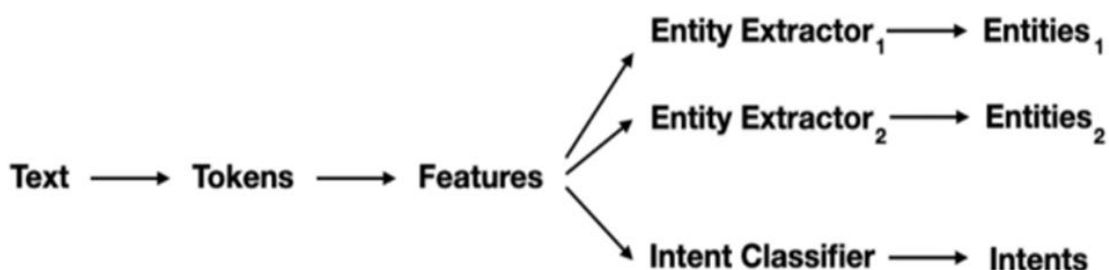


Slika 13. Korišćenje DIET klasifikatora

Većina algoritama koje je RASA hostovala u prošlosti radili su otkrivanje entiteta ili klasifikaciju namera, ali nisu radili oboje. To je takođe značilo da modeli klasifikacije namera samo gledaju na karakteristike rečenica u NLU planu izvršavanja i ignorišu karakteristike tokena. Zato je DIET komponenta veoma korisna.

4. Ekstrakcija entiteta (engl. Entity Extraction)

Iako je DIET u mogućnosti da nauči kako da otkrije entitete, ne preporučuje se obavezno korišćenje za svaku vrstu entiteta. Na primer, entitetima koji prate strukturirani obrazac (patern), kao što su brojevi telefona, nije potreban algoritam da bi ih otkrili. Umesto toga, može se koristiti alat za ekstrakciju entiteta - Regex (engl. Regex Entity Extractor). Zbog toga je uobičajno imati više od jednog tipa ekstratora entiteta u NLU planu izvršavanja. Ovo je prikazano na slici 14.



Slika 14. Ekstrakcija entiteta u NLU planu izvršavanja

Nakon pregleda različitih tipova komponenti u NLU planu izvršavanja, daje se objašnjenje kako ove komponente dele informacije jedna sa drugom.

3.1.1. Interakcija - prosleđivanje poruke

NLU plan izvršavanja predstavlja niz komponenti. Ove komponente se obučavaju i obrađuju onim redosledom kojim su navedene u fajlu „config.yml“ sa slike 15.



Slika 15. Fajl „config.yml“ (redosled komponente u fajlu)

To znači da se konfiguracija izvršavanja komponenti u NLU planu izvršavanja može posmatrati kao linearni niz koraka kroz koji podaci treba da prođu. Kada korisnik razgovara sa četbotom RASA interno čuva trag svakog stanja fraze preko objekta „Message“. Ovaj objekat se obrađuje svakim korakom u NLU planu izvršavanja. Slika 16. dijagram daje pregled onoga što se dešava kada se poruka obrađuje.



Slika 16. Dijagram obrade poruke

U ovom dijagramu treba naglasiti sledeće:

1. Poruka prvo počinje kao kontejner sa samo čistom korisničkom frazom (rečenicom).
2. Nakon što poruka prođe kroz tokenizator, ona se deli na tokene. Na dijagramu su tokeni predstavljeni kao stringovi, dok su interno predstavljeni objektom „Token“.
3. Kada poruka prođe kroz *CountVectorsFeaturizer*, primećuje se da su dodate *sparse* karakteristike. Postoji razlika između karakteristika sekvence i cele fraze (rečenice). Nakon prolaska kroz drugi karakterizator broj *sparse* karakteristika se povećava.
4. *DIETClassifier* traži „*sparse_features*” i „*dense_features*” u poruci da bi napravio predviđanje. Nakon toga, objektu poruke se dodaje predviđana namera.

Svaki put kada poruka prođe kroz korak NLU plana izvršavanja, objekat poruke dobija nove informacije. To znači da se može nastaviti sa dodavanjem koraka u NLU planu

izvršavanja, ako ima potrebe za dodavanje informacije poruci. To je takođe razlog zašto se mogu ubaciti dodatni modeli ekstrakcije entiteta. Kao što se vidi svaki korak u procesu može dodati informacije u poruku. To znači da se može dodati više koraka za izdvajanja entiteta.



Slika 17. Dodavanje informacija u poruku

U ovom delu teze prethodno su kratko opisana međusobna delovanja komponente unutar RASA NLU plana. Veoma je bitno dobro razumeti kako komponente deluju u NLU planu jer će to pomoći u odlučivanju izbora relevantnih komponenti za potrebe našeg četbota. Takođe je važno shvatiti da se može potpuno prilagoditi NLU plan izvršavanja. Mogu se ukloniti komponente ako nisu potrebne, a mogu se i dodati svoje nove komponente ako je potrebno koristiti sopstvene alate. Ovo može biti posebno bitno ako se radi sa jezikom koji nije engleski i ako se žele koristite prilagođeni jezički alati sa kojima smo upoznati.

3.2. POREĐENJE DVA POZNATA NLU SERVISA

Zbog ograničenja rada i zbog potrebe dokazivanja postavljenih hipoteza u vezi nezavisnosti arhitekture od NLU servisa ovde je ukratko data analiza i poređenje dva veoma popularna NLU servisa: Dialogflow i RASA. Naime, tokom razvoja prve verzije četbota ADA, o kojoj će biti više reči u poglavlju evaluacija četbota, najpre je korišćen opšte poznat *Google-ov* NLU servis tzv. *Dialogflow*. Vrlo brzo su uviđene prednosti i nedostaci istog. Upravo zbog nekih nedostataka testirana je RASA kao jedan od najpouzdanijih NLU servisa otvorenog koda u smislu ocene poverenja [59]. Rezultat poređenja ova dva servisa je dovela do shvatanja potrebe korišćenja oba, ali i do razumevanja problema zavisnosti od NLU proizvođača. Ovo je takođe dovelo do ideje koje je implementirana u predloženoj arhitekturi. Dialogflow i RASA su dva različita NLU servisa sa istom svrhom. Tabela 1. ukratko opisuje njihove razlike i razloge zašto su izabrani kao testni servisi za predloženu arhitekturu. Svakako treba napomenuti da bez obzira na njihovu popularnost i ovde predloženu upotrebu, svaki novi NLU servis bi se na sličan način primenio u celokupnom predloženom rešenju.

Tabela 1. Osnove razlike između Dialogflow i RASA

Dialogflow	RASA
Nema instalacija, odmah se počinje sa radom	Zahteva se instalacija više komponenti
Lakša upotreba i nema potrebe za tehničkim predznanjem	Zahteva tehničko predznanje
Zatvoren sistem	Otvoren sistem, dostupan sors kod
Veb interfejs za razvoj četbota	Nema interfejsa
Ne može da se hostuje na sopstvenim računarima	Hostuje se na sopstvenim računarima
Integrisan sa Google asistenom, Skype, Slack, FB mesindžer itd.	Nema predefinisanih integracija

Oba NLU servisa koriste NLU zasnovanu na ML i trenirani su prirodnim korisničkim iskazima označenim entitetima. *Dialogflow* ima validaciju koja ukazuje da postoji određena namera za koju je potrebno više podataka za trening. Takođe je obezbeđen sistemskim entitetima kao što su brojevi, datum, vreme itd. RASA, sa druge strane može da prilagodi izbor komponenti u NLU planu zasnovan na količini trening podataka. Može da koristi unapred trenirani model koristeći veću količinu podataka. RASA pruža funkcionalnost za procenu namera i entiteta. RASA ima opciju za trening uvezenih podataka različitih formata i iz različitih izvora. U *Dialogflow* ne postoji opcija za uvoženje ili korišćenje podataka o obuci, i prihvata samo jedan red po frazi kao trening podatak. Dakle, ako neko ima mnogo trening podataka i ako četbot ima problema sa razumevanjem većine unosa, onda je izvođenje ovih ispravki u delu za trening u *Dialogflow* prilično dosadan zadatak. Obe platforme pružaju podršku za više jezika. Da bi se podržao Srpski jezik u *Dialogflow*, potrebno je u arhitekturi dodati *Google* prevodilac. RASA omogućava dodavanje podrške ostalim jezicima uz pomoć *spaCy* kao ugrađene komponente. Pored toga RASA omogućava da se ubaci sopstvene komponente. Za jezike koji nemaju unapred trenirane ubačene reči RASA predlaže NLU plan izvršavanja bez *spaCy* biblioteke. Unapred trenirane ubačene reči su od pomoći jer je već kodirano lingvističko znanje. Za najčešće korišćene jezike to je veoma korisna biblioteka jer razume velike količine teksta koji se koristi za izgradnju sistema razumevanja prirodnog jezika. Nažalost, za Srpski jezik nema unapred treniranih ubačenih reči, i to je razlog zašto se nije koristila *spaCy* biblioteka već neke druge komponente. Jedna od prednosti RASA je ta što je u pitanju otvoren kod tj. možete je koristiti bez ikakvih troškova. RASA se može primeniti na sopstvenim serverima u prostorijama svoje kancelarije (engl. On premise) i u oblaku (engl. On cloud). *Dialogflow* skladišti i primenjuje modele u *Google* oblaku. RASA je dodala *Bidirectional Encoder Representation* iz *Transformer*a u NLU planu izvršavanja pomažući u kreiranju boljih modela. RASA takođe pruža dodatnu prednost dodavanja sopstvenog prilagođenog modela u NLU planu izvršavanja za bilo koji zadatak. Iako je RASA veoma agresivna u dodavanju novih karakteristika, one koje zvanično nisu podržane zahtevaju dodatni inženjering za podešavanje odnosno dodatno znanje i trud. Sa druge strane *Dialogflow* pruža laku platformu za korišćenje i jednostavan proces integracije na nekoliko komunikacionih kanala koji smanjuju većinu vremena razvoja u poređenju sa RASA.

Dialogflow je odlična platforma sa dobrim modelima i unapred treniranim entitetima, ali ne podržava prilagođene modele. RASA je takođe dodala podršku za *Tensorboard 2* koji se koristi za vizuelizaciju metrike treninga. Pomaže da se razume da li je model pravilno obučen i može da izvrši izmene hiperparametara na osnovu metrike da bi poboljšali model.

Kao što je već rečeno NLU plan izvršavanja se sastoji od različitih komponenti koje rade uzastopno za obradu korisničkog unosa u strukturani izlaz. Postoje komponente za klasifikaciju namera, za prepoznavanje entiteta, za tokenizaciju, prethodnu obradu, itd. Svaka komponenta obrađuje ulaz i/ili kreira izlaz. Redosled komponenti je naveden u konfiguracionoj datoteci. Isprobano je više konfiguracionih datoteka za RASA test. Da bi izvukli maksimum iz testiranih podataka za trening, treniran je i procenjen naš model na različitim kanalima i različitim količinama podataka. Ponavljajući ovaj proces sa različitim procentima podataka za trening, došlo se do zaključka kako se svaki NLU plan izvršavanja ponaša povećanjem količine podataka za trening. Ceo proces je izveden više puta za svaku navedenu konfiguraciju.

Dialogflow pomaže da se lakše napravi četbot i ne zahteva nikakvo poznavanje veštačke inteligencije. Ali ako postoji potreba za širim spektrom prilagođavanja i više kontrole nad načinom na koji se pravi četbot, RASA bi verovatno bila bolja opcija, iako ostaje premisa: Sve što je složenije, verovatno je i komplikovanije.

4. RAZVOJ ČETBOTA

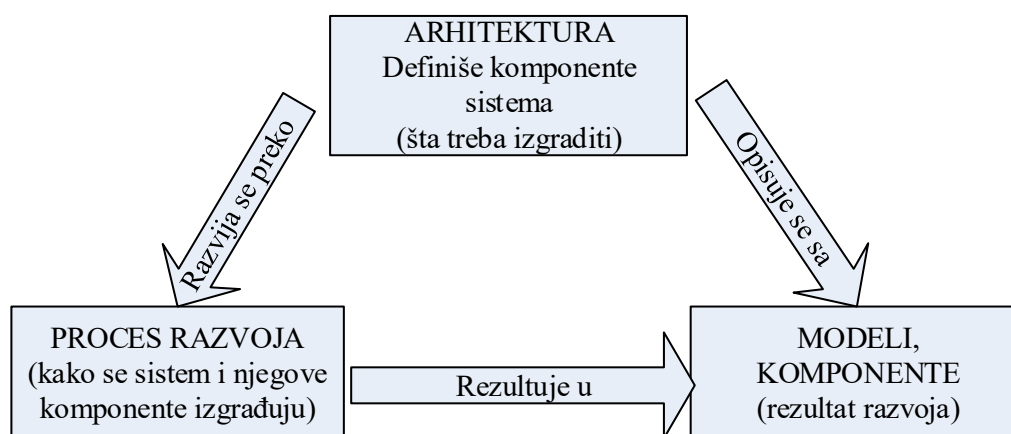
Kao što je već napomenuto, velika brzina kojom četbot i slične tehnologije dobijaju na popularnosti su dovele do toga da su kompanije primorane da preispitaju strategije za razvoj četbota. Više nije dovoljno jednostavno imati klijenta na četu na veb stranicama, ili imati bota na *Facebook*-u. Očekivanja korisnika od aplikacija nastavljaju da rastu, zato je najvažnije imati naprednu, održivu i standardizovanu platformu za razvoj četbota. U svetlu svih tehnoloških razvoja, postaje sve veća potreba postojanja opšte standardizovane arhitekture i metodološkog okvira za razvoj četbotova. Arhitektura je temelj strukture, ili skup struktura, koji mogu da se koriste za razvoj velikog broja aplikacija. Trebala bi da opisuje metodu za projektovanje ciljnog stanja i da sadrži skup alata i pruži zajednički rečnik. Jednostavno rečeno omogućava razumevanje, primenu, održavanje i dalji razvoj četbota.

U toku razvoja bilo kog informacionog sistema najveći problem predstavlja složenost. Za rešavanje konkretnih izazova neophodno je nekako izvući najbitnije koncepte iz velikog broja koncepata koji postoje u stvarnom svetu. Ovaj problem se rešava upotrebom tri osnovna metodološka principa [60]:

1. Dekompozicija sistema na manje delove. Na ovaj način se definiše arhitektura sistema kroz komponente koje treba izgraditi. Arhitektura prikazuje šta treba izgraditi;
2. Arhitektura se razvija preko drugog bitnog principa tzv. procesa razvoja sistema (metodologija razvoja sistema) koji opisuje podelu na faze razvoja čime se određuje životni ciklus procesa razvoja.
3. Modeli, komponente, projektni šabloni, različite biblioteke i aplikacioni okviri su treći bitan princip. Oni omogućavaju da se opiše arhitektura jer se kroz njih vidi od čega se sastoji i izgrađuje predložena arhitektura sistema. Oni su takođe rezultat izabranog procesa razvoja.

Na slici 18. se daje međusobni odnos predloženih metodoloških principa koji predstavljaju osnovne elemente svakog informacionog sistema, iako se oni mogu posebno

nezavisno posmatrati. „Arhitektura IS definiše organizaciju celokupnog sistema, statičke i dinamičke karakteristike pojedinih delova sistema, kao i način njihovog povezivanja u veće celine. S obzirom da u suštini definiše šta treba izgraditi, arhitektura IS upravlja, utiče na, i vodi procese životnog ciklusa IS. Drugačije rečeno, arhitektura sistema, a samim tim i sistem, se izgrađuje kroz pojedine procese. Modeli su takođe jedna forma apstrakcije, uprošćena prezentacija kompleksne realnosti, koja na taj način opisuju komponente definisane u arhitekturi“ [61].



Slika 18. Metodološki principi

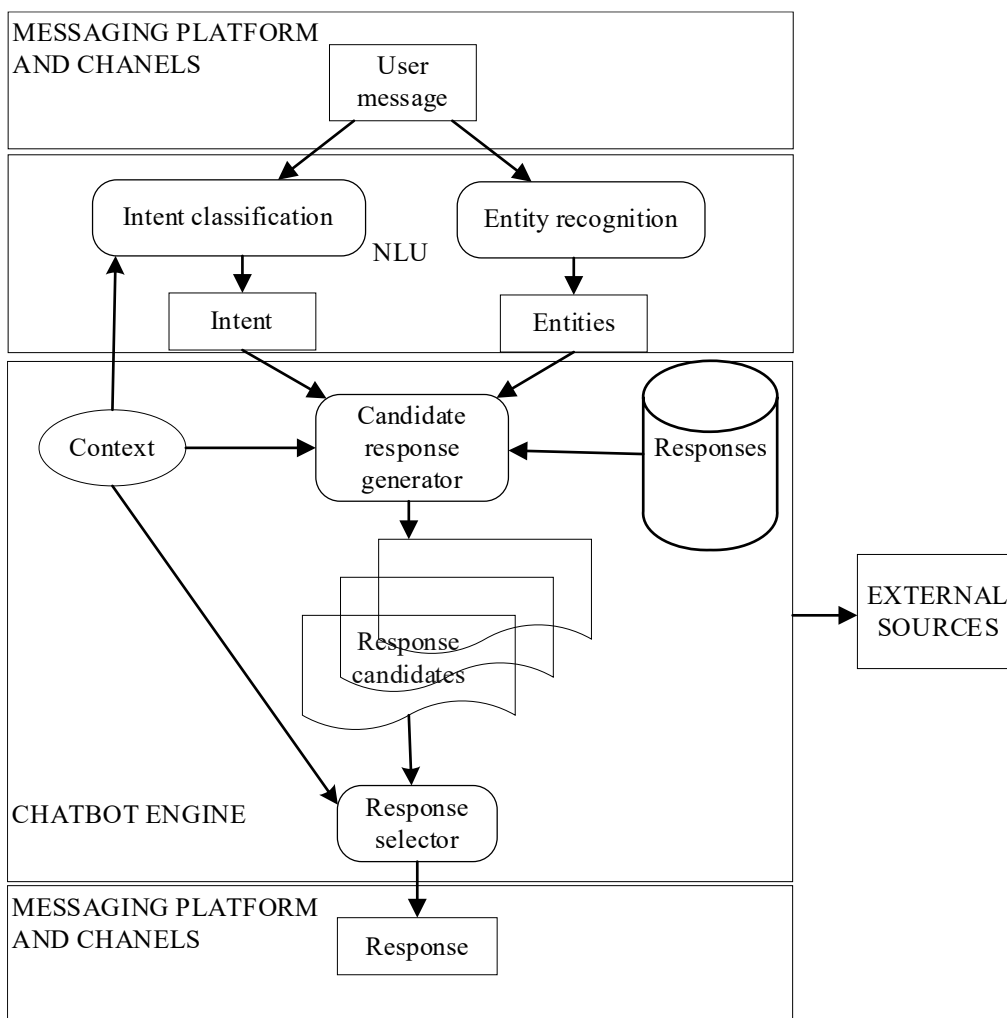
Uz pomoć ovde predloženih metodoloških principa u ovom radu se realizuje jedinstven metodološki okvir za razvoj platforme za kreiranje adaptivnog četbota. Prvo se daje opšta referentna arhitektura, a potom i njemu odgovarajuća metodologija razvoja.

4.1. OPŠTA REFERENTNA ARHITEKTURA

Kao što je već rečeno dizajn i razvoj četbota podrazumeva primenu različitih tehnika. Razumevanje šta treba da ponudi i u koju kategoriju četbot treba da pripada pomaže dizajneru četbota da izabere algoritme ili platforme i alate za njegovu izgradnju. Istovremeno, takođe pomaže krajnjim korisnicima da razumeju šta mogu da očekuju. Zahtevi za dizajniranje četbota obuhvataju tačnu predstavu znanja, strategiju generisanja

odgovora i skup unapred definisanih neutralnih odgovora za poruke korisnika koje četbot nije razumeo.

Prateći opšti metodološki okvir iz prethodnog poglavlja potrebno je da se sistem podeli na sastavne delove u skladu sa standardom kako bi se mogao pratiti modularni pristup razvoju. Proces počinje zahtevom korisnika, koji preko neke komunikacione platforme (npr. *Viber*, *FB messenger*, *WhatsApp*, itd) ili preko aplikacije koja koristi unos teksta ili govora šalje četbotu poruku (frazu, upit, rečenicu). Dakle, obrada poruke počinje od razumevanja korisnikove poruke (engl. user message). Opšti referentni model četbota [62] je prikazan na slici 19.

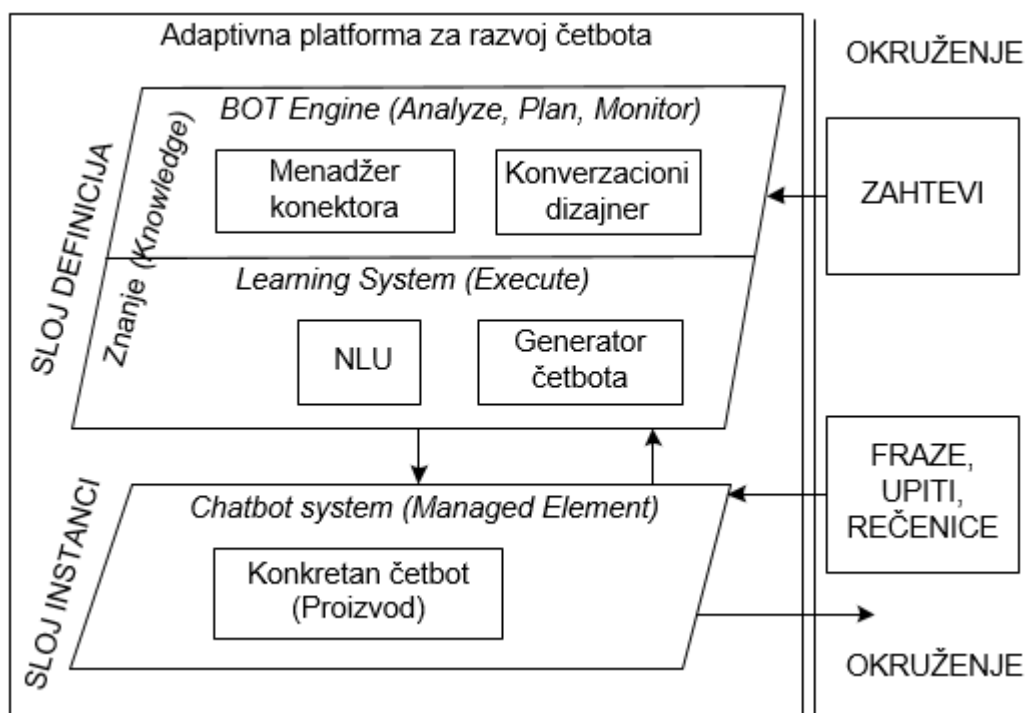


Slika 19. Opšta referentna arhitektura

Nakon što četbot primi korisničku poruku, NLU komponenta za obradu prirodnog jezika je analizira kako bi razumela nameru korisnika i povezane informacije (entiteti, sinonimi i sl.). Komponenta klasifikacije namere identifikuje nameru (engl. Intent classification) korisničke poruke. Obično je to izbor jedne od brojnih unapred definisanih namera, mada sofisticiraniji četbotovi mogu identifikovati više namera iz jedne poruke. Za potrebe izbora namere, klasifikacija namera može koristiti kontekstne informacije, kao što su namere prethodnih poruka, korisnički profil i navike. Komponenta za prepoznavanje entiteta (engl. Entity recognition) izvlači strukturane bitove informacija iz poruke. Na primer, vremenski četbot može izvući lokaciju i datum. Kada četbot dostigne najbolju interpretaciju koju može, mora da odredi kako da nastavi dalje. Četbot može direktno da reaguje na nove informacije, pamti sve što je razumeo i čeka da vidi šta će se sledeće desiti, može da zahteva više informacija o kontekstu ili da traži dodatno pojašnjenje. Kada se zahtev razume, dolazi do izvršenja akcije i pronalaženja informacija. Generator kandidata odgovora (engl. Candidate response generator) radi sve proračune specifične za domen radi obrade zahteva korisnika. On može da koristi različite algoritme, da obavi tražene radnje ili preuzme informacije iz svojih baza znanja ili da pozove nekoliko spoljašnjih API-a, ili čak da zamoli čoveka da pomogne u stvaranju konačnog odgovora ili izvršenja zadatka. Rezultat ovih kalkulacija je lista kandidata za odgovor (engl. Response candidates). Svi ovi odgovori trebali bi biti tačni u skladu sa domenskom logikom. Generator odgovora mora da koristi kontekst konverzacije, kao i namere i entitete izvučene iz poslednje korisničke poruke. Selektor odgovora (engl. Response selector) samo ocenjuje sve kandidate za odgovor i bira odgovor koji bi trebalo najbolje da odgovori na korisnikovu poruku. Četbot može da ima odvojene module za generisanja odgovora i izbor odgovora kao što je NLG. Arhitektura mora da omogući i integraciju sa internim i spoljnim eksternim izvorima podataka (engl. External sources). Imamo primera gde se dodatno ubacuje komponenta za upravljanja dijalogom koja radi slične funkcije kao generator, ali ona je deo samostalnog razvoja programera. Ova komponenta takođe čuva i ažurira kontekst razgovora za koji su vezani trenutna namera, identifikovani entiteti ili entiteti koji nedostaju, a potrebni su za ispunjavanje korisničkih zahteva. Pored toga, ova komponenta traži informacije koje nedostaju, obrađuje pojašnjenja korisnika i postavlja dodatna pitanja.

Ovde predloženu referentnu arhitekturu adaptivne platforme za razvoj četbota na slici 20. čine tri glavna nivoa:

- Okruženje – predstavlja zahteve iz poslovnog okruženja i nove fraze (upite, rečenice) za učenje.
- Sloj definicija (engl. Definition Layer) – ima osnovnu ulogu da kreira sve definicije scenarija četbota, omogućiti različite obrade razumevanja prirodnog jezika i komunikacione kanale, izvršava, adaptira i prati proces konverzacije četbota sa korisnikom.
- Sloj instanci (engl. Instance layer) – predstavlja četbot koji se kreira i koji konstantno uči pod uticajem novih fraza, upita i rečenica.



Slika 20. Referentna arhitektura adaptivne platforme za razvoja četbota

Zbog jednostavnosti, adaptivnosti, mogućnosti ponovnog korišćenja i podele odgovornosti među slojevima, četbotovi se kreiraju kroz dva sloja:

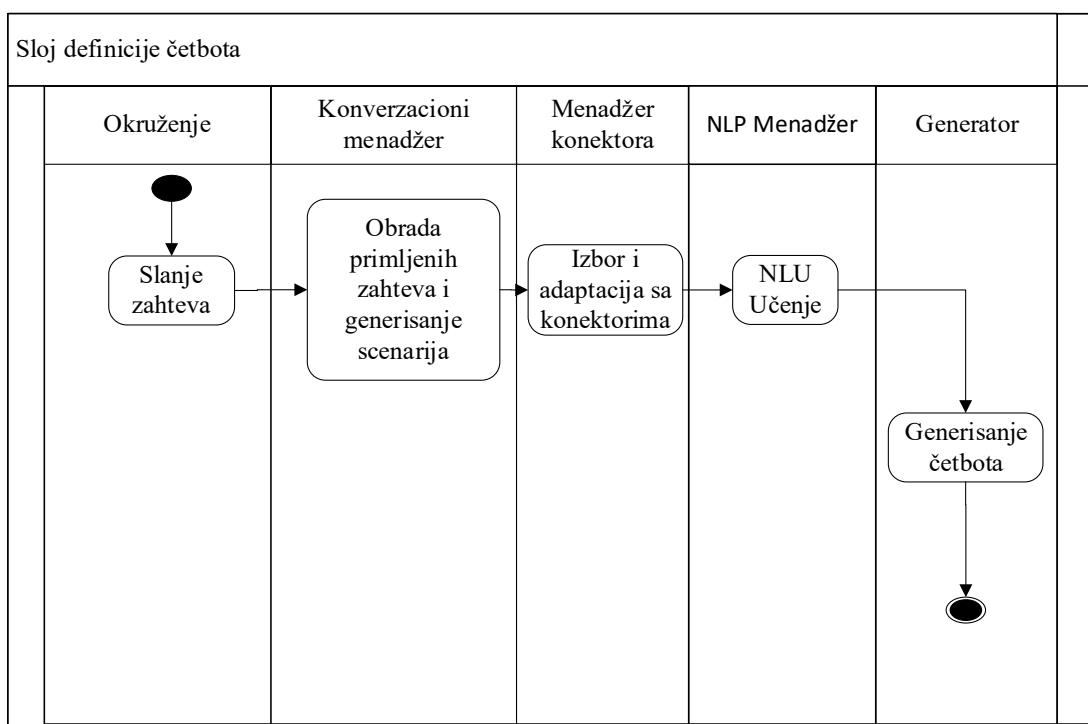
1. U sloju definicija izvršava se generisanja četbota i njemu odgovarajućih definicija scenarija, praćenje stanja četbota i učenje četbota na osnovu toga. Ovaj sloj se sastoji od BOT engine-a i sistema za učenje (engl. Learning System - LS). BOT engine se

sastoji od internih i eksternih konektora i konverzionog menadžera. U ovom sloju se analiziraju zahtevi, planiraju i definišu scenarija i prati njihovo izvršavanje. Ovaj sloj je u stalnom kontaktu sa četbot sistemom (engl. Chatbot System) koji je deo sloja instanci četbota da bi se po potrebi ažurirao konkretan četbot. Sloj definicija se zasniva na konektovanim NLU i komunikacionim kanalima kao osnovnim tehnološkim mehanizmima za razumevanja namere korisnika, generisanje njemu odgovarajućeg odgovora i komunikaciju, dok se generator četbota koristi za razvoj konkretnog četbota. U ovoj arhitekturi elemenat upravljanja je četbot sistem odnosno konkretan četbot. U BOT *engine*-u se koristi pristup zasnovan na cilju kao adaptaciona tehnika rezonovanja gde se adaptaciona (re)konfiguracija predefiniše u vreme projektovanja. Ovaj pristup specificira kriterijume koji karakterišu željenu konfiguraciju sistema. Na ovaj način se obezbeđuje fleksibilnost za definisanje odgovarajućeg ponašanja koristeći visoke ciljeve koji su bliži ljudskom načinu rezonovanja. Dizajner analizira poruke (frazе, upite) koje se prate, zatim se definišu mehanizmi korelacije i kompleksnih situacija kao i kontekst istorije sistema da bi se utvrdilo kad je promena neophodna. Ako je promena neophodna zahtev se prosleđuje generatoru koji se nalazi u sistemu za učenje. On određuje šta treba da se menja i kako da se promeni da bi se proizveo najbolji izlaz. Ovde se definiše koja instanca četbota mora da se menja. Izvršavanje kao deo definicije sistema obezbeđuje mehanizme koji kontrolišu izvršavanje plana sa posebnom pažnjom na rekonfiguraciju konektora. Menja se ponašanje četbot instance, dok se znanje odnosi na informacije koje BOT *engine* čuva o četbotovima.

Kratak normalan scenario, odnosno redosled izvršavanja u sloju definicija predložene referentne arhitekture se opisuje kroz sledeće aktivnosti odnosno kroz dinamički model prikazan na slici 21:

- Zahtevi iz okruženja, kao i fraze u samom četbot sistemu (konkretan četbot) okidaju generisanje ili izmenu četbot sistema.
- Zatim se generišu ili menjaju scenarija uz pomoć konverzionog menadžera koji brine i o kontekstu svakog scenarija. Kontekst se integriše u scenarija čime se obezbeđuje dinamična adaptacija.

- Neophodno je da se definišu svi potrebni interni i eksterni konektori koji omogućavaju pre svega prijem poruka (frazu, upita ili rečenica) od strane korisnika kao i obradu prirodnog jezika bez obzira na proizvođača ovih NLU platformi i komunikacionih kanala.
- NLU omogućava učenje svih potrebnih koncepata za razumevanje namera i entiteta.
- Generator četbota obrađuje sve prikupljene informacije i ako je potrebno generiše novu izmenu četbota (konkretan četbot). Adaptacija može da bude generisanje novog konkretnog četbota i/ili dodavanje novih scenarija i/ili dodavanje novih učenja za postojeća scenarija.
- Na kraju se sve izmene (novi četbot sa novima scenarijima ili poboljšana scenarija – nova učenja) šalju ka četbot sistemu.



Slika 21. Dinamički model sloja definicija

2. Sloj instanci četbota omogućava praćenje stanja konkretnog četbota i obezbeđuje potreban interfejs za prvi sloj. Takođe, sloj instanci treba da omogući funkcionalnosti

konkretnom četbotu, a u vreme izvršavanja odgovoran je za njegovu adaptaciju. Aktivnosti u ovom sloju su:

- Adaptacija četbota uz pomoć novih informacija, menjajući trenutne instance četbota ili generišući novi konkretan četbot.
- Konkretni četbotovi moraju da uče i da se adaptiraju na nove scenarije sa ciljem da budu dostupni, upotrebljivi i pre svega svrsishodni.

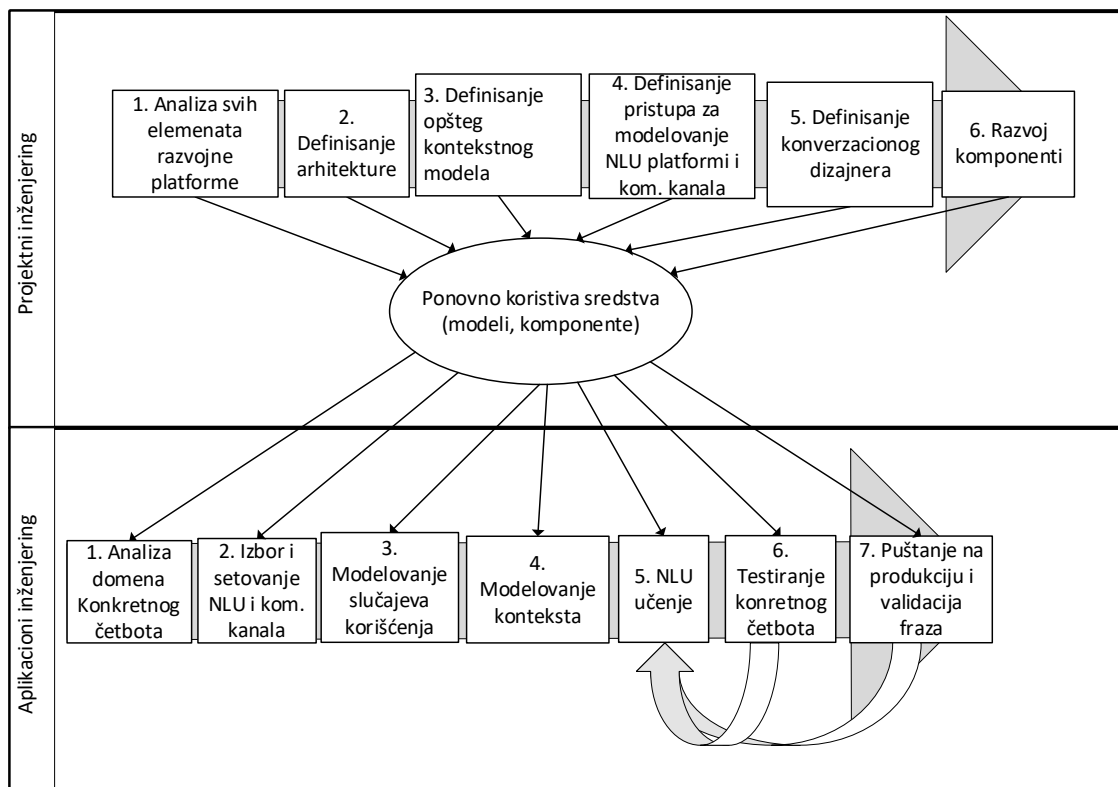
Referentna arhitektura četbota se bazira na već spomenutom pristupu tzv. konektovanim NLU i komunikacionim kanalima. Bitna karakteristika konektovanog NLU je ponovna upotreba mehanizma učenja. Ovo podrazumeva da se učenje može prilagoditi i konfigurisati na razne četbotove. BOT *engine* posmatra rad četbotova i realizuje izmene brinući o kontekstu i adaptaciji četbota. Treba napomenuti da ovakva arhitektura omogućava dinamičnu promenljivost više puta tokom svog životnog ciklusa.

4.2. METODOLOGIJA RAZVOJA ČETBOTA

Razvoj četbota koji nudi dobro iskustvo klijentima zahteva saradnju interdisciplinarnog tima poslovnih analitičara, dizajnera, naučnika podataka, inženjera mašinskog učenja i programera softvera. Metodologija razvoja četbota kombinuje nekoliko modernih okvira i metodologija, uključujući dizajnersko razmišljanje, AI inovacijske sprintove i agilan razvoj softvera.

Predložena metodologija je rezultat kombinacije različitih metodoloških pristupa.

Prateći metodologiju, razvoj adaptivnog četbota zasnovan na konektovanom NLU se sastoji iz dve faze: projektne i aplikacione inženjering. Svaka faza je zatim podeljena na aktivnosti koje treba da urade u okviru faza. Prednost ove podele je realizacija prilagodljive platforme za generisanje četbotova. Shodno prethodno navedenom, metodologija razvoja četbota je prikazana dualnim životnim ciklusom prikazanim na slici 22.



Slika 22. Dualni životni ciklus razvoja adaptivne platforme

I Projektni inženjering

1. Analiza svih potrebnih elemenata razvojne platforme za generisanje četbota
2. Definisane arhitekture platforme za razvoj četbota
3. Definisane opšteg kontekstnog modela
4. Definisane pristupa za modelovanje NLU platformi i komunikacionih kanala
5. Definisane opšteg konverzionog dizajnera za modelovanje tokova
6. Razvoj četbot komponenti

II Aplikacioni inženjering

1. Analiza domena konkretnog četbota
2. Izbor i setovanje NLU servisa i komunikacionih kanala
3. Modelovanje slučajeva korišćenja (scenarija) kroz konverzioni dizajner
4. Modelovanje konteksta
5. NLU učenje
6. Testiranje konkretnog četbota
7. Puštanje četbota u produkciju i validacija fraza

5. PROJEKTI INŽENJERING

U ovom poglavlju se prikazuju glavni rezultati osnovnih aktivnosti projektnog inženjeringa koje su u skladu sa procesom razvoja adaptivnog četbota.

5.1. ANALIZA ELEMENATA RAZVOJNE PLATFORME

U ovoj aktivnosti se identifikuju sve potencijalne kategorije četbotova. Analiza domena identifikuje zahteve kao sredstva i karakteristike relevantne za konkretan četbot. Potrebno je najpre analizirati kategorije četbotova koje razvojna platforma treba da omogući da kreira. S obzirom da smo četbotove klasifikovali u različite kategorije na osnovu predloženih kriterijuma, ovde se predlaže razvoj četbotova koji su:

- Orijentisani i neorijentisani na zadatak;
- U interakciji sa korisnikom tekstualnim i glasovnim putem;
- Zasnovane na veštačkoj inteligenciji kad je u pitanju analiza namere i generisanja odgovora;
- zatvorenog domena, odnosno domenski specifični četbotovi;
- Više intrapersonalni četbotovi, što podrazumeva fokus na lične scenarije gde četbot pomaže korisnicima da ispune lične zadatke u njihovom ličnom životu;
- U mogućnosti da iskoriste ljudsku intervenciju odnosno pomoć četbotu;
- Dobri četbotovi;
- Četbotovi koje će se razvijati sa otvorenim i zatvorenim platformama.

Dizajner treba da identifikuje opseg svih potencijalnih četbotova. To podrazumeva identifikovanje domena i oblasti primene svih četbotova kao i ciljeva svih četbotova i vezi između njih. Ovde se predlažu sledeći domeni i oblasti primene:

- Svakodnevni život - upotreba i integracija četbota za postizanje ličnih ciljeva korisnika u svakodnevnim životnim aktivnostima. Pored opšte namene četbota kao što su pitanja i odgovori (engl. Q&A), ovaj domen uključuje različite oblasti primene specifične za domen kao što je obrazovanje.
- Poslovna podrška - Četbotovi u ovom domenu su dizajnirani kao softverski alati za podršku zaposlenima u internim poslovnim procesima. Glavni cilj je da se obezbedi efikasan, intuitivan interfejs za tradicionalne poslovne procese za postizanje poluautomatskih obavljanje takvih procesa.

Glavni ciljevi integracije četbota u softverske sisteme u razvojnoj platformi su: korisnička podrška, pružanje informacija, izvršavanje odgovarajućih akcija, učenje korisnika (obrazovanje).

Ne treba zaboraviti da se četbotovi naslanjaju na specifične konverzacione kontekste. Ovi konteksti konverzacije spadaju u zatvorene domene. Svaki četbot koji bude napravljen kroz ovde predloženu razvojnu platformu imaće svoje kontekste i biće specijalizovan za odgovore na pitanja vezane za taj domen. Za ovako nešto potrebno je izraditi opšti kontekstni model što je predstavljeno u trećoj fazi projektnog inženjeringa.

Neophodno je takođe izabrati tehnike i tehnologije koje će se koristiti za realizaciju četbota. S obzirom da je prirodni jezik osnova konverzacije većine interfejsa četbota izbor je pao na komplementarne NLP tehnike zasnovane na veštačkoj inteligenciji.

5.2. DEFINISANJE ARHITEKTURA PLATFORME ZA RAZVOJ ČETBOTA

Ovde se daje predlog jedne arhitekture adaptivne platforme za razvoj četbota definišući osnovne komponente. U vreme izvršavanja arhitektura treba da poseduje sva sredstva koja mogu da obrade promene, generišu scenarija, proslede scenarija, dinamički aktiviraju i deaktiviraju scenarija. Arhitektura takođe treba da obezbedi realizaciju sredstava koja su (ne)rekonfigurativnih i da se kao takva mogu testirati. Generator četbota

vrši proces generisanja četbota koristeći *BOT engine* i definicije koje su napravljene uz pomoć konverzionog dizajnera.

Softverska arhitektura i okviri naprednog razmišljanja su zaista važan deo savremenih tehnoloških dostignuća [63, 64].

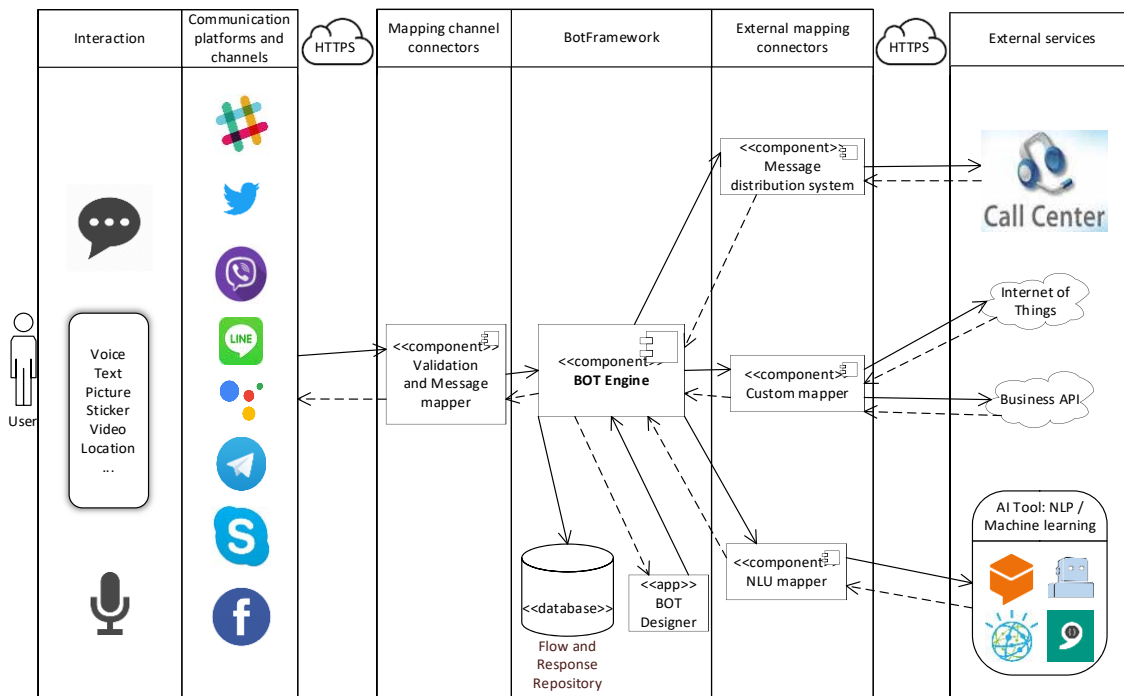
Logička arhitektura ovde predloženog rešenja prikazana na slici 23. koja je u skladu sa prethodno predloženom opštom referentnom arhitekturom iz poglavlja 4.1. Kao što se vidi arhitektura se sastoji od četiri osnovna gradivna bloka:

- Korisnik;
- Komunikaciona platforma (kanal) preko koje korisnici komuniciraju.
- Konverzionu AI platformu.
- Eksterni servisi među kojima je NLU najvažniji.

Četbot platforma treba da uključi *BotFramework* i dva konektora: mapirajući konektori kanala (engl. Mapping channel connectors) i konektore za eksterno mapiranje (engl. External mapping connector). Između ostalog, ova platforma služi i za povezivanje, izgradnju, testiranje i implementaciju inteligentnog četbota. Komunikacija između *BotFramework*-a i oba konektora se obavlja preko protokola *Hipertekt Transfer Protocol Secure* (HTTPS). Cela platforma je zasnovana na mikroservisima. Arhitektura treba da je horizontalno skalabilna. Skalabilnost možemo definisati kao mogućnost sistema da se prilagodi promenama nastalih zbog povećanja opterećenja i to sa ciljem održivosti brzine, pouzdanosti i dostupnosti. Kod horizontalnog skaliranja povećanje performansi se postiže dodavanjem novih računara (čvorova) u mrežu računara. Horizontalno skaliranje zahteva razbijanje sekvencijalnog dela logike na manje delove tako da se mogu izvršavati paralelno na više mašina. Ideja je da se kroz horizontalno skaliranje podiže više instanci iste aplikacije koristeći softverske kontejnere kao npr. tehnologija dokera koji predstavljaju pakete pojedinačne aplikacije koje sadrže sve što je potrebno za pokretanje i izvršavanje. Iako se na jednom serveru istovremeno može izvršavati više kontejnera, isti operativni sistem mora sve da pokreće.

Validacija i mapiranje poruka koje se primaju sa komunikacione platforme vrše se u mapirajućem konektoru kanala. On ima za cilj da pripremi poruku, koja će biti prosleđena sledećoj komponenti u zavisnosti od vrste poruke primljene od komunikacione platforme. Svaka komunikaciona platforma ima svoj metamodel i skup

pravila mapiranja koja su definisana u bazi podataka. Ove podatke u bazi konektor koristi da popuni objekat svim potrebnim podacima i pomogne da bude nezavisan od samo jedne komunikacione platforme.



Slika 23. Logička arhitektura

Životni ciklus poruke počinje na konektoru i tamo se završava jer je njegova uloga da primi poruku i vrati poruku od četbota na kanal sa kog je poruka poslata. Da bi poruka bila primljena sa komunikacione platforme u sistem, mora se konfigurirati metod vebhuka (engl. webhook). U tu svrhu se koristi konektor poruka koji sadrži definisane metode koje prihvataju povratni poziv od komunikacione platforme. Konektor poruka mora prvo da bude pretplaćen na događaje za razmenu poruka odobrene od strane komunikacione platforme i da podesi jedinstveni lokator resursa za povratni poziv (URL). Na osnovu toga četbot platforma će moći da prima događaj poslate sa komunikacionih platformi. Prva funkcija povezane komponente je da prepozna odakle je poruka poslata, odnosno sa koje komunikacione platforme se desio događaj koji je prouzrokovao da sistem primi poruku. Ova komponenta ga prepoznaje jednostavno preko URL-a. Na osnovu URL podatka, komponenta prepoznaje sa koje komunikacione platforme je poruka primljena i preuzima pravila za validaciju i mapiranje konteksta koji je stigao na zahtev. Na osnovu pravila proverava se validnost poruke, odnosno da li je poruku poslata

komunikaciona platforma na koju smo se pretplatili ili ne. Ako je poruka ispravna, može se proslediti modulu za mapiranje. U ovom delu, poruka je mapirana u objekat sa kojim *BotFramework* radi koristeći metamodel i definisana pravila mapiranja. Konektori poruka moraju da obezbede pravilno mapiranje svih tipova poruka za odgovarajuću komunikacionu platformu. Objekti prosleđeni *BotFramework*-u sadrže i podatke o komunikacionoj platformi i korisniku, tako da se poruka može pravilno proslediti pravom korisniku. Navedeno mapiranje funkcioniše i u suprotnom smeru, odnosno kada *BotFramework* treba da vrati poruku ka određenoj komunikacionoj platformi. Metamodeli i pravila mapiranja komunikacionih platformi nisu deo ove teze zbog obima dokumentacije kao i želje da se teza fokusira na metamodele i pravila mapiranja NLU servisa koja će biti objašnjena u poglavlju 5.3.2.

BotFramework se sastoji od mnogih komponenti koje ovaj sistem čine skalabilnim i podržava većinu kriterijuma za izgradnju dobrog okvira preduzeća predloženog u radu [65].

Ovaj okvir je centralni i glavni deo arhitekture koji sadrži celokupnu logiku za obradu primljenih poruka, radi sa kontekstima, tokom konverzacije, logičkim odlučivanjem, integracijom sa unapred definisanim scenarijima i odgovorima, i razna druga potrebna pravila. Sadrži nekoliko važnih komponenti kao što su *BOT engine*, bot dizajner, procesor jezika, generator, itd. Svaka primljena poruka se obrađuje zasebno i prolazi kroz niz faza. Svaki korisnik ima svoju sesiju konverzacije (ćaskanja, razgovora) koja vodi evidenciju o svim porukama između korisnika i četbota. Ovde se takođe vide sve započete, suspendovane i završene instance scenarija koje prate njihovu definiciju scenarija. Zbog toga svaka sesija konverzacije predstavlja jedan ili više scenarija između korisnika i četbota koji sadrži kontekste koji utiču na trenutni i budući tok konverzacije. *Bot engine* je složena komponenta koja čini centralni deo *BotFramework*-a i sastoji se od nekoliko manjih komponenti koje će kasnije biti detaljno opisane.

Eksterni konektori za mapiranje služe za povezivanje sa eksternim servisima kao što su NLU servisi. Ovi konektori takođe imaju posebna unapred definisana pravila mapiranja za svaki specifični tip i implementaciju. Pored unapred definisanih konektora za rad sa gore navedenim servisima, postoje i prilagođeni konektori za eksterne podatke neophodne u konverzaciji u kojima se mapiranja mogu menjati u zavisnosti od potrebe

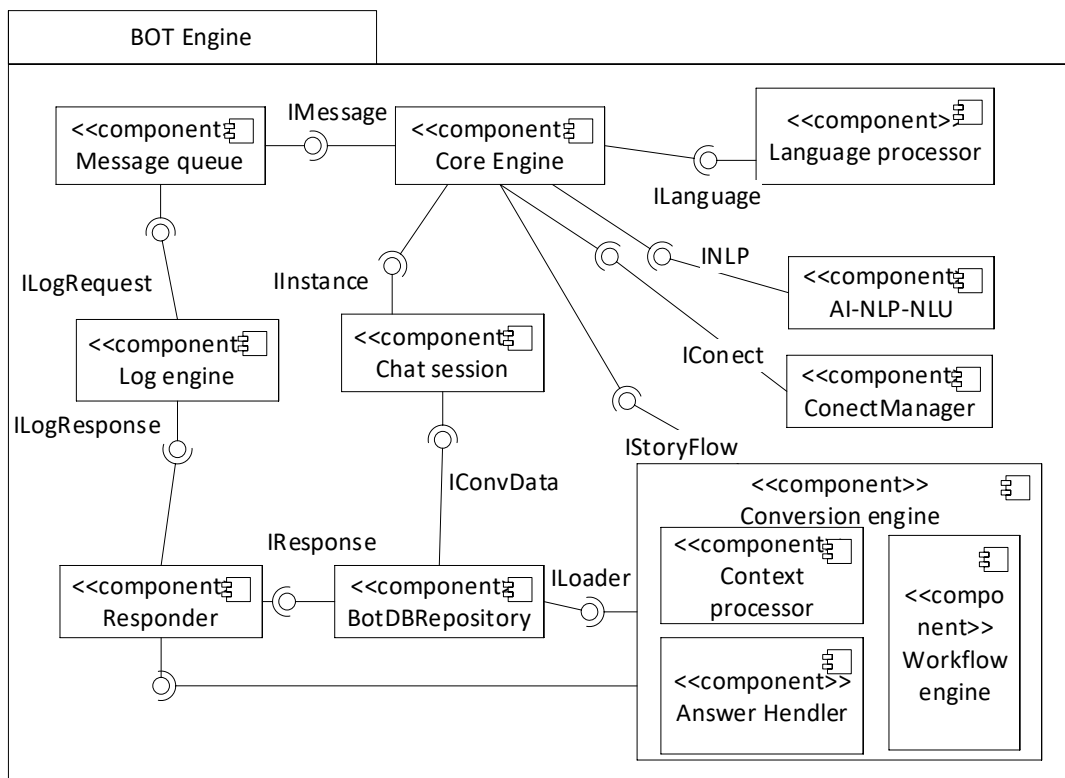
da se bilo koji razgovor uspešno realizuje. Eksterni konektori za mapiranje takođe sadrže komponente koje izlažu servise drugim sistemima koji učestvuju u procesu komunikacije sa korisnikom. Neki od njih su: različiti inteligentni uređaji, poslovni sistemi, postojeći kontakt centri, različiti servisi za obradu prirodnih jezika itd. Ako uzmemo na primer kontakt centar moguće je da se preko eksternog konektora preuzme konverzacija sa korisnikom. *BotFramework* se može integrisati sa postojećom aplikacijom kontakt centra uz pomoć komponente *Message Distribution Connector*. U slučaju složenih pitanja, korisnik se preusmerava i razgovara sa operaterom iz odgovarajućeg kontakt centra. Proširivost predloženog okvira pojednostavljuje integraciju drugih korisnih eksternih usluga.

BotDesigner predstavlja vizuelni alat opremljen kompletom za razvoj softvera za odgovarajuću platformu za implementaciju. On nudi mogućnosti koje olakšavaju interakciju četbot-korisnika i u kojem je moguće je direktno trenirati (učiti) NLU da bi se izbegla zavisnost od NLU interfejsa dobavljača. Takođe, *BotDesigner* omogućava vizuelno generisanje scenarija, odnosno njegovog toka. Tok razgovora mapira sve potencijalne pravce u kojima se diskusija može razvijati, sa mnogo grana za sve mogućnosti. Tok razgovora je odgovoran za predviđanje svih mogućih ulaznih poruka i reakcija četbota. Ovo pomaže korisnicima da brzo ostvare svoje ciljeve i lakše dođu do informacija kao i da daju prioritet poslovnim potrebama. *BotDesigner* treba da je intuitivan i jednostavan za razumevanje i korišćenje.

Svi neophodni podaci za upravljanje toka scenarija (engl. Scenario flow), odgovori zasnovani na kontekstu, identifikovane namere i entiteti se nalaze u repozitorijumu toka i odgovora (engl. Flow and response repository). Ova baza podataka se koristi za čuvanje svih unapred predefinisanih scenarija sa njihovim akcijama i uslovima, odgovora sa potrebnom segmentacijom, konverzacija, korisnika i logova. Osim prethodno navedenog, u bazi se snimaju podaci aktivnosti korisnika, bez obzira da li je četbot uspeo ili nije uspeo da obradi njihova pitanja. Ona prihvata sva podešavanja koja se odnose na konektore, komponente i opšta podešavanja sistema. Svi metamodeli i pravila mapiranja se takođe čuvaju u ovoj bazi podataka.

5.2.1. Struktura komponenti BOT engine-a

U nastavku se daje detaljna uloga i opis svake pojedinačne komponente sa slike 24.



Slika 24. Dijagram komponenti BOT engine-a

BOT engine se sastoji od niza komponenti koje su međusobno povezane. Core Engine (CE) je posebna komponenta koja je centralni deo ovog BOT engine-a. Ova komponenta upravlja drugim komponentama unutar BotFramework-a. Mora da zna tačan redosled izvršavanja akcije, da prati izvršenje akcija i grešaka, da šalje obaveštenja, itd. U zavisnosti od konfiguracije, svaka akcija prolazi kroz log da bi se pratio tok tako složene komponente.

U komponenti Message queue pažnja je posvećena redosledu poruka koje stižu na kanal i njihovom puštanju na obradu. Sve poruke koje ulaze i izlaze iz BOT engine se nadgledaju u sistemu evidencije pomoću ove komponente.

Svaka sesija časkanja ima svoj životni ciklus koji prati najmanje jednu definiciju scenarija kreiranog preko *BotDesigner*-a. Kada korisnik pokrene scenario, komponenta *Chat Session* kreira jednu instancu sesije za tog korisnika.

Korisnik može u svakom trenutku da promeni jezik na kome se obraća četbotu, a komponenta *Language Processor* je sposobna da na osnovu ulazne poruke prepozna jezik i odgovorna je za razumevanje jezika koji se menja. Ukoliko je potrebno, ova komponenta mora da obezbedi najbolji mogući prevod na odgovarajući jezik kao i proveru pravopisa (engl. Spell checker) kako bi AI-NLP-NLU komponenta mogla da obradi tekst.

Na osnovu namere i konteksta, zadatak *Conversation engine* je da odluči koji scenario treba da obradi i da li je to novi scenario ili nastavak postojećeg. Ovde se takođe vrši obrada konteksta koju obavlja *Context Processor* koji upravlja celokupnim kontekstnim modelom. Ova komponenta ima mogućnost da ponudi prepoznate kontekste iz kontekstnog modela i da se četbot u toku razgovora prebaci sa jedne na drugu novu temu (engl. Context switching) i ima opciju da prepozna da li je korisnik želeo da promeni bilo koju vrednost konteksta korišćenu u prethodnom razgovoru (engl. Context filling). Mehanizam toka posla (engl. Workflow) je takođe ugrađen u predloženi okvir. *Workflow engine* ima zadatak da vodi računa o svakom scenariju koji prati njegovu definiciju scenarija, odnosno redosled akcija (radnji) tekuće instance scenarija. Svaka akcija ima kontekste koji bi trebalo da budu ispunjeni da bi se izvršila. Pored toga, svaka akcija ima opciju ugrađene ili prilagođene validacije za validiranje korisničkog unosa i preskakanje obrade AI-NLP-NLU komponente. Primer ovoga je kada neka akcija u nekim scenarijima zahteva broj, imejl, telefon koji se može potvrditi bez komponenti za prevod i AI-NLP-NLU. Komponenta *Answer Handler* ima zadatak da pruži adekvatne odgovore za akcije u okviru scenarija. Odgovori mogu biti različitog tipa: tekst, slika, URL, HTML, audio, dugmići, lokacija, odnosno unija tipova poruka definisanih komunikacionom platformom. Za određenu akcije može se definisati više odgovora istog ili različitog tipa u pravilnom redosledu kojim će biti isporučeni.

Komponenta *Responder* priprema poruku za slanje nazad konektoru poruka da bi je ovaj prosledio krajnjem korisniku. S obzirom da je konektor predstavljen kao mikroservis, ova komponenta se obraća konektoru preko Hipertekt Transfer Protocol (HTTP) RESTful servisa i prosleđuje mu pripremljenu poruku.

Mnoge komponente u *BotFramework*-u se naslanjaju na skladištenje i pristup bazi podataka. Na primer, *Conversation engine* treba da zna za definicije scenarija i njihove instance. *Chat session* radi sa svim korisnicima i njihovom komunikacijom. Takođe je veoma važno zapamtiti sve poruke, entitete i namere. Svi metamodeli i odgovarajuća pravila mapiranja su takođe važni i treba ih čuvati u bazi podataka. *Connect Manager* je odgovoran za ovo mapiranje i održavanje istog, dok je zadatak komponente *BotDBRepository* da se obezbede svi neophodni interfejsi za komunikaciju sa bazom podataka. *BotDBRepository* je otvoren za većinu poznatih sistema za upravljanje bazom podataka kao što su: MS Sql Server, PostgreSQL, MySQL, Oracle itd.

NLU je izazovan problem koji zahteva značajnu ekspertizu. Ovo je jedan od razloga zašto kompanije često odlučuju da ne prave sopstvena rešenja već koriste druge platforme servisa. *Google Dialogflow* [66], *RASA* [67], *Microsoft LUIS.ai* [68], *Facebook Wit.ai* [69] i *IBM Watson* [44] su neke od najpopularnijih NLU servisnih platformi. Oni prihvataju prirodni jezik i vraćaju strukturirane podatke (podaci podeljeni u jednostavan i organizovan format za jednostavnu obradu, na primer JSON). Ove velike kompanije su pretvorile dugogodišnje iskustvo u platforme koje pružaju dobra rešenja manjim ili većim kompanijama za mašinsko učenje. Predložena arhitektura precizno podržava ovaj pristup preko *AI-NLP-NLU* komponente. Poruka (fraza, rečenica) dolazi na ulaz u ovu komponentu kako bi se prepoznala namera korisnika i kako bi se mogla poslati u dalju obradu i detektovati sa svim potrebnim parametrima. Svaki eksterni NLU servis ima sopstvena pravila i mapiranja koja se primenjuju prilikom kreiranja zahteva ka servisiranju i serijalizaciji odgovora. Sva ova mapiranja se setuju prilikom dodavanja servisa za NLU da bi se obezbedila nezavisnost od samo jednog NLU proizvođača. Glavni zadatak NLU komponente je da prosledi otkrivene namere i entitete iz eksternih servisa u pravilnom obliku koji *BOT engine* razume kako bi mogao da nastavi sa procesom. Ova komponenta takođe treba da ima mogućnost da poseduje N usluga koje će biti u jednom/više modu, odnosno aktivnom/pasivnom režimu koji može pomoći u boljem otkrivanju namere. Ako postoji dva NLU servisa u višestrukom režimu, zahtev za obradu se prosleđuje paralelno ka oba servisa. Dva odgovora se uzimaju u obzir i odgovor koji najbolje ispunjava postavljene kriterijume nastavlja obradu. Jedna od glavnih uloga ove komponente je otkrivanje namere i entiteta. Pored ove uloge, ova komponenta ima i druge uloge administrirane u alatu za administraciju eksternog NLU servisa. Svaki NLU

servis ima svoj metamodel i pravila mapiranja, koja koristi AI-NLP-NLU komponenta da popuni objekat svim potrebnim podacima i pomogne da bude nezavisan od samo jednog dobavljača NLU servisa.

5.3. DEFINISANJE OPŠTEG KONTEKSTNOG MODELA

Kontekst (kontekstne informacije) predstavlja jednu od centralnih fokusa u svakoj konverzaciji, pa tako i u komunikaciji sa četbotom. Kontekstno računarstvo omogućava skladištenje kontekstnih informacija vezanih za podatke učesnika u konverzaciji tako da njihovo tumačenje bude što lakše i svrsishodnije. „Sama reč kontekst, potiče od latinskih reči *con* (sa ili zajedno) i *texere* (utkati, isplesti), a ona opisuje i aktivan proces koji se bavi načinom na koji ljudi tkaju svoje iskustvo u celo okruženje, tj. daje neko značenje” [70]. Reč kontekst se veoma često koristi u prirodnom jeziku kao okolnosti nekog entiteta, situacije, okoline, pozadine ili semantike značenja rečenice. Najčešća definicija je napisana u radu [71]: „Kontekst je svaka informacija koja može da se koristi za karakterizaciju situacije entiteta. Entitet je osoba, mesto ili objekat koji se smatra relevantnim ili interakcija između korisnika i aplikacije, uključujući korisnika i aplikaciju“. Kontekst je veoma dinamičan. Promene u okolini, uključujući četbota i korisnika, moraju biti otkriveni u realnom vremenu, a četbot treba da se prilagodi tim promenama. Glavni cilj konteksta je da se generiše manje interakcija između korisnika i četbota.

Bitno je da četbot ima mogućnost tzv. promene konteksta (engl. Context switching). To se dešava kada se razgovara o jednoj temi, a zatim u sred razgovora pređe na drugu temu. Na primer, kada se prijavljuje ispit, student može iznenada da pređe na raspored predavanja. Promena konteksta je čin promene konteksta razgovora, a to se definitivno može zamisliti kao promena teme.

U nastavku je dat primer značaja konteksta za četbota koji može da prijavi ispita ili da pruži informacije o rasporedu predavanju i sl. U ovom primeru može da postoji najmanje 3 slučaja korišćenja:

1. Nema konteksta (kontekstnih informacija), a poruka korisnika glasi: “Projektovanje IS”.
 - S obzirom da nema konteksta, u suštini ne zna se šta je namera korisnika. Međutim, dobar četbot može da prepozna entitet predmet i da postavi pitanje korisniku: „Šta vas interesuje u vezi predmeta Projektovanje IS?“, iako korisnik možda misli da je jasno da želi da prijavi ispit jer je on student.
2. Ima konteksta (kontekstnih informacija) „Prijava ispita“, a poruka korisnika glasi: “Projektovanje IS”.
 - Korisnik je naveo da želi da prijavi ispit iz predmeta „Projektovanje IS“. S obzirom da postoji kontekstna informacija „Prijave ispita“, jasno je da korisnik želi da prijavi ispit, a ne da pita kad su predavanja. U ovom slučaju poruka „Želim da prijavim ispit iz Projektovanja IS“ i poruka „Projektovanje IS“ treba da odgovaraju istoj nameri. Odgovor na pitanje kako se došlo do kontekstnih informacija da je u pitanju prijava ispita može se dobiti iz istorije prethodne konverzacije korisnika sa četbotom ili kontekstnih informaciju kao što su: „četbot komunicira sa aktivnim studentom“, „završena su predavanja za tekući semestar“ i „u toku je prijava ispita“. U najnesigurnijem scenariju četbot bi jednostavno mogao korisniku da postavi sledeće pitanje: „Pretpostavljam da želiš da prijaviš ispit – Projektovanje IS?“ i da ponudi opcije „Da“ i „Ne“.
3. Ima konteksta (kontekstnih informacija) „Raspored predavanja“, a poruka korisnika glasi: “Projektovanje IS”.
 - Korisnik je naveo da želi da sazna kad su predavanja iz predmeta „Projektovanje IS“. S obzirom da postoji kontekstna informacija „Raspored predavanja“, jasno je da korisnik želi da sazna kad su predavanja, a ne da prijavi ispit. U ovom slučaju poruka „Kad su predavanja iz predmeta Projektovanja IS“ i poruka „Projektovanje IS“ treba da odgovaraju istoj nameri. Odgovor na pitanje kako se došlo do kontekstnih informacija da je u pitanju „Raspored predavanja“ može se dobiti iz istorije prethodne konverzacije korisnika sa četbotom ili kontekstnih informaciju kao što su: „u toku su predavanja za tekući semestar“ i „prijava ispita nije u toku“. U najnesigurnijem scenariju četbot bi jednostavno mogao korisniku da postavi

sledeće pitanje: „Pretpostavljam da želiš da saznaš kad su predavanja iz predmeta Projektovanje IS?“ i da ponudi opcije „Da“ i „Ne“.

Još jedan odličan primer je odgovor korisnika „Da“ ili „Ne“, čije značenje u potpunosti zavisi od konteksta („Da“ može biti jednako „Da, želim da prijavim ispit iz predmeta Projektovanje IS“ ILI „Da, želim da saznam kad su predavanja iz predmeta Projektovanje IS“).

Ovi primeri pokazuju dve stvari:

- Jedna reč može imati različite namere u zavisnosti od konteksta dijaloga.
- Iste poruke (frazе, upiti, rečenice) mogu se pojaviti u više namera.

Ako četbot koristi kontekst za dobijanje relevantnih informacija i/ili pružanje usluga korisniku, gde relevantnost zavisi od zadatka i potrebe korisnika onda se kaže da je četbot kontekstan. Zbog toga, kontekst dobija ogroman značaj u razvoju četbotova. “Formalna predstava konteksta je neophodna za proveru konzistentnosti, kao i da se osigura rasuđivanje na osnovu kontekstnog modela” [72]. Kontekstni model je neophodan za izbor scenarija koji će da se izvrši u svakoj potencijalnoj instanci kontekstnog modela. Jedna od najbitnijih karakteristika četbota treba da je mogućnost adaptacije pomoću kontekstnog modela. U primeru četbota kontekstnost podrazumeva sposobnost četbota da otkrije i reaguje na osnovu konteksta (kontekstnih informacija). Četbot je kontekstan ako okida odgovarajući servis i/ili prilagođava svoje ponašanje u odnosu na promene u kontekstu. “Četbotovi treba da su u stanju da prilagode svoje ponašanje u skladu sa promenama situacije bez eksplicitne intervencije korisnika, a modelovanje konteksta je jedan od najvažniji koraka za razvoj kontekstnih sistema” [60]. Modelovanje konteksta je način dobijanja kontekstnih informacija, u obliku kontekstnog modela, za potrebe učenja, rasuđivanja i/ili reagovanja na kontekst u četbotu. “Modelovanje konteksta predstavlja specifikaciju svih entiteta i veza između njih koji su potrebni da bi se opisao kontekst u celini” [73].

S obzirom na prethodno sve rečeno, dizajner konteksta mora da modeluje sve kontekstne informacije (kontekstni model) za moguće konkretne četbotove. Da bi definisali odgovarajući kontekstni model u ovoj tezi se prihvata predloženi referentni

kontekstni model sa slike 25. Na ovoj slici se vidi deset kategorija konteksta čime se želi postići odgovarajuća granularnost. “Predložena kategorizacija daje sintezu velikog broja predloženih kategorizacija i jasne razlike između kategorija” [60]. Konkretni kontekst (konkretna kontekstna informacija) se dobija iz predloženih dimenzija odgovarajućih kontekstnih kategorija za svaki odgovarajući domenski problem odnosno konkretni chatbot. Sve ove kategorije su veoma bitne za interpretaciju situacije i razumevanje dalje akcije (odgovora) svakog budućeg chatbota. U nastavku su date kontekstne kategorije sa svojim dimenzijama:

1.) Korisnik:

- Interni atributi: (ljubav, mržnja, razni osećaji, lični profil i osobine);
- Eksterni atributi: (prezime, ime, visina, starost, pol, težina, itd.).

2.) Aplikacija:

- Ciljana platforma, zahtevi i mogućnosti aplikacije, potrebna memorija, itd.

3.) Okruženje:

- Fizičko okruženje: (vremenski uslovi, temperatura, realni koncepti i entiteti, gužva, brzina kretanja, vidljivost, zvuk, objekti u blizini, itd.);
- Digitalno okruženje: (računarska mreža, virtualna realnost, itd.).

4.) Kontekst aktivnosti (namere): opisuje šta radi entitet, a može da se opiše sa eksplicitnom namerom, ciljem, zadacima tj. strukturom zadatka i akcijama npr. prijava ispita. Kontekst aktivnosti pokriva aktivnosti u kojem je trenutno entitet i u koje će se ubuduće uključiti i daje odgovor na pitanje šta je njegova namera odnosno: „Šta entitet želi da postigne i kako?“.

5.) Vreme:

- Definisano: (vreme dešavanja i situacije, plan obilaska, vremenska zona, tačan period godine ili dana, itd.);
- Nedefinisano: (povremeno, vreme od naručivanja do isporuke, deo dana, itd.).

6.) Lokacija:

- Fizička lokacija koja se deli na apsolutnu (tačna adresa) i relativnu (put od kuće do posla);
- Virtualna: (jedinствeni identifikator resura - URI).

7.) Informacija: (osobina značenja delova informacija dobijena u raznim oblicima ili formatima npr. tabela, slika, tekst i sl.).

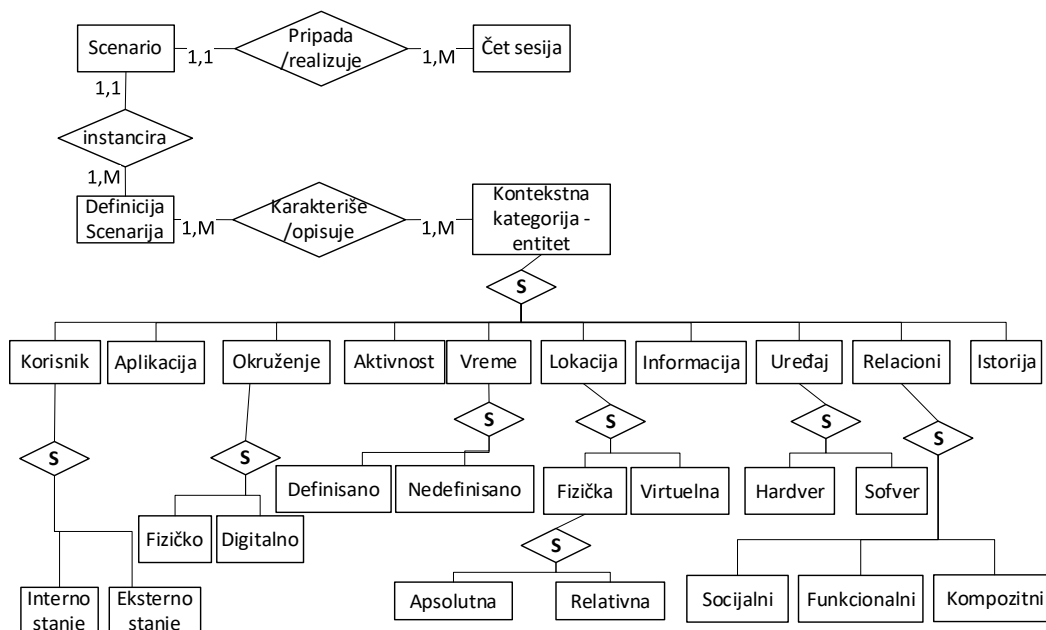
8.) Uređaj:

- Hardver: (Procesor, memorija, itd.);
- Softver: (dostupnost softvera u nekim uređajima).

9.) Relacioni kontekst [74]: predstavlja vezu između različitih kontekstnih dimenzija, ovaj kontekst agregira i predstavlja različite tipove relacija između elemenata određenog kontekst orijentisanog setovanja. U ovu kategoriju kontekstnih informacija spadaju relacije između entiteta (osobe, stvari, uređaji, servisi, tekst, film, slike, zvukovi). Relaciona kategorija je podeljana na socijalnu, funkcionalnu i kompozitnu:

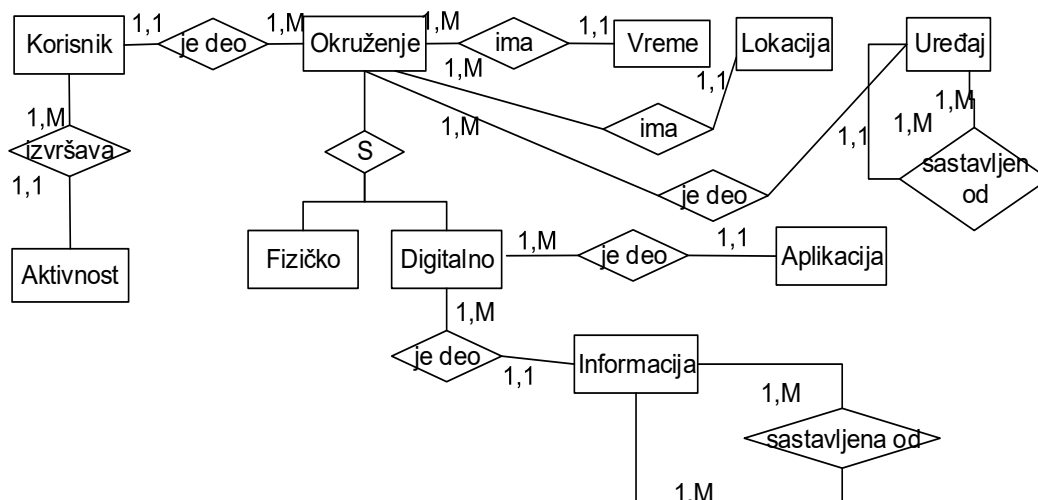
- Socijalna relacija: podkategorija koja opisuje socijalni aspekt tekućeg konteksta entiteta. Obično su to međuljudske relacije kao što su socijalne asocijacije, srodstvo, ili zajedničke potrebe između dve ili više osoba. Na primer, socijalne relacije mogu da sadrže informacije o prijateljima, neprijateljima, komšijama, članovima istog kluba, rođacima, kupcima, kolegama itd.
- Funkcionalna relacija: govori nam da jaden entitet koristi drugi entitet za određene potrebe i sa određenim efektom. Takve relacije pokazuju fizičke osobine kao što su korišćenje čekića, sedenje na stolici ili rad za računarom. Pored toga, ova vrsta relacije pokazuje komunikacione i interakcione osobine kao što su kucanje u tekst editor ili pričanje na mikrofona. Zatim mentalne i kognitivne osobine kao što su čitanje članka, prezentovanje ili razumevanje nekog koncepta.
- Kompozitna (agregacija i asocijacija) relacija: predstavlja relaciju između entiteta kao celine i njenog dela. Na primer ljudsko telo ima ruke i noge, zajednički štampač pripada različitim odeljenjima i ljudima itd.

10.) Istorijski kontekst: (istorijski bitne informacije).



Slika 25. Referentne kontekstne kategorije

Slika 26. prikazuje odnos kontekstnih kategorija koje su takođe deo predloženog referentnog kontekstnog modela [60].



Slika 26. Odnos između kategorija konteksta

Bez obzira što se u ovoj tezi predlaže kontekstni model koji je nezavisan od proizvođača NLU platforme, zbog razumevanja i značaja konteksta ipak će se ukratko objasniti kako se upravlja kontekstnim informacija od strane dve najpopularnije NLU platforme, *Dialogflow* i *RASA*. Naime, *Dialogflow* zaista u svojoj implementaciji koristi

kontekste koji predstavljaju trenutno stanje zahteva korisnika i koji omogućavaju četbotu da prenosi informacije od jedne namere do druge.

Ovo znači da je kontekst lepak koji povezuje namere. Kontekst predstavlja neophodan uslov koji mora da postoji pre nego što se namera okine (pokrene). On mora biti prisutan u prethodnoj kao i u trenutnoj nameri. Konteksti omogućavaju da se kontrolišu tokovi razgovora tako što omogućavaju da se definišu određena stanja u kojima konverzacija mora da bude da bi se pogodila određena namera. Normalno, *Dialogflow* pogađa namera ako trening fraze veoma liče na poruku krajnjeg korisnika. Međutim, kada se kontekst dodeli nameri, *Dialogflow* će uzeti u obzir tu nameru za podudaranje samo ako je kontekst aktivan. Postoje dve vrste konteksta (ulazni i izlazni kontekst) koje omogućavaju da se aktiviraju i deaktiviraju konteksti, a koji mogu da kontrolišu tok konverzacije. Kada se dodeli nameri, ulazni kontekst govori *Dialogflow* da se podudara sa namerom samo ako se poruka krajnjeg korisnika blisko poklapa i ako je taj kontekst aktivan. Kada se dodeli nameri, izlazni kontekst govori *Dialogflow* da aktivira kontekst ako već nije aktivan ili da održava kontekst nakon što se namera poklopila.

S druge strane *RASA* koristi koncept slotova koji mogu, ali ne moraju da utiču na tok scenarija. *RASA* koristi slotove za upravljanje kontekstom. Slotovi u *RASI* predstavljaju memoriju četbota. Slotovi čuvaju delove informacija na koje četbot treba da se pozove kasnije i oni mogu, ali ne moraju da usmere tok konverzacije. Gde god je potrebna perzistentnost u čuvanju vrednosti, predlaže se korišćenje slotova. *RASA* daje veliku fleksibilnost kada je u pitanju definisanje i korišćenje slotova. Postoje različiti tipovi slotova kao što su: tekstualni, logički, pa čak i prilagođeni tipovi. Na osnovu tipa slota može se izabrati da li slot treba ili ne treba da utiče na tok konverzacije. Dakle, način na koji slot utiče na konverzaciju zavisice od njegovog tipa.

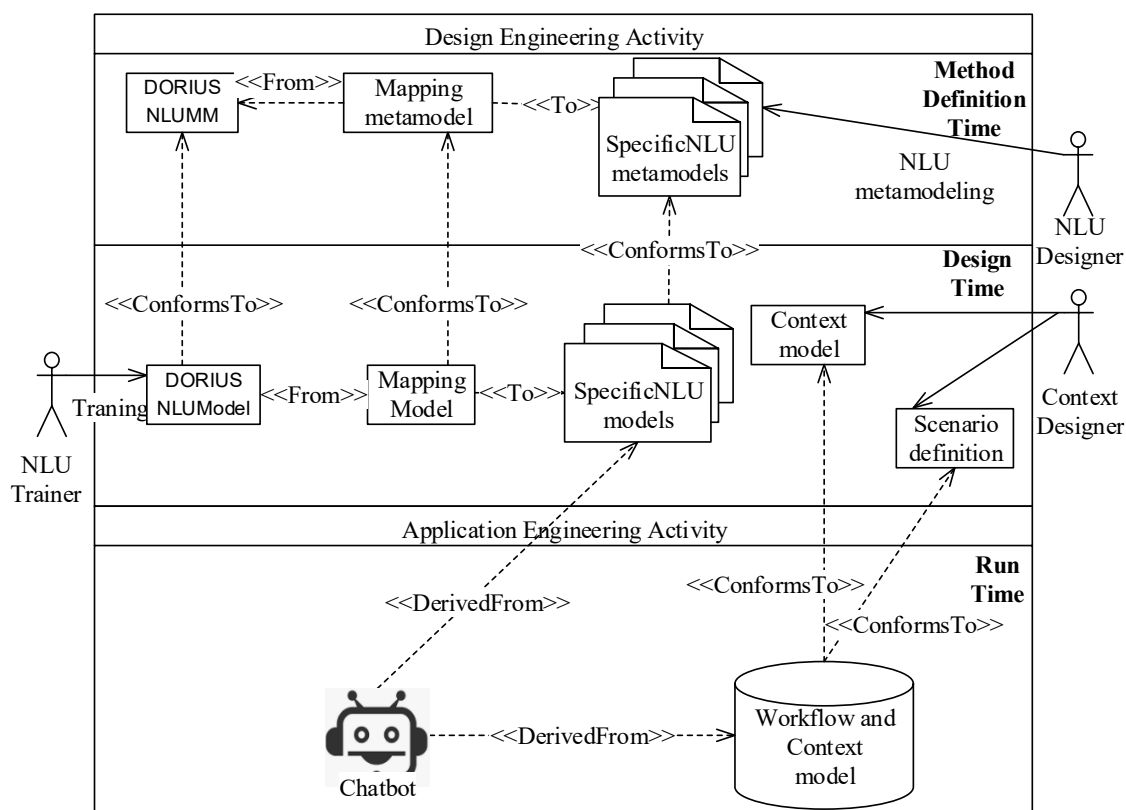
5.4. DEFINISANJE PRISTUPA ZA MODELOVANJE NLU PLATFORMI I KOMUNIKACIONIH KANALA

Predloženi pristup je najpre motivisan idejom da se razvoj četbota zasniva na nezavisnosti od NLU proizvođača i komunikacionih kanala odnosno da se obezbedi jednostavan mehanizam koji će lako da obezbedi uvođenje svake (nove) NLU tehnike ili komunikacione platforme za razmenu poruka. NLU nezavisnost se definiše kroz tri glavna dela: (1) generički metamodel koji definiše opšte koncepte NLU-a i njihovih odnosa, (2) generički metamodel specifičnog NLU servisa (npr. *Dialogflow*, *RASA*) i (3) skup pravila mapiranja koji preslikavaju svaki koncept metamodela specifičnog NLU servisa u koncept generičkog metamodela. Nezavisnost od komunikacionih kanala se takođe definiše kroz tri glavna dela: (1) generički metamodel koji definiše opšte koncepte svih komunikacionih kanala i njihovih odnosa, (2) generički metamodel specifičnog komunikacionog kanala (npr. FB mesindžer, Viber) i (3) skupa pravila mapiranja koji preslikavaju svaki koncept specifičnog komunikacionog kanala u koncept generičkog metamodela.

Kao što je već rečeno, u ovoj tezi će se opisati samo NLU nezavisnost zbog obima dokumentacije. Ceo pristup koji se ovde objašnjava se slično može primeniti i na nezavisnost od komunikacionih platformi.

Platforma za razvoj adaptivnog četbota treba da prati pravila mapiranja, a zatim da izvrši ukupnu logiku četbota. Pristup koji se predlaže je baziran na agilnom razvoju, gde su sredstva predstavljena kroz modele i metamodele. Mapiranje i cela arhitektura modela je data koristeći dijagram paketa sa slike 27. Ovi paketi u UML dijagramu predstavljaju modele, dok su razne veze između modela predstavljene stereotipnim asocijacijama zavisnosti između odgovarajućih paketa. Ovde se prikazuje jedna potpuna arhitektura modela koja identifikuje sve modele (uključujući njihove metamodela) i njihove veze potrebne za ovaj pristup. U odnosu na vreme kad su kreirane, dijagram klasifikuje modele u tri kategorije koje su predstavljeni kao plivačke staze.

- ✓ *Method definition time* obuhvata modele, odnosno metamodele kreirane u vreme definicije modela koje definišu NLU dizajneri. Ovde se prvenstveno misli na generički DORIUS metamodel, specifične metamodele određenih NLU platformi i metamodele međusobnih mapiranja između opšteg i specifičnog NLU metamodela.
- ✓ *Design time* obuhvata DORIUS modele koje kreiraju NLU treneri. Ovde takođe spada automatsko kreiranje specifičnih NLU modela tokom projektovanja četbotova, kao i kreiranje kontekstnog modela od strane kontekstnog dizajnera.
- ✓ *Runtime* kategorija uključuje razumevanje namera korisnika od strane četbota koji koristi specifične NLU modele i kontekstne informacije iz prethodno definisanog kontekstnog modela.



Slika 27. Arhitektura modela

Shodno tome, sredstva koja se kreiraju u kategoriji *Method Definition* i *Design time* uglavnom odgovaraju metamodelima, modelima i modelima mapiranja. Kreiranje sredstava se deli u dve različite kategorije, a to su: apstraktna sredstva i konkretna

sredstva. Apstraktna sredstva su potrebna kako bi se kreirala konkretna sredstva koja se koriste u toku korišćenja četbota. Apstraktna sredstva uglavnom odgovaraju metamodelima, a konkretna sredstva uključuju modele i modele preslikavanja odnosno generatore koda. Mapiranje opšteg DORIUS NLU metamodela i specifičnog NLU metamodela pomaže u razvoju četbota i automatizaciji procesa definisanja pristupa nezavisnosti od NLU proizvođača. Takođe treba napomenuti da je korišćenjem ovog pristupa omogućeno lakše prebacivanja starih četbotova, koji rade na samo jednom NLU servisu, na neki novi NLU servis uz već spomenuti DORIUS NLU metamodel.

S obzirom da se model objekti veze (MOV) koristi kao jezik definicije ontologije [75, 76, 77, 78], opšti NLU metamodel se definiše koristeći šeme MOV koncepata. Ta šema mora biti u skladu sa MOV metamodelom. Metamodeli bilo kog specifičnog NLU servisa se koriste za predstavljanje njemu odgovarajuće NLU platforme, takođe u skladu sa odgovarajućim MOV metamodelom. Definicija mapiranja se dobija modelom mapiranja koji mapira koncepte šeme opšteg DORIUS NLU metamodela na koncepte odgovarajućeg specifičnog NLU metamodela. Definisano mapiranje treba da prati pravila i ograničenja koja su definisana od strane metamodela mapiranja. Kontekstni model i definicije scenarija su napravili kontekstni dizajneri jer su oni ti koji definišu i implementiraju slučajeve korišćenja. Zahvaljujući ovoj ulozi, oni su odgovorni za efikasan rad četbota.

Suština ovog poglavlja se zasniva na realizaciji mapiranja između opšteg DORIUS NLU metamodela i specifičnog NLU metamodela odgovarajućeg proizvođača kao i prikazu odgovarajućih metamodela i pravila mapiranja između njih.

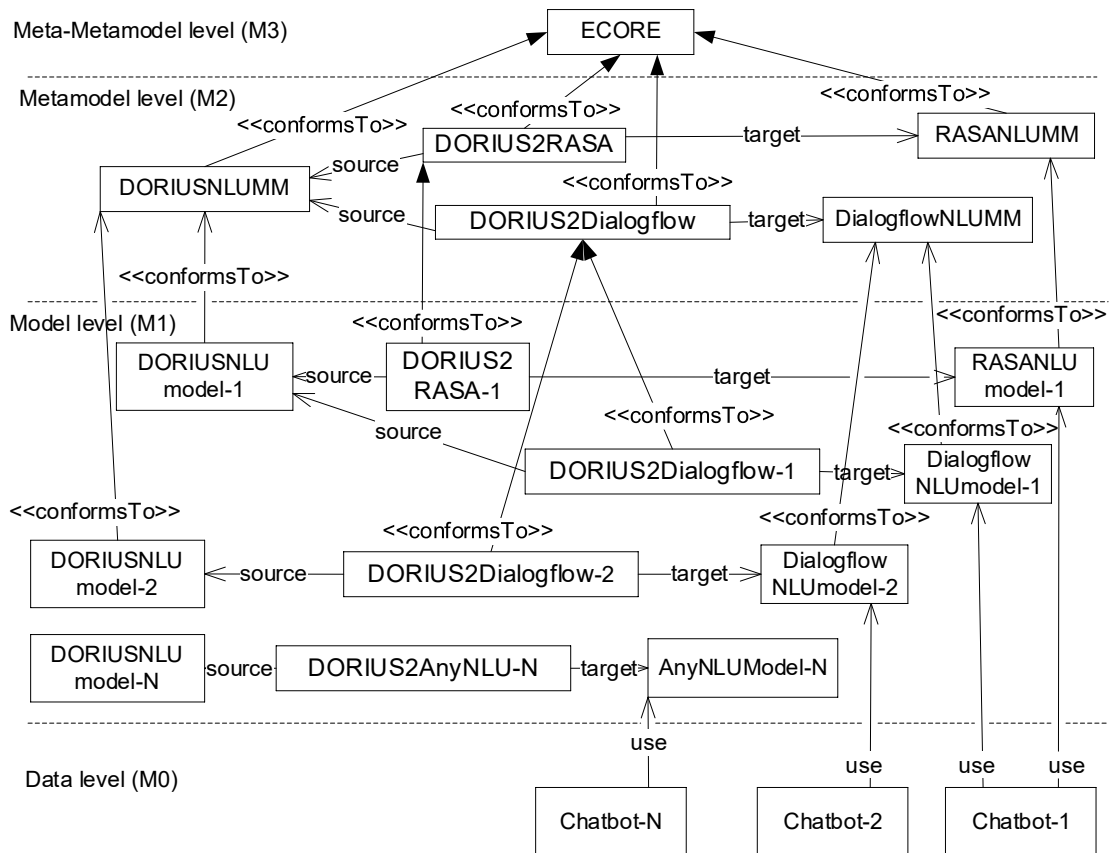
5.4.1. Opšti postupak kreiranja adaptivnog četbota nezavisnog od NLU platforme

Opšti postupak za kreiranje četbota nezavisnog od NLU platforme sa slike 28. prolazi kroz sledeće korake:

1. Najpre se pravi opšti NLU metamodel u ovom pristupu nazvan DORIUS metamodel.

2. Nakon toga je potrebno za svaki specifični NLU servis, koji je kreirao određeni proizvođač, napraviti njemu odgovarajući metamodel.
3. Za svaki od ovih specifičnih platformi tj. njihovih NLU metamodela se vrši mapiranje sa DORIUS NLU metamodelom čime se dolazi do pravila mapiranja.
4. Zatim u fazi aplikacionog inženjeringa klijent, za koga se generiše četbot, izabere odgovarajuće NLU servise.
5. Nakon analize slučajeva korišćenja, definisanja scenarija i kontekstnog modela, NLU trener treba da unese sve vrednosti (učenje) potrebne za DORIUS NLU model i automatski se vrši preslikavanje konkretnih vrednosti iz DORIUS NLU modela u izabrane NLU modele prateći pravila iz modela mapiranja. Rezultat ovog preslikavanja su jasni modeli mapiranja između ovih modela.
6. Koristeći novo generisane NLU modele i kontekstne informacije odgovarajućeg kontekstnog modela koje u stvari predstavljaju realni kontekst (za konverzaciju i trenutnu situacija), dobija se konfiguracija četbota. Za različite NLU servise mogu se dobiti i različite konfiguracije četbota.

Na slici 28. se jasno vidi kako se DORIUS NLU model-1 mapira na dve NLU platforme (*Dialogflow* i *RASA*), dok *Chatbot-1* može da koristi ova dva NLU servisa.



Slika 28. Opšti postupak za kreiranje adaptivnog četbota

”Model preslikavanja je izabran za mapiranje jer nudi nekoliko bitnih prednosti. Ako prihvatimo princip da je sve model, moguće je praktično obuhvatiti sve informacije kroz modele, kao i veze i korespondencije između modela. Ovo omogućava da se smeste, analiziraju i ažuriraju veze (linkove) sa istim alatom za modelovanje u kojem smo i modelovali. Ovaj način mapiranja najčešće nije automatizovan proces” [60]. U principu za ove modele preslikavanja sve važi kao i za modele tkanja [79] tj. oni su posebni modeli za sebe. Razlika je što oni ne koriste koncepte iz modela tkanja, već koncepte MOV-a.

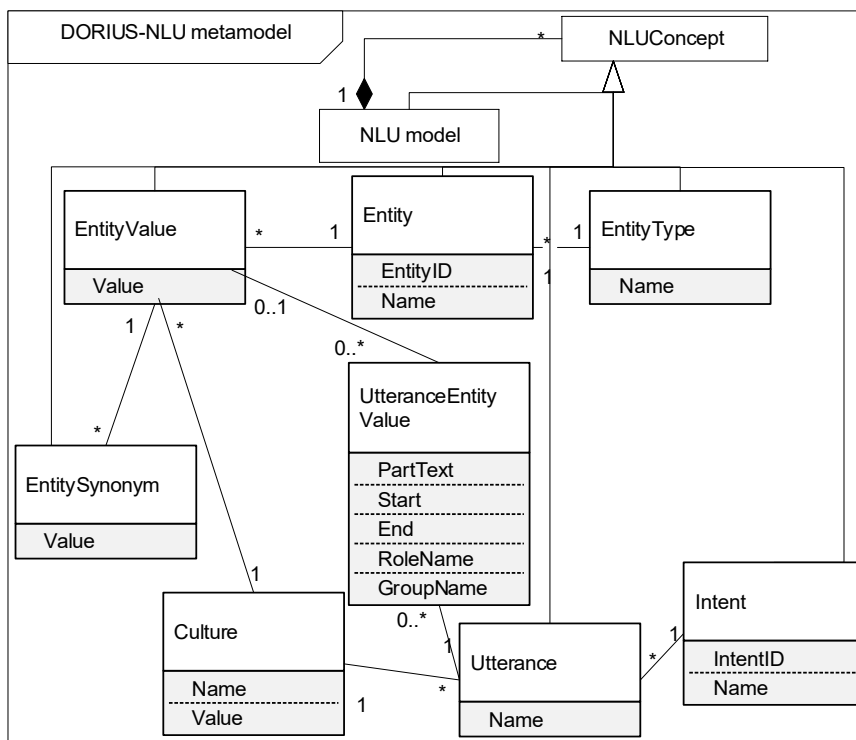
“Ako se postavi pitanje zašto koristiti model preslikavanja umesto modela transformacije, treba odgovoriti da model preslikavanja najčešće nije automatski proces. Za transformaciju, ulaz je jedan model, dok je izlaz neki drugi model. U preslikavanju za ulaz se koriste dva modela, a nastaje jedan model preslikavanja. Model transformacije je u skladu sa fiksnim metamodelom (metamodel jezika transformacija), dok u modelu

preslikavanja to nije slučaj jer za svaku različitu NLU platformu treba napraviti novi metamodel. Preslikavanje izbegava da se pravi jedan veliki metamodel „za sve“. Umesto ovoga, odvojeno se čuvaju individualni metamodeli, jer se svaki metamodel bavi svojim domenom i lakše se radi sa njima, dok su u isto vreme oni međusobno povezani u mreži metamodela. Interesantno je napomenuti da model preslikavanja najbolje funkcioniše kad su metamodeli koji učestvuju u ovome zasnovani ili opisani kroz isti (meta) metamodel” [60].

5.4.2. NLU metamodeli i odgovarajuća pravila mapiranja

NLU je mehanizam za razumevanje prirodnog jezika koji klasifikuje poruke prema namerama i izdvaja relevantne informacije iz porukea koji se nazivaju entiteti. Glavna uloga NLU je da pokuša da razume unos korisnika, nameru razgovora i izdvoji entitete koji su potrebni da bi se izvršila akcija korisnika ili poslovna logika. O NLU servisima se može razmišljati kao o skupu API-ja visokog nivoa za pravljenje sopstvenog jezičkog parsera koristeći postojeće NLU i biblioteke mašinskog učenja.

”Metamodeli koji se ovde koriste su komplementarni sa *Meta Object Facility* (MOF). Ova veza između dva (meta) modela ne mora implicitno da predstavlja transformaciju iz jednog u drugi ili obrnuto. Ona nam jednostavno govori da ova dva elementa koja su povezana po nekom pravilu dele u stvari neki semantički link, npr. odgovoran je za, izmeren je sa, utiče na, itd.” [60]. Model preslikavanja daje pravila mapiranja elemenata opšteg NLU metamodela u elemente specifičnih NLU metamodela svih proizvođača. Nakon definisanja odgovarajućih metamodela najpre je neophodno odgovoriti na pitanje kompatibilnosti metamodela njihovih različitih modela. Potrebno je definisati pravila mapiranja selektovanjem pojedinačnog elementa opšteg NLU sa njemu ekvivalentnim elementima metamodela specifičnog NLU servisa. Dakle, da bi bili nezavisni od samo jednog NLU eksterog servisa, potrebno je razviti sopstveni NLU metamodel pod nazivom DORIUS metamodel. Deo metamodela je prikazan na slici 29.



Slika 29. DORIUS Metamodel

Osnovni razlog razvoja sopstvene verzije metamodela je bila potreba da se omogući lakša korespondencija između koncepata najpoznatijih NLU platformi. U ovoj tezi daju se dva veoma popularna NLU servisa (*Dialogflow* i *RASA*). Zatim se dalje pokazuje kako je DORIUS NLU metamodel mapiran na metamodele *Dialogflow* i *RASA*. „Definisana preslikavanja moraju da prate pravila i ograničenja, koja su definisana metamodelom mapiranja“ [80]. Isečak *Dialogflow* i *RASA* metamodela prikazan je na slikama 30 i 32. Ovo su takođe originalni metamodeli koje je autor samostalno razvio. Verzije *Dialogflow* i *RASA* koje su se koriste u ovom pristupu za potrebe razvoja njihovih metamodela su zvanično opisane u [66, 67]. Svi ovi metamodeli i odgovarajuća pravila mapiranja su definisani u bazi podataka u vreme projektovanja.

Dizajner unosi trenirane podatke u NLU koncepte DORIUS metamodela za svaku poruku (frazu, iskaz, rečenicu). NLU trenirani podaci se sastoje od primera korisničkih poruka kategorizovanih po nameri. Kada klijenti četbota odluče koje NLU servise žele da koriste, odgovarajući NLU metamodel se automatski učitava iz DORIUS metamodela. Koristeći pravila mapiranja, automatski se kreiraju, održavaju i prosleđuju objekti sa svim potrebnim treniranim podacima odgovarajućim NLU metamodelima. Nakon toga se

puštaju trenirani podaci za učenje četbota. Rezultat ovih akcija su NLU servisi koji su obučeni prirodnim korisničkim frazama označeni entitetima. U vreme izvršavanja, NLU servisi parsiraju poruku (frazu, iskaz, rečenicu) na osnovu tehnika mašinskog učenja, upoređuje i proveravaju da li se podudara. NLU servisi vraćaju nameru i entitete nazad ka AI-NLP-NLU komponentu sa procentom tačnosti razumevanja namere. Kao što je već rečeno, platforma obezbeđuje da se može koristiti više NLU servisa, ali komponenta *Core Engine* bira onu koja ima veću verovatnoću tačnosti.

Koristeći ove metamodele predložene u ovom pristupu, takođe je lako uključiti nove NLU servise. Sve što treba da se uradi je da se kreira metamodel za taj novi NLU servis i odgovarajuća pravila mapiranja u DORIUS metamodel, koji će automatski proširiti izbor NLU servisa [81]. Naravno, postoji mogućnost proširenja DORIUS metamodela u slučaju da novi NLU servis ima neke nove karakteristike (koncepte) koje do sada nisu bili prepoznati, ali opet se tu nakon proširenja DORIUS metamodela radi mapiranje koje omogućava nezavisnost od NLU platforme.

Dodatna prednost DORIUS metamodel je ta da on može pomoći „starom“ četbotu da pređe na drugi NLU servis. Pod pretpostavkom da postoji četbot koji koristi samo *Dialogflow* servis, korišćenjem DORIUS metamodela moguće je lako prebaciti četbot sa *Dialogflow* na *RASA* servis i obrnuto. U ovom primeru, DORIUS koncepti se učitavaju iz *Dialogflow* metamodela pomoću pravila mapiranja, a zatim se njegovi koncepti mapiraju na *RASA* metamodel.

DORIUS NLU koncept predstavlja najapstraktniji koncept u modelu podataka NLU. Specijalizovan je koristeći NLU koncepte koji su konkretniji:

- *Intent* kategoriše nameru korisnika u jednom razgovoru. Namera je grupa fraza sa sličnim značenjem. Namera se odnosi na cilj koji korisnik ima na umu kada kuca pitanje ili komentar. Atribut *IntentId* predstavlja jedinstveni identifikator namere. Atribut *Name* je naziv namere.

Na primer, ako je ime namere *ProfessorConsultation*, *IntentId* bi mogao da bude na primer *Guid123*.

- *EntityType* predstavlja tip entiteta. Atribut *Name* može biti sistemski ili prilagođen (engl. custom).

Na primer, *EntityType* može biti prilagođen.

- *Entity* predstavlja unos entiteta za odgovarajući *EntityType*. Entitet može biti sistemski ili prilagođeni *EntityType*. Entiteti su označeni u primerima treninga sa imenom (engl. Name) entiteta. Entiteti su mehanizam za identifikaciju i izdvajanje korisnih podataka iz poruka (frazu, iskaz, rečenicu) na prirodnom jeziku. Dok namere omogućavaju razumevanje motivacije iza određenog korisničkog unosa, entiteti su strukturirani delovi informacija koji se mogu izdvojiti iz tih fraza. Pored naziva entiteta, entitet možemo označiti vrednostima i sinonimima.

TeacherName može biti primer entiteta koji je prilagođeni *EntityType*. *EntityID* može biti na primer *3en-22t*.

- *EntityValue* predstavlja primarnu vrednost povezanu sa unosom entiteta. Entitet može imati jednu ili više vrednosti (engl. Value).

Na primer, ako je entitet *TeacherName*, *EntityValue* može biti: *Rade Mati* , *Miloš Kabiljo* itd.

- *EntitySynonym* predstavlja sinonim za *EntityValue*. Sinonimi se mogu koristiti kada postoji više načina na koje korisnici upućuju na istu stvar. *EntityValue* može da ima jedan ili više *EntitySynonym*.

Na primer, ako je *EntityValue* - *Rade Mati* , sinonimi entiteta mogu biti: *R.M.*, *Radetom Mati em*, *Radu Mati u*.

- *Utterance* (srb. fraza, poruka, upit, rečenica) je primer (trening) fraze za ono što korisnici mogu da otkucaju ili kažu. Za svaku nameru možemo imati više fraza. Kada korisnička poruka izgleda (odgovara) jednoj od ovih fraza, NLU pronalazi nameru.

Primeri fraza mogu biti:

- Kada profesor Rade Matić ima konsultacije?
- Kada mogu da se konsultujem sa profesorom Radetom Matićem?
- Kada mogu da posetim profesora R.M.?
- *UtteranceEntityValue* predstavlja vrednost entiteta u jednom ili više fraza. Fraza može imati nula ili više vrednosti entiteta. Početne (engl. start) i krajnje (engl.

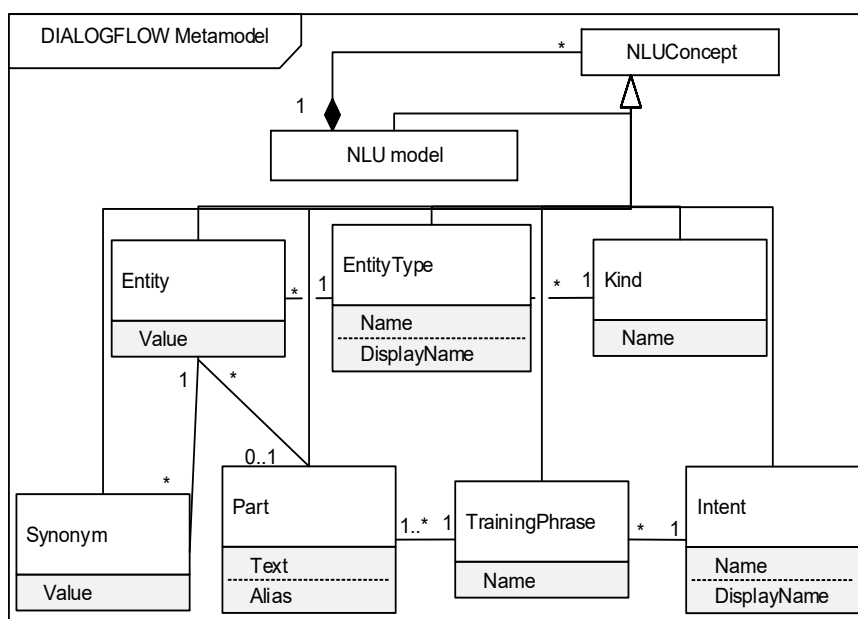
end) pozicije atributa su važne jer tako model zna koje znakove treba izdvojiti i koristiti za trening modela. *PartText* predstavlja uređenu (sortiranu) listu delova fraze. Povezani delovi *PartText* formiraju frazu. Pomoću *RoleName* se mogu definisati entiteti sa specifičnim ulogama u frazi. *GroupName* omogućava da se entiteti grupišu zajedno sa određenom oznakom grupe. Oznaka grupe se može koristiti za definisanje različitih sortiranja.

Na primer, u frazi: „Kada profesor Rade Matić ima konsultacije?“, vrednost entiteta „Rade Matić“ je *PartText* koji počinje sa pozicije 15 i završava se na poziciji 25.

- *Culture* pruža mogućnost višejezične podrške. Fraza koja je povezana sa nekom namerom se može definisati za svaki jezik pojedinačno, kao i entiteti koji se koriste u njima. Svaka kultura može imati svoj skup fraza i entiteta.

Metamodel *Dialogflow* je dat na slici 30. NLU koncept je dalje specijalizovan na koncepte koji su konkretniji:

- Intent
- Kind
- EntityType
- Entity
- Synonym
- TrainingPhrase
- Part



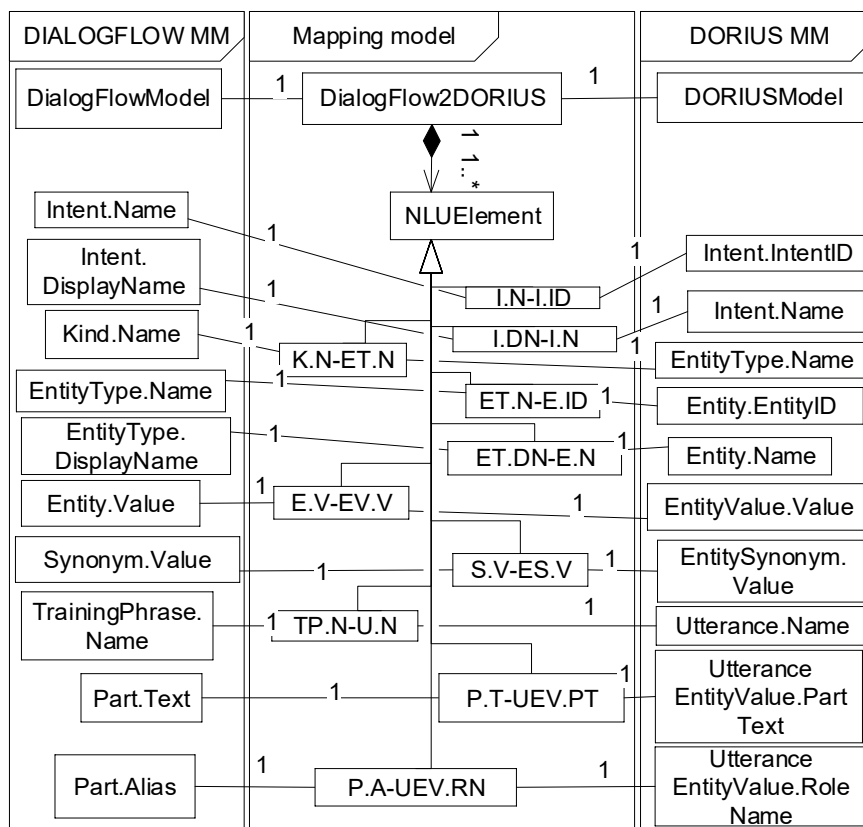
Slika 30. Dialogflow metamodel

Mapiranje između koncepata DORIUS NLU i *Dialogflow* NLU je determinisano sledećim pravilima:

- Dialogflow2DORIUS pravilo: Svaki model Dialogflow-a se mapira na model DORIUS-a.
- I.N-I.ID pravilo: Svaki Intent.Name se mapira na Intent.IntentID.
- I.DN-I.N pravilo: Svaki Intent.DisplayName se mapira na Intent.Name.
- K.N-ET.N pravilo: Svaki Kind.Name se mapira na EntityType.Name.
- ET.N-E.ID pravilo: Svaki EntityType.Name se mapira na Entity.EntityID.
- ET.DN-E.N pravilo: Svaki EntityType.DisplayName se mapira na Entity.Name.
- E.V-EV.V pravilo: Svaki Entity.Value se mapira na EntityValue.Value.
- S.V-ES.V pravilo: Svaki Synonym.Value se mapira na EntitySynonym.Value.
- TP.N-U.N pravilo: Svaki TrainingPhrase.Name se mapira na Utterance.Name.
- P.T-UEV.PT pravilo: Svaki Part.Text se mapira na UtteranceEntityValue.PartText.
- P.A-UEV.RN pravilo: Svaki Part.Alias se mapira na UtteranceEntityValue.RoleName.

Na slici 31. se vidi model preslikavanja koji daje detaljan prikaz dozvoljenih korespondencija između koncepata *Dialogflow* NLU i DORIUS NLU prateći definisana pravila. Mapiranje između dva konkretna modela *Dialogflow* i DORIUS je predstavljeno klasom Dialogflow2DORIUS. Ona obuhvata sva mapiranja između konkretnih elemenata *Dialogflow* i DORIUS modela nastalih korišćenjem pravila mapiranja. Klasa NLUElement je instanca takvih korespondencija. Za svako pravilo mapiranja postoje odgovarajuće podklase NLUElement, koje su nazvane po pravilu. Jasno se vidi iz metamodela da su mapiranja između koncepata modela *Dialogflow* i DORIUS-a jedinstvena i nedvosmislena.

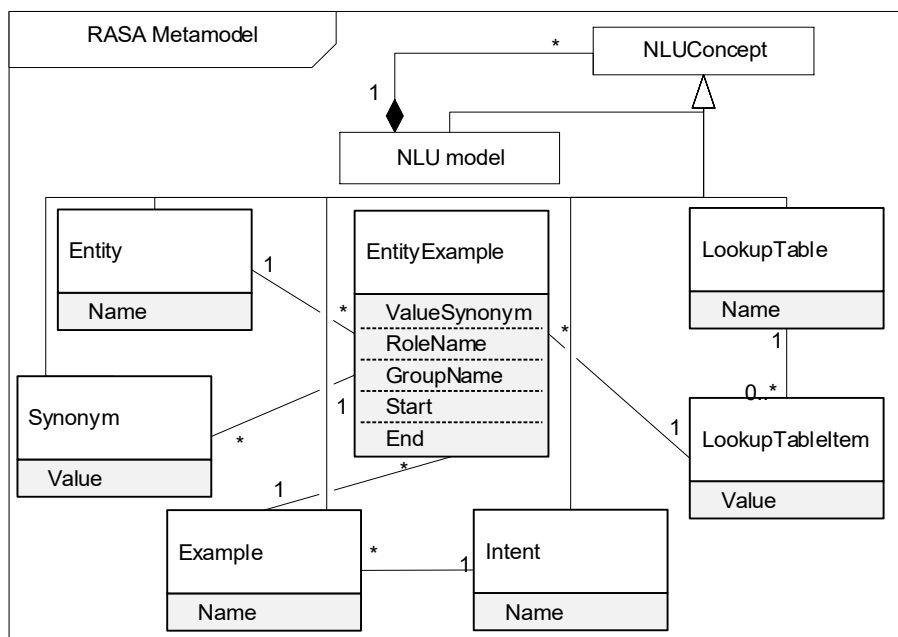
Sledeća slika daje Dialogflow2DORIUS pravila mapiranja svih koncepata.



Slika 31. Model preslikavanja Dialogflow2DORIOUS

Isečak RASA metamodel sa najapstraktnijim RASA NLU konceptom je dat na slici 32. On se dalje specijalizuje na konkretnije NLU koncepte:

- Intent
- Entity
- EntitySynonym
- Example
- EntityExample
- LookupTable
- LookupTableItem



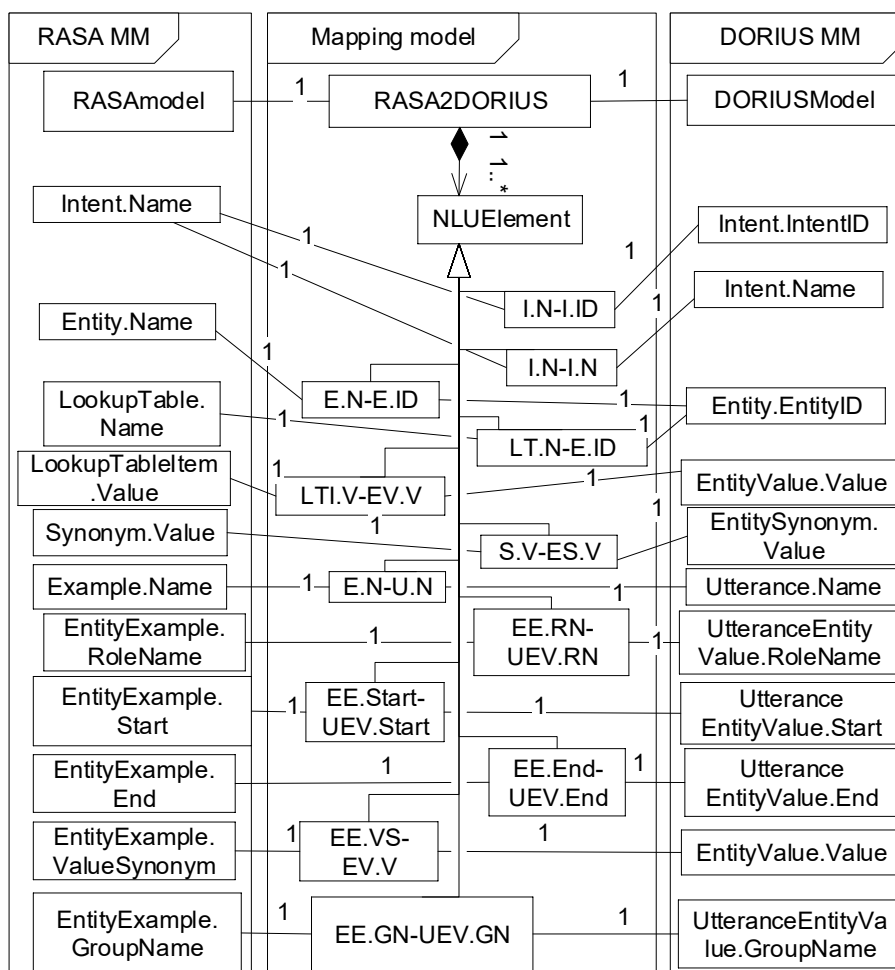
Slika 32. RASA metamodel

Mapiranje između RASA NLU i DORIUS NLU koncepata se determinisan sa sledećim pravilima:

- RASA2DORIUS pravilo: Svaki RASA model se mapira na DORIUS model.
- I.N-I.ID pravilo: Svaki Intent.Name model se mapira na Intent.IntentID.
- I.N-I.N pravilo: Svaki Intent.Name model se mapira na Intent.Name.
- E.N-E.N pravilo: Svaki Entity.Name model se mapira na Entity.EntityID.
- LT.N-E.N pravilo: Svaki LookupTable.Name model se mapira na Entity.EntityID.
- LTI.V-EV.V pravilo: Svaki LookupTableItem.Value model se mapira na EntityValue.Value.
- S.V-ES.V pravilo: Svaki Synonym.Value model se mapira na EntitySynonym.Value.
- E.N-U.N pravilo: Svaki Example.Name model se mapira na Utterance.Name.
- EE.VS-EV.V pravilo: Svaki EntityExample.ValueSynonym model se mapira na EntityValue.Value.
- EE.Start-UEV.Start pravilo: Svaki EntityExample.Start model se mapira na UtteranceEntityValue.Start.
- EE.End-UEV.End pravilo: Svaki EntityExample.End model se mapira na UtteranceEntityValue.End.
- EE.RN-UEV.RN pravilo: Svaki EntityExample.RoleName model se mapira na UtteranceEntityValue.RoleName.

- EE.GN-UEV.GN pravilo: Svaki EntityExample.GroupName model se mapira na UtteranceEntityValue.GroupName.

Slično kao sa slike 31, slika 33 prikazuje model preslikavanja koji daje detaljan prikaz dozvoljenih korespondencija između koncepata RASA NLU i DORIUS NLU prateći definisana pravila. Mapiranje između dva konkretna modela RASA i DORIUS je predstavljeno klasom RASA2DORIUS. Ona obuhvata sva mapiranja između konkretnih elemenata RASA i DORIUS modela nastalih korišćenjem pravila mapiranja. Klasa NLUElement je instanca takvih korespondencija. Za svako pravilo mapiranja postoje odgovarajuće podklase NLUElement, koje su nazvane po pravilu. Jasno se vidi iz metamodela da su mapiranja između koncepata modela RASA i DORIUS-a jedinstvena i nedvosmislena. Sledeća slika daje RASA2DORIUS pravila mapiranja svih koncepata.



Slika 33. Model preslikavanja RASA2DORIUS

Iz metamodela se opet jasno uviđa da su mapiranja između koncepata modela *RASA* i *DORIUS* jedinstvena i nedvosmislena. Svaki elemenat specifičnog NLU metamodela mora da ima svoj odgovarajući jedan elemenat u opštem *DORIUS* metamodelu.

5.5. DEFINISANJE KONVERZACIONOG DIZAJNERA ZA MODELOVANJE KONVERZACIONIH TOKOVA

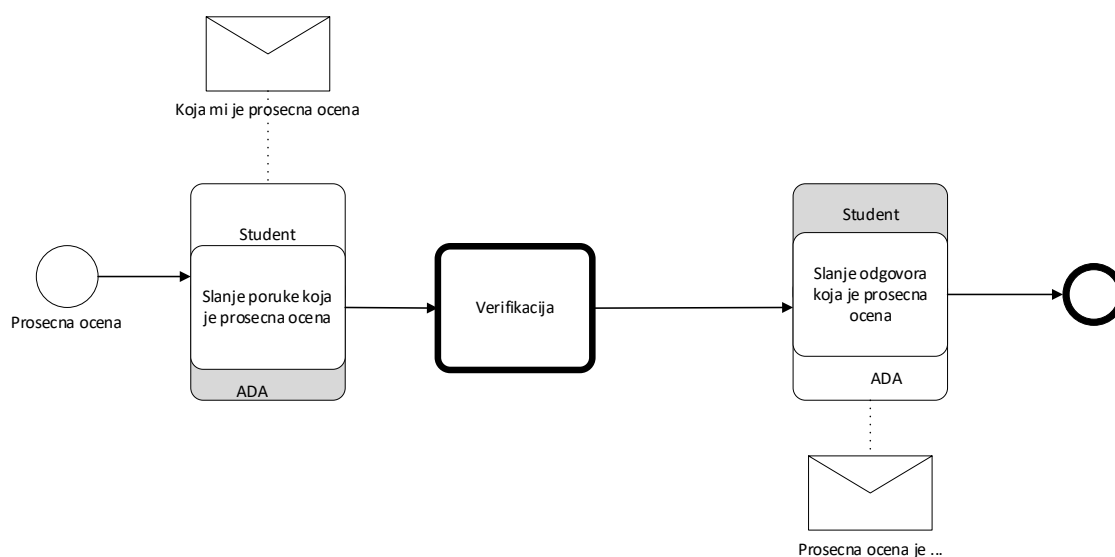
Potrebno je obezbediti konverzacioni dizajner koji pruža mogućnost modelovanja konverzacionog toka svakog slučaja korišćenja (scenarija). Konverzacioni dizajner treba da predstavlja vizuelni alat opremljen kompletom za razvoj softvera za odgovarajuću platformu za implementaciju. Kroz ovaj alat je moguće direktno podučavati NLU da bi se izbegla zavisnost od NLU interfejsa dobavljača. Takođe, dizajner omogućava vizuelno generisanje scenarija, odnosno njegovog toka kao i svih odgovora. Tok razgovora mapira sve potencijalne pravce u kojima se diskusija može razvijati, sa mnogo grana za sve mogućnosti. Tok razgovora je odgovoran za predviđanje svih mogućih ulaznih poruka i reakcija četбота. Ovo pomaže korisnicima da brzo ostvare svoje ciljeve i lakše dođu do informacija kao i da daju prioritet poslovnim potrebama. Dizajner treba da je intuitivan i jednostavan za razumevanje i korišćenje.

Dizajner takođe treba da obezbedi mogućnost modelovanja bilo koje vrste integracije sa eksternim servisima (API, baze podataka i sl) kao i modelovanja izuzetaka i tehničkih ograničenja u tim scenarijima. Neke od vrsta integracija su:

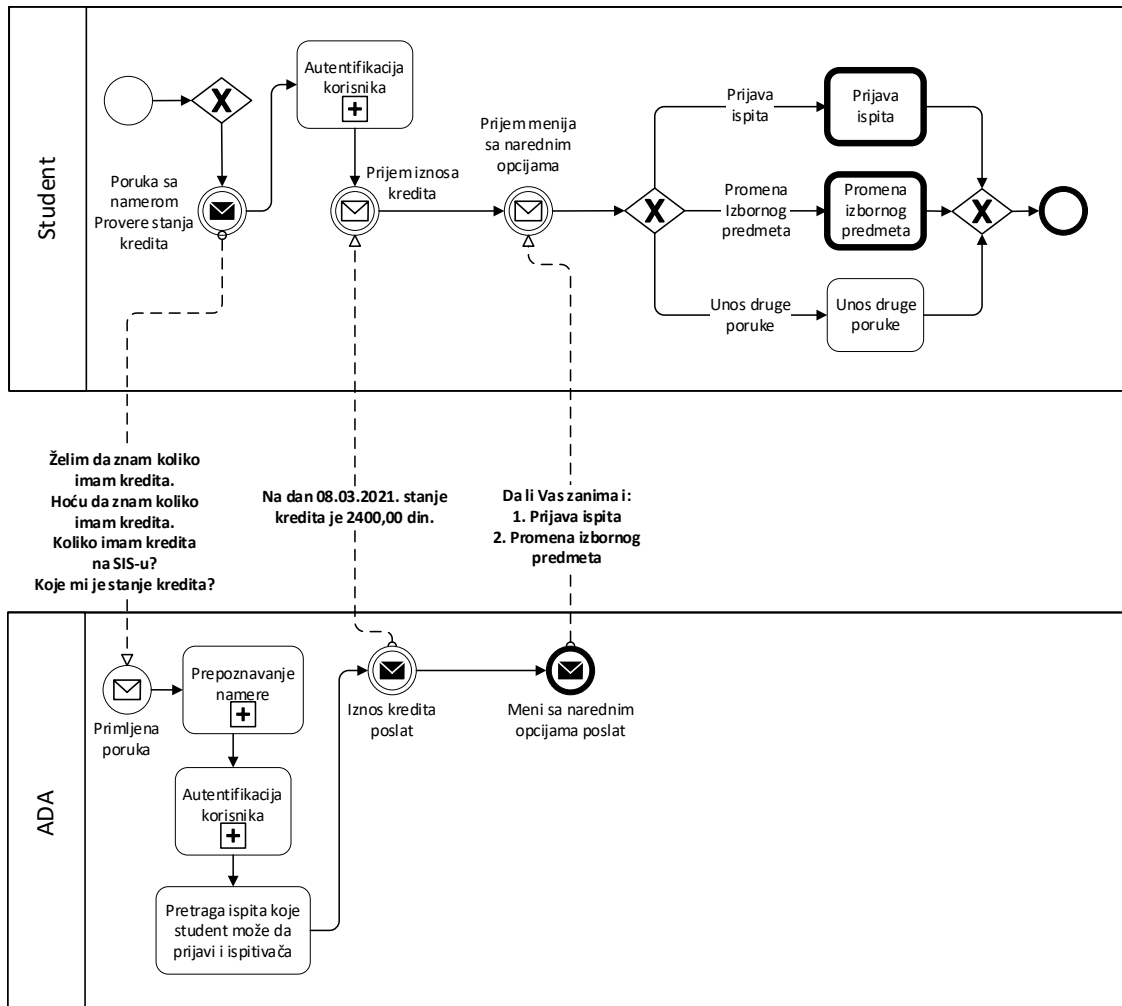
- CRM (npr. Hubspot, Zoho, Salesforce, i sl.);
- Kalendar (Google kalendar, i sl.);
- ERP sistemi (SAP i sl.);
- Sistemi plaćanja (PayPal, i sl.);
- Mape (Google Maps, Apple Maps);
- Alatkе za skladištenje u oblaku (Dropbox, Google Drive);
- Razne drugi biznis sistemi.

Za potrebe modelovanja toka konverzacije koriste se standardni jezici koji treba da budu jednostavni i razumljivi za poslovne korisnike, neinformatičare i analitičare. Pored toga oni moraju da podrže detalje opisa složenih poslovnih procesa i da budu formalizovani u cilju implemenacije istih. S obzirom da još uvek ne postoji standard za modelovanje konverzacionih tokova u implementaciji četbota potrebno je naći jedno novo rešenje. Modelovanje i notacija poslovnog procesa kroz opšte prihvaćeni standard BPMN (engl. Business Process Model and Notation) je najbitniji i najkorišćeniji jezik za modelovanje poslovnih procesa kao standard u menadžmentu poslovnih procesa. Model nacrtan koristeći BPMN standard se definiše preko dijagrama kolaboracije (engl. Collaboration diagram). On je baziran na blok-dijagramu toka i prilagođen je za kreiranje grafičkih modela operacija poslovnih procesa. Izgleda lako razumljiv za najveći deo poslovnih analitičara i menadžera. Ovaj standard ima dosta elemenata koji su dovoljni za modelovanje konverzacionog toka svakog scenarija. Zato se ovde predlaže da konverzacioni dizajner treba da podrži BPMN, jer se proces posmatra iz “ptičije perspektive“ da bi se videla komunikaciona interakcija između korisnika koji razmenjuju poruke.

Primeri dijagrama scenarija koje konverzacioni dizajner treba da obezbedi su dati na slici 34. i 35. Ove slika pokazuju primenu BPMN standarda i njegovih dijagrama. Na slici 34. i 35. su prikazani dijagram koreografije za dobijanje informacije o prosečnoj ocenu studenta, odnosno dijagram kolaboracije za potrebe dobijanja informacija o stanju kredita studenta. Sve ove scenarije je potrebno da se mogu pregledati, izmeniti i izbrisati.



Slika 34. BPMN dijagram koreografije za proveru prosečne ocene studenta



Slika 35. BPMN dijagram kolaboracije za proveru stanja kredita

5.6. RAZVOJ ČETBOT KOMPONENTI

Pod aktivnošću razvoja četbot komponenti smatra se kreiranje opštih modela i komponenti za sve potencijalne tipove četbota koristeći standardne paterne u izabranom implementacionom okruženju. Nakon razvoja neophodno je njihovo testiranje. Sve ovo treba da popuni ponovo koristiva sredstva adaptivne platforme za razvoj četbota. Opisana

arhitektura i implementacija četbota kao što je već rečeno treba da je zasnovana je na veb API servisnoj arhitekturi i mikroservisima. Ova jednostavna arhitektura je izabrana jer omogućava RESTfull servis i JSON objekte, a može da je koriste klijenti koji razumeju XML ili JSON. Takođe, veb API može biti hostovan unutar IIS ili same aplikacije.

Prilikom razvoja i projektovanja arhitekture četbota korišćeno je nekoliko tehnika za skalabilnost. S obzirom da zahtevi stalno rastu a tehnologija napreduje, zahvaljujući mikroservisima arhitektura se može zameniti moćnijim i bržim komponentama. Izvođenje upita se izvršava asihrono i čuvaju se u red (engl. queue). Asinhrona obrada uklanja neka od uskih grla koja utiču na performanse. Skalabilnost je takođe povećana izborom *Angular*-a kao jedan od modernih okvira za aplikaciju sa jednom stranicom, poboljšavajući ukupne performanse.

Detaljna implementacija ovih komponenti nisu predmet teze, iako će pojedini detalji implementacije iz aplikacionog inženjeringa biti opisani u sledećem poglavlju.

6. APLIKACIONI INŽENJERING

Iako *Weaver* platforma ne podržava veći deo faza ovde predložene metodologije razvoja adaptivnog četbota odnosno projektnog inženjeringa, ona je uspešno iskorišćena za fazu aplikacionog inženjeringa, odnosno za potrebe dokazivanja NLU nezavisnosti od proizvođača. Pre opisa svih aktivnosti u fazi aplikacionog inženjeringa u nastavku će se dati detaljnije o *Weaver*-u.

Weaver platforma za četbot predstavlja *enterprise* rešenje iz oblasti softverskih proizvoda koji u sebi poseduju veštačku inteligenciju. Zasnovana je na razumevanju i obradi prirodnog jezika. Ovaj podskup veštačke inteligencije daje mogućnost analize unetih korisničkih poruka i fraza. Međutim, ova analiza se ne radi jednostavnom pretragom ključnih reči, već se radi sa potpunim razumevanjem prirodnog jezika i konteksta cele poruke. Platforma poseduje administraciju korisnika koji mogu da koriste alat, kao i monitorisanje i osnovne izveštaje vezane za kanale komunikacije. Administrativni alat poseduje vizuelni dizajner scenarija za definisanje istih koje četbot podržava. Tok konverzacije mapira sve potencijalne pravce u kojima se konverzacija može odvijati, uz mnoštvo različitih grana za sve mogućnosti. Administracija je vizuelizovana i uz trening je lako savladiva. Platforma je u mogućnosti da iz poruke otklanja greške, a sve to primenom najsavremenijih tehnoloških dostignuća iz oblasti veštačke inteligencije. Važno je napomenuti da se sama konverzacija između četbot i čoveka odvija interaktivno. Jedinstven mehanizam toka same konverzacije kalkuliše šta je sledeća akcija sa kojom treba da se obrati korisniku i zavisi od celog toka konverzacije. Platforma poseduje mogućnost integracije sa sistemima unutar i izvan organizacije čime sam četbot dobija dodatnu vrednost u smislu mogućnosti da korisniku pruži personalizovanu informaciju ili mogućnost da koristi transakcioni ili bilo koji drugi servis. Jednostavna poruka: “Želim da kupim knjigu Projektovanje IS” uz dodatne korake za autorizaciju je najjednostavniji način da se izvrši veoma kompleksna akcija. Četbot iz teksta razume šta je namera korisnika. Sve mogućnosti *Weaver* platforme korisnici mogu da koriste kroz sledeće kanale: *Viber*, *FB messenger*, *Instagram*, *WhatsApp*, *Web chat* (postojeći veb sajt) itd. Jednostavan način komunikacije i korišćenja servisa ruši barijere

između komplikovanog interfejsa aplikacija i ljudi kojima je servis neophodan. Sve prethodno napisano, postavlja *Weaver* platformu ravnopravnom sa svetskim liderima iz ove oblasti. *Weaver* u osnovi objedinjuje kanale komunikacije u jednu tačku, konkretno sve kanale koje je tehnički moguće integrisati. Komunikacija po svim kanalima se čuva u jednom repozitorijumu. Obrade, analize i izveštavanja mogu da se vrše na jednom mestu.

6.1. ANALIZA DOMENA KONKRETNOG ČETBOTA

U analizi domena konkretnog četbota treba da se analizira domen (oblast) kojim će se baviti konkretan četbot. Potrebno je da se analizira dokumentacija, što podrazumeva prikupljanje celokupne poslovne dokumentacije, uključujući poslovnu politiku i pravila. Zatim je neophodno razumevanje organizacione kulture, ciljeva poslovanja i strukture informacija. Ovo je sve obavezno uraditi koristeći sva raspoloživa sredstva koja su kreirana u fazi projektnog inženjeringa. Ovo podrazumeva:

- Postavljanje ciljeva četbota.
- Kreiranje detaljne mape puta za razvoj četbota i analiza zahteva projekta.
- Upoznavanje i definisanje problema domena i uloge četbota u njemu. Neki od primera domena su: obrazovanje, elektronska prodaja, zdravstvo, marketing itd.
- Održavanje radionica sa domenskim ekspertima u kojima sistem analitičar mora da razotkrije i specificira sve zadatke koji se prepoznaju u razgovoru sa domenskim ekspertom.
- Analizu zahteva kroz: poslovnu dokumentaciju koja se koristi (popunjeni i prazni zahtevi, ugovori, nalozi, rešenja i sl.), upitnike, intervju, upoređivanje, posmatranje, uzorkovanje, merenje, verifikaciju, itd.
- Identifikaciju i specifikaciju slučajeva korišćenja koje je moguće implementirati na opštem nivou razumevanja.

Beogradska akademija poslovnih i umetničkih strukovnih studija (BAPUSS) [82] je želela da iskoristi mogućnosti koje nudi napredna tehnologija u vidu digitalnog

asistenta zasnovanog na veštačkoj inteligenciji, sa ciljem da unapredi interakciju između studenata i računara, a samim tim i između korisnika njenih obrazovnih usluga i Akademije. Akademija je razvila ADA četbot [83] koristeći *Weaver* platformu kompanije SAGA kako bi svojim studentima pružili najbolju moguću uslugu tokom studija. Tako je Akademija postala prva obrazovna institucija u Srbiji u oblasti razvoja četbota.

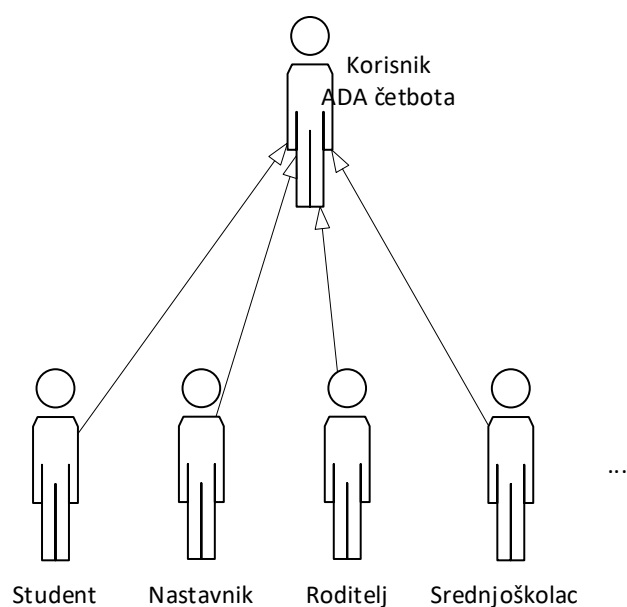
6.1.1. Akademska Digitalna Asistentkinja - ADA

U ovom delu rada se prikazuje jedan primer faze aplikacionog inženjeringa. Primer je opisan slučajevima korišćenja iz domena visokog obrazovanja.

Pandemija KOVID 19 ima široko rasprostranjene implikacije na sistem obrazovanja. Sa jedne strane, ona je ugrozila kontinuitet i kvalitet rada obrazovnih sistema, a sa druge strane, je otvorila vrata neophode digitalne transformacije obrazovanja i potrebu brže implementacije učenja na daljinu. Sposobnost zemalja da se brzo prilagode novim uslovima rada obrazovnih sistema nije ista i u velikoj meri zavisi od nivoa razvijenosti. Prema podacima UN, pandemija KOVID-19 stvorila je najveći poremećaj obrazovnih sistema u istoriji, imajući univerzalni uticaj na učenike i nastavnike širom sveta, od predškolskog obrazovanja, preko srednjih škola, tehničkog i stručnog obrazovanja i obuke, do visokih škola, univerziteta, ustanova za obrazovanje odraslih ili ustanova za razvoj veština. Jasno je prepoznata potreba kreiranja kvalitetnih digitalnih obrazovnih sadržaja, značaj IKT obrazovanja i osposobljavanja profesora i učitelja i iznad svega značaj dobre infrastrukture koja bi omogućila funkcionisanje ovako transformisanih obrazovnih sistema. Digitalne tehnologije nude potpuno nove odgovore na pitanje šta ljudi uče, kako uče, gde i kada uče. Tehnologija može da omogući nastavu i studentima pristup digitalnim materijalima, u različitim formatima kroz virtuelni svet povezivajući prostor i vreme. Da bi ostali konkurentni, posebno univerziteti, treba da osmisle svoje okruženje za učenje i informisanje tako da digitalizacija postane alat koji proširuje i dopunjuje odnos učenik-nastavnik, kao i da se u interaktivnoj komunikaciji studenti maksimalno uključuju u kreiranje sadržaja i u proces zajedničkog rada, učenja, istraživanja i saradnje. Što se tiče visokog obrazovanja prelazak na učenje na daljinu bio je prilika za proširivanje fleksibilnih modaliteta učenja, postavljajući osnovu za održivi prelazak na više onlajn učenja u ovom obrazovnom podsektoru u budućnosti.

Akadska Digitalna Asistentkinja (ADA) je prva implementacija četbota u prosveti na Balkanu koja kroz prirodnu komunikaciju pruža efikasniju realizaciju usluga i uštedu vremena. Realizaciju ovog projekta BAPUSS je započela vodeći se stavom da će četbot bitno uticati na poboljšanje, pre svega, informisanja i pružanja usluga od strane obrazovne ustanove, a zatim i unapređenja znanja studenata. Četbot verovatno neće rešiti sve probleme koje studenti trenutno imaju, ali može da predstavlja veoma koristan alat koji može da poveća efikasnost rada BAPUSS-a, digitalno doprinese i podrži realizaciju obrazovnog procesa.

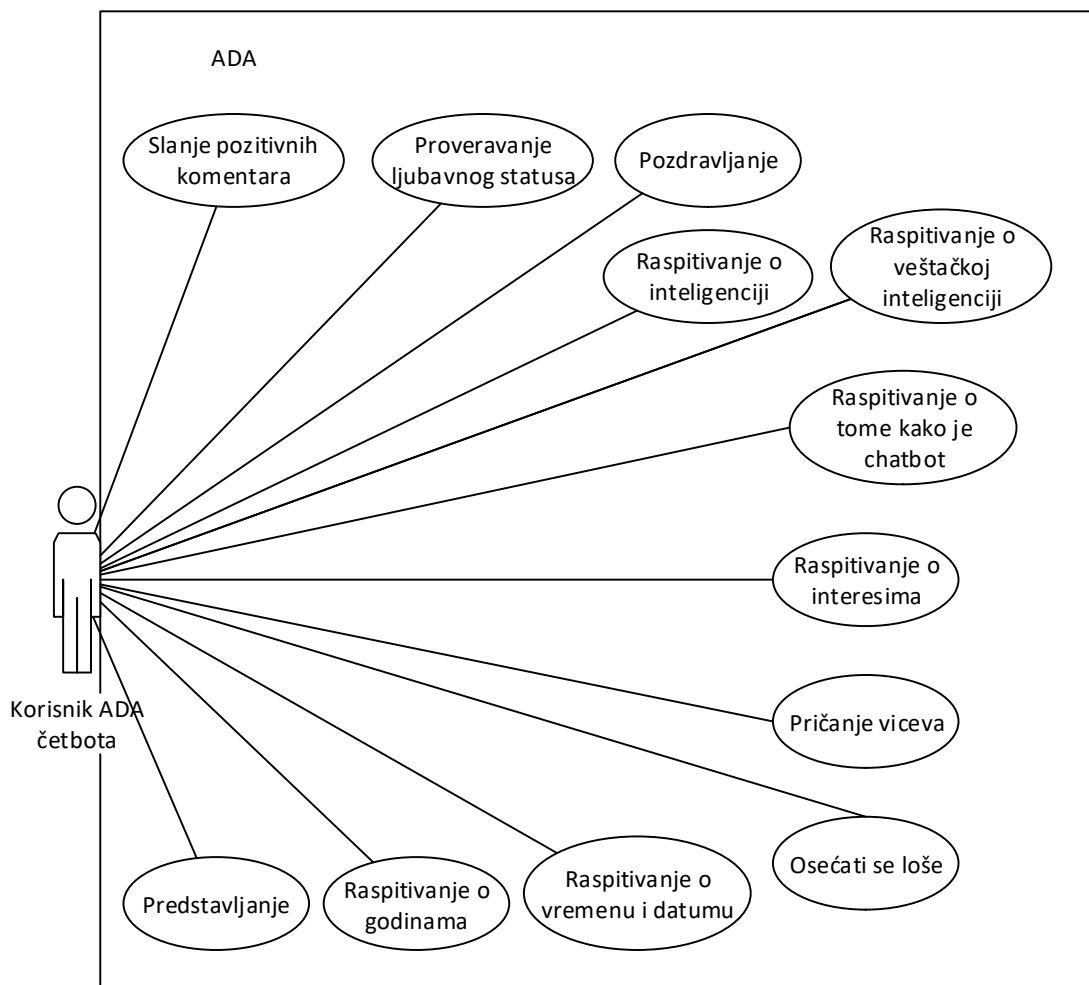
Radi jasnoće i boljeg razumevanja, u primeru se analiza domena konkretnog četbot ilustruje samo sa nekoliko dijagrama slučajeva korišćenja i dijagramima aktivnosti.



Slika 36. Generalizacija uloga korisnika u ADA četbotu



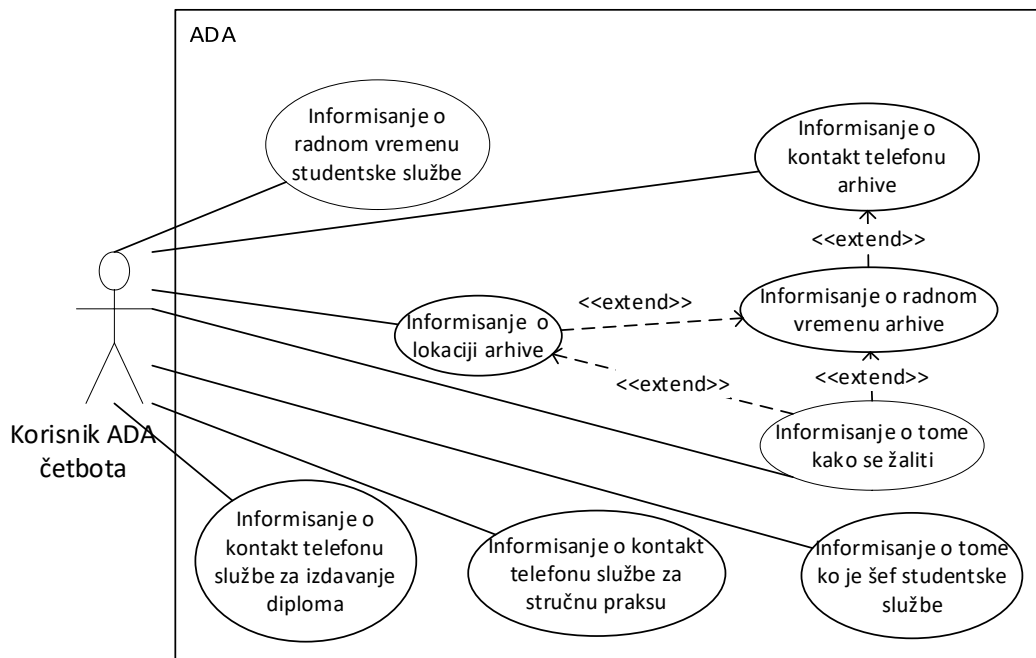
Slika 37. Dijagram slučajeva korišćenja za pružanje opštih informacija



Slika 38. Dijagram slučajeva korišćenja za zabavne scenarije



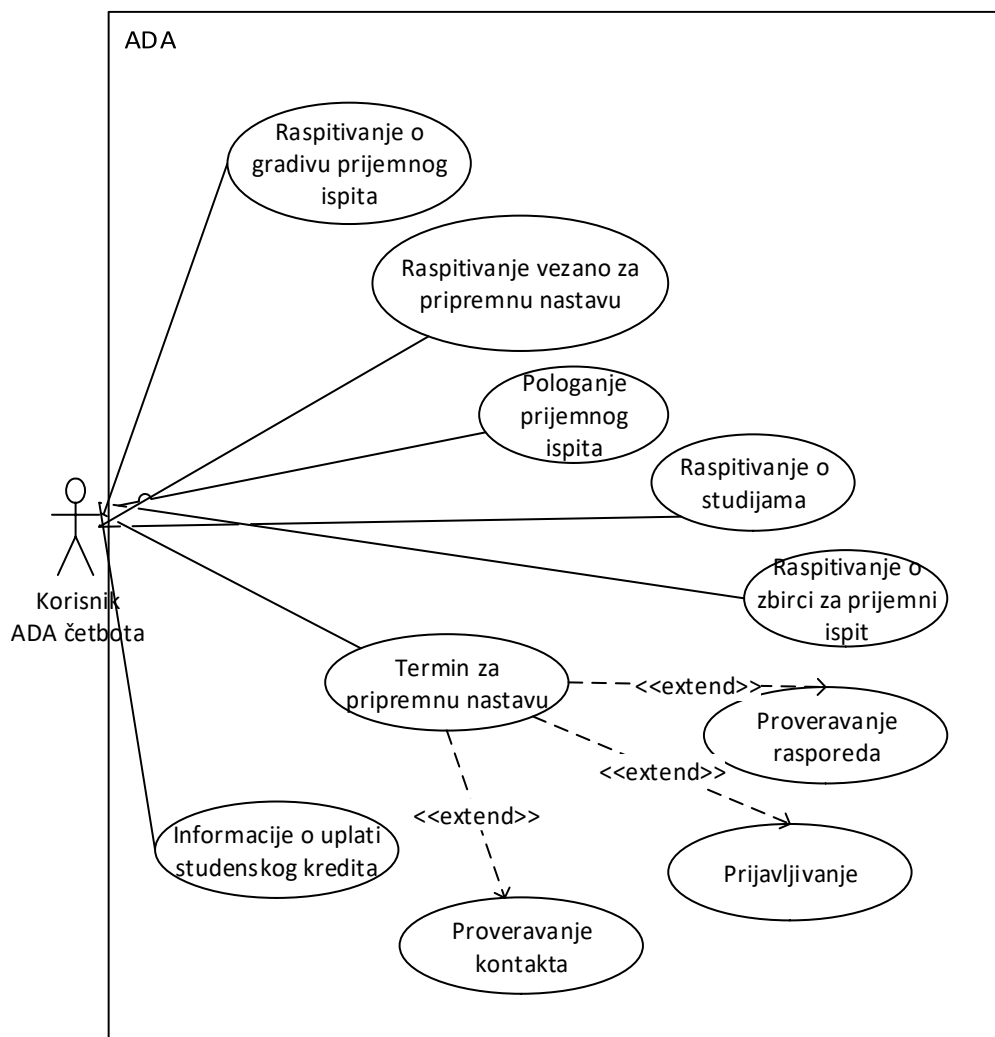
Slika 39. Dijagram slučajeva korišćenja za pružanje informacija o Akademiji



Slika 40. Dijagram slučajeva korišćenja za informacije o studentskom servisu



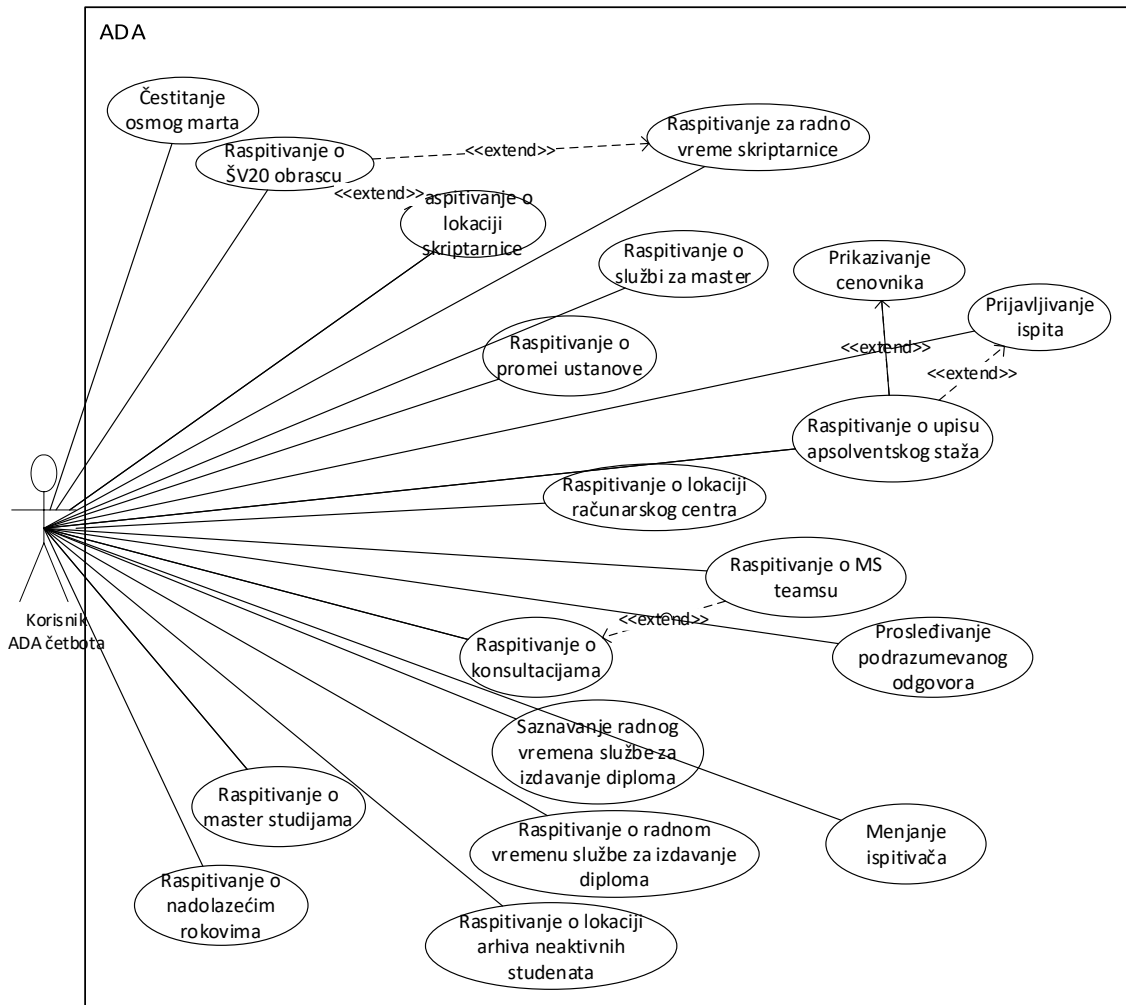
Slika 41. Dijagram slučajeva korišćenja za dodatne informacije o Akademiji



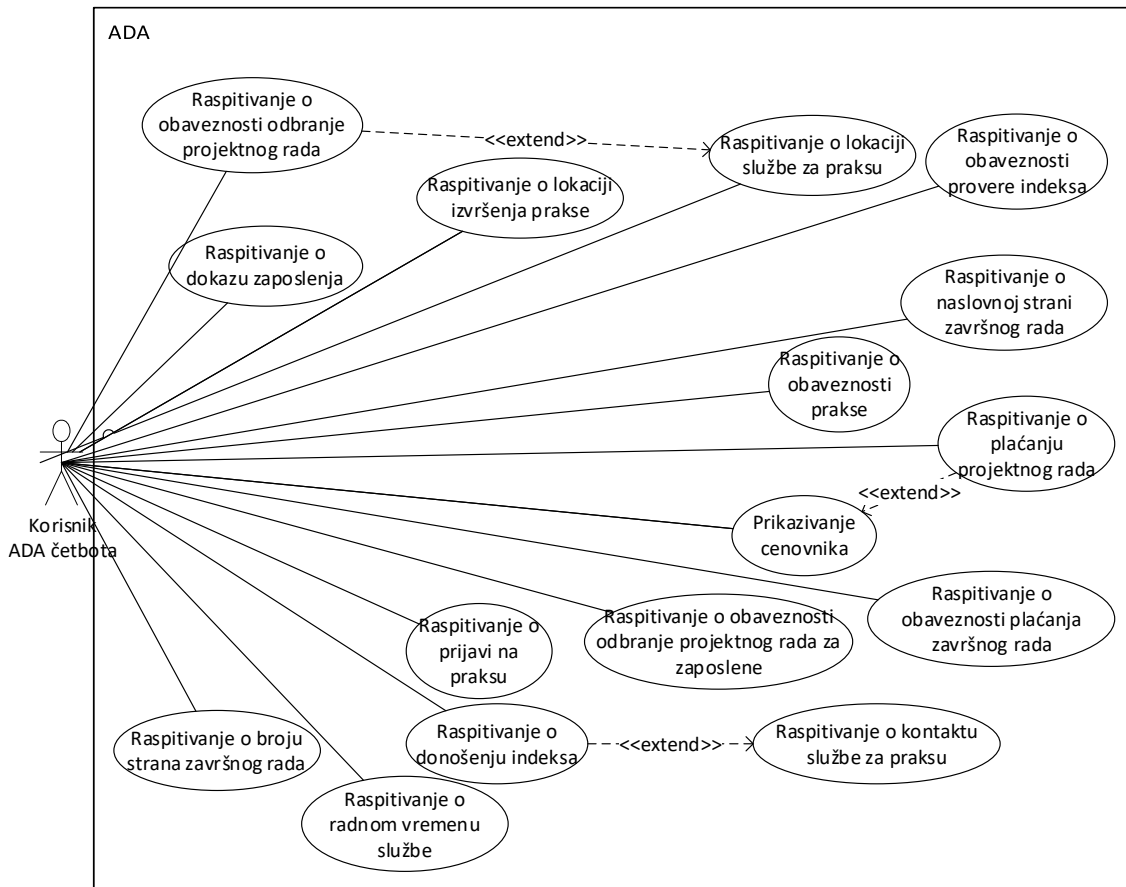
Slika 42. Dijagram slučajeva korišćenja za pripremu nastave i polaganje ispita



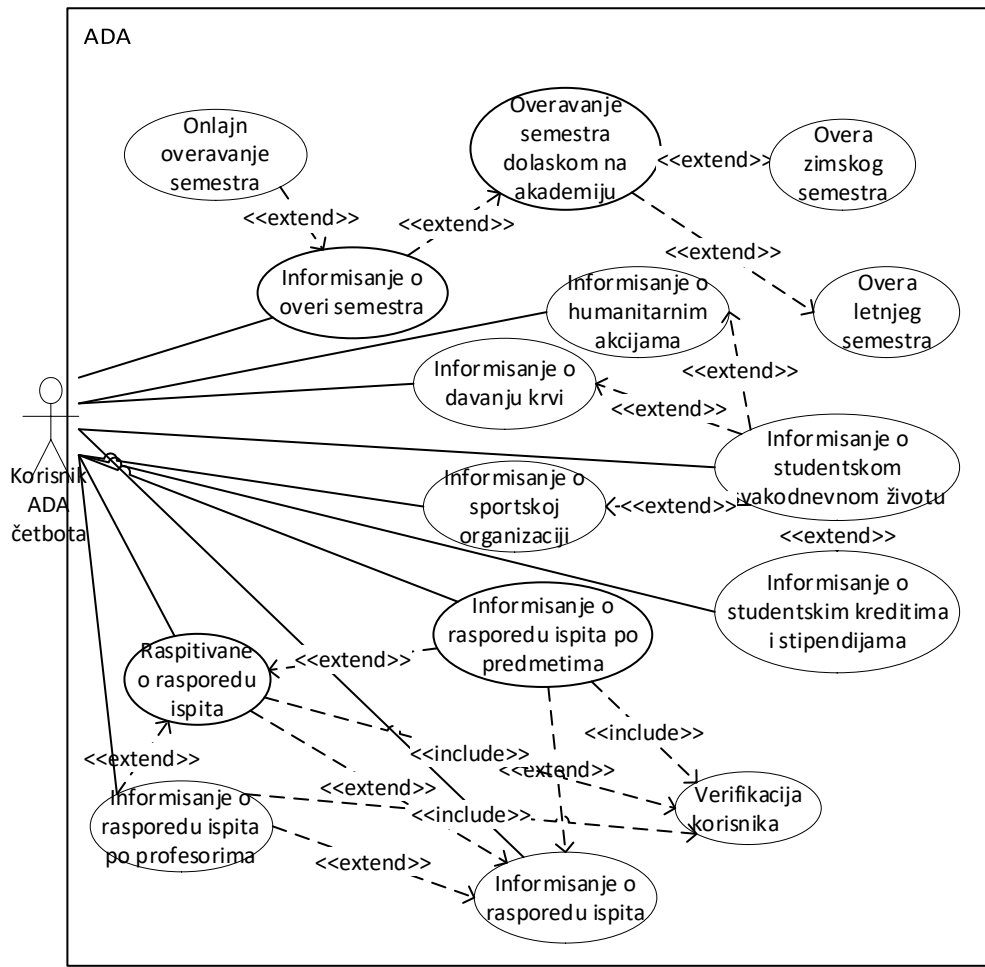
Slika 43. Dijagram slučajeva korišćenja za kontakte, stud. programe, rangiranje i potvrde



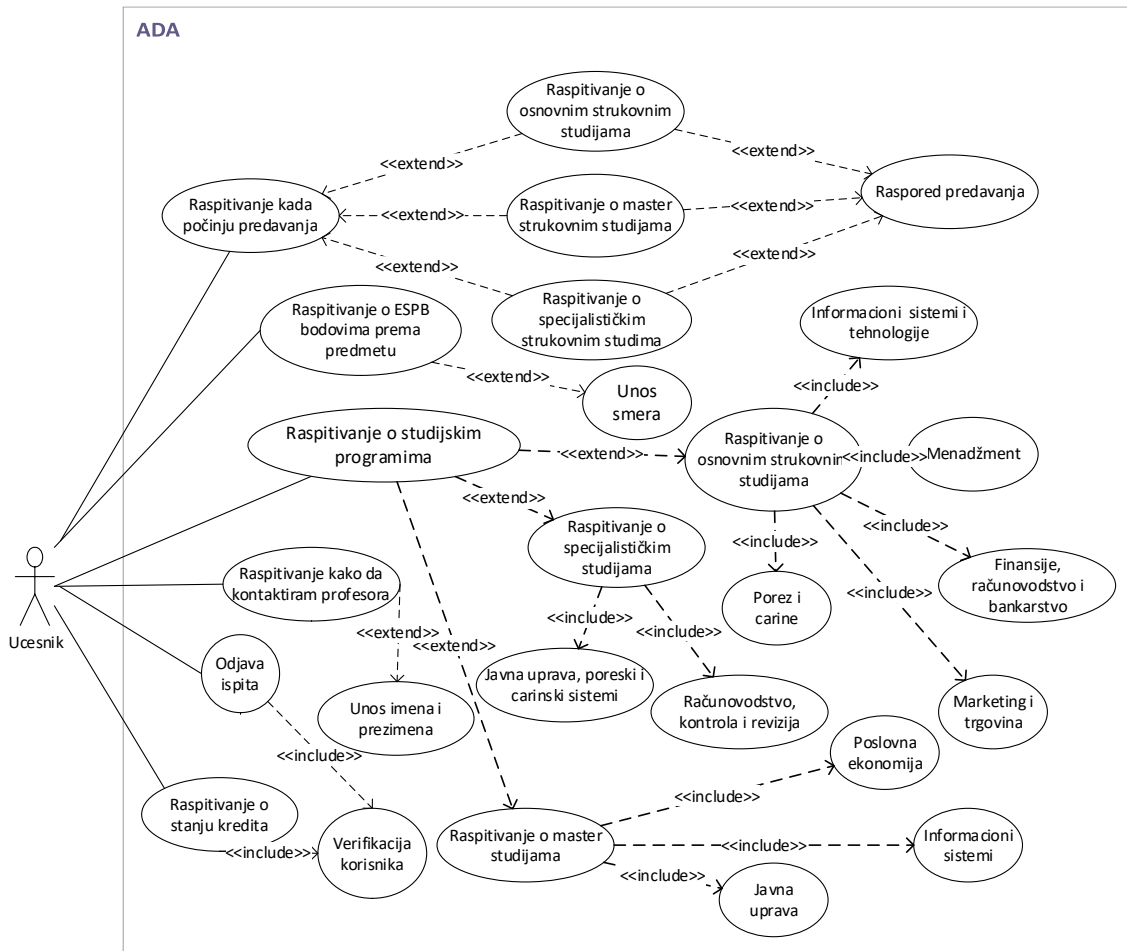
Slika 44. Dijagram slučajeva korišćenja za prijavu ispita, konsultacije i rokove



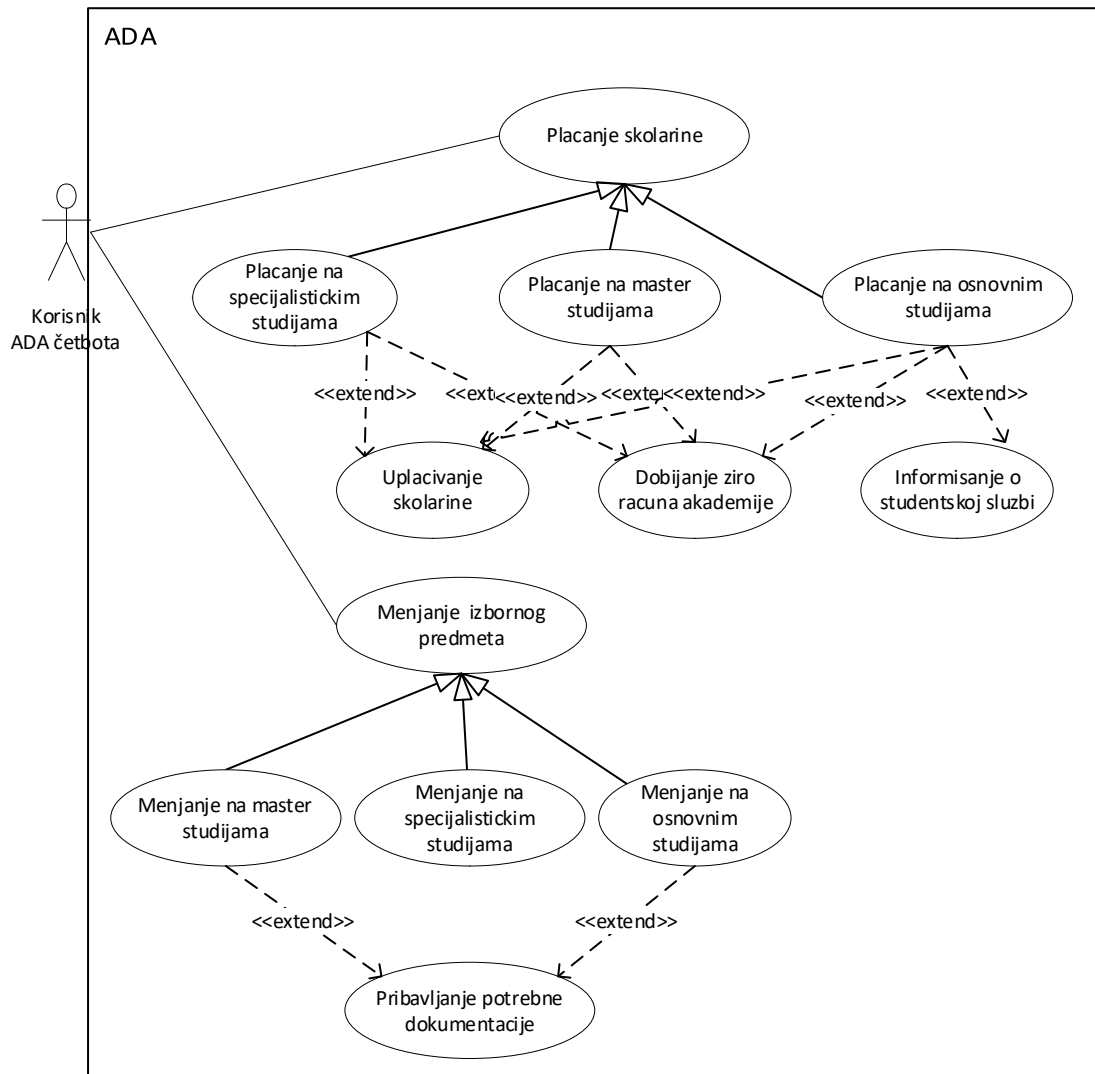
Slika 45. Dijagram slučajeva korišćenja za stručnu praksu



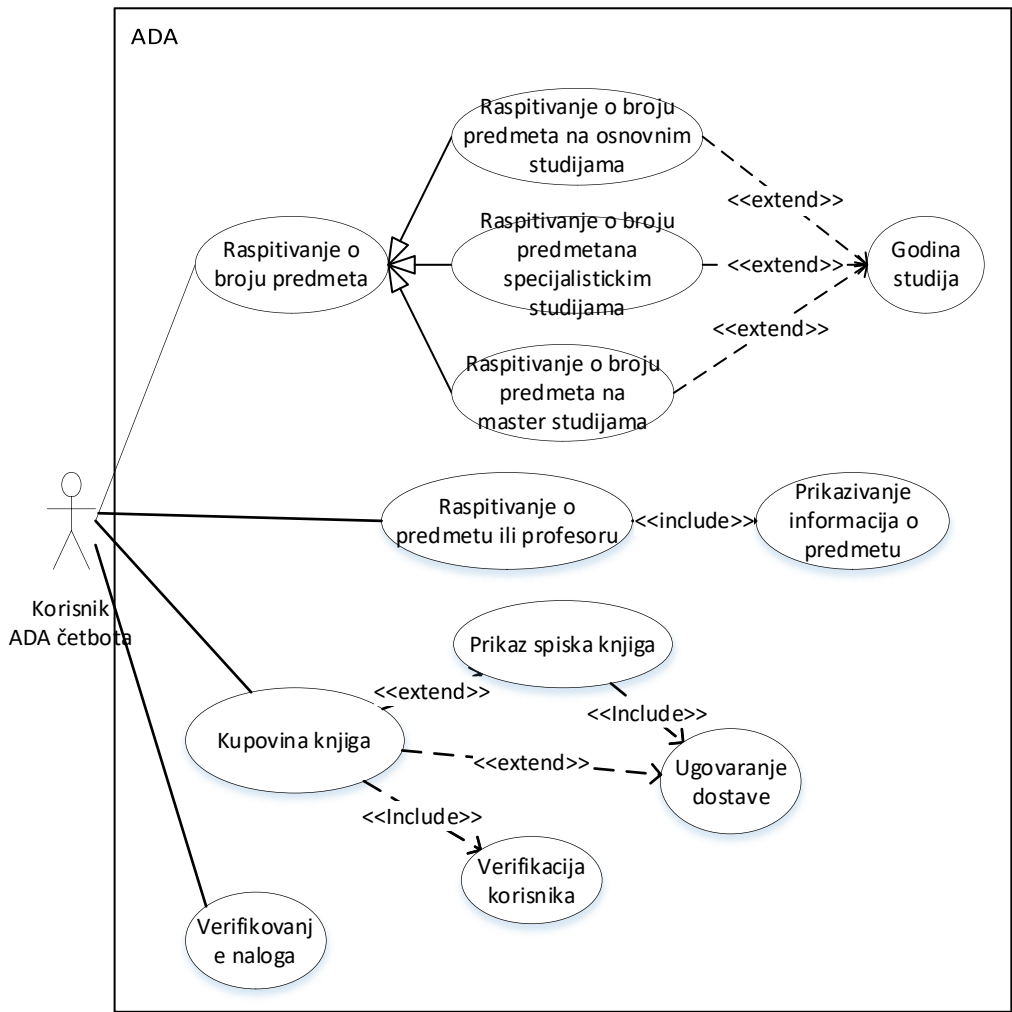
Slika 46. Dijagram slučajeva korišćenja za overu semestra i dodatne informacije



Slika 47. Dijagram slučajeva korišćenja za odjavu ispita, stanje kredita i predavanja



Slika 48. Dijagram slučajeva korišćenja za školarinu i izmenu izbornog



Slika 49. Dijagram slučajeve korišćenja za kupovinu knjiga i pregled predmeta

U nastavku se daje nekoliko primera opisa jednostavnih slučajeva korišćenja odnosno njihovih scenarija:

Tabela 2. Detaljan scenario „Kako si?“

Ime slučaja korišćenja	Kako si	
Cilj u kontekstu	Zabava	
Preduslovi	/	
Glavni akter	Učesnik	
Glavni tok	Koraci	Akcije
	1.	Akter šalje poruku: „Dobro sam“, “Osećam se odlično”
	2.	ADA prima poruku
	3.	ADA pronalazi odgovor
	4.	Šalje odgovor: „Sjajno, jako mi je drago da to čujem! :) Da li mogu da pomognem?“
Alternativni tok		
	4.1	Šalje odgovor: „Supeeeeer. Da li mogu pomoći nekako?“
	4.2	Šalje odgovor: “Drago mi je da to čujem! Da li mogu pomoći nekako?”
	4.3	Šalje odgovor: ”Drago mi je! Da li mogu pomoći nekako?”

Tabela 3. Detaljan scenario „Prijava ispita“

Ime slučaja korišćenja	Prijava ispita	
Cilj u kontekstu	Prijava ispita	
Preduslovi	Da je korisnik autentifikovan	
Glavni akter	Učesnik	
Glavni tok	Koraci	Akcije
	1.	Akter šalje poruku u kontekstu prijave ispita
	2.	ADA prima poruku
	3.	ADA proverava da li je korisnik autentifikovan
	4.	ADA proverava da li je ispitni rok aktivan
	5.	ADA šalje poruku sa informacijama o studentovom kreditu
	6.	ADA šalje poruku sa predmetima koje student može da prijavi
	7.	Student bira jedan od ponuđenih predmeta
	8.	ADA šalje poruku sa ispitivačima koji mogu ispitivati izabrani predmet
	9.	Student bira ispitivača za izabrani predmet

	10.	ADA šalje poruku sa detaljima o prijavi ispita i traži sa opcijama potvrde izbora (Da/Ne)
	11.	Student bira opciju Da
	12.	ADA određuje cenu prijave ispita
	13.	ADA vrši uvid u stanje kredita studenta
	14.	ADA šalje poruku o uspešno prijavljenom ispitu A kod ispitivača B.
	15.	ADA šalje poruku sa pitanjem da li želi da prijavi još jedan ispit.
Alternativni tok		
	4.1	ADA pokreće scenario za autentikaciju korisnika
	5.1	ADA šalje poruku „Prijava ispita nije u toku“
	11.1	Student bira opciju Ne
	11.2	ADA šalje poruku sa pitanjem da li želi da prijavi još jedan ispit.
	14.1	ADA šalje poruku da student nema dovoljno kredita

Tabela 4. Detaljan scenario „Srećna nova godina“

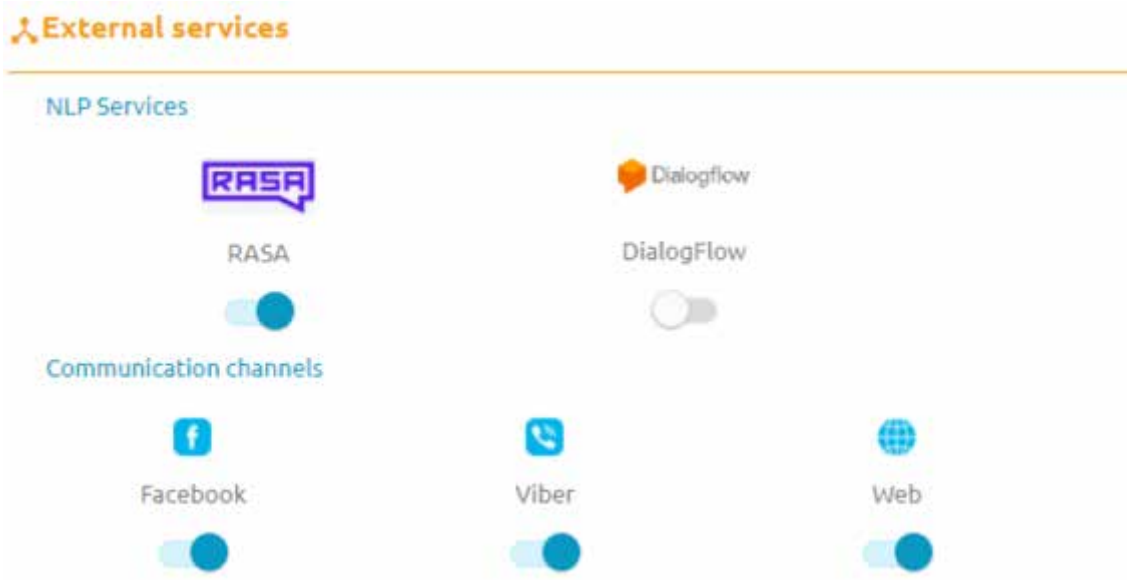
Ime slučaja korišćenja	Dijalog Nova Godina	
Cilj u kontekstu	Zabava	
Preduslovi	/	
Glavni akter	Učesnik	
Glavni tok	Koraci	Akcije
	1.	Akter šalje poruku: „Srećna Nova godina!“
	2.	ADA prima poruku
	3.	ADA pronalazi odgovor
	4.	Šalje odgovor: „Srećna Nova godina i da položiš sve ispite! :)“
Alternativni tok		
	4.1	Šalje odgovor: „ Srećna Nova godina!
	4.2	Šalje odgovor: “Srećna nova godina i tebi! Želim da ti se ispune sve želje! :)”

Tabela 5. Detaljan scenario „Šta radiš?“

Ime slučaja korišćenja	Dijalog Šta radiš	
Cilj u kontekstu	Zabava	
Preduslovi	/	
Glavni akter	Učesnik	
Glavni tok	Koraci	Akcije
	1.	Akter šalje poruku: „Šta radiš?“
	2.	ADA prima poruku
	3.	ADA pronalazi odgovor
	4.	Šalje odgovor: „Čekam da mi studenti postavе nova pitanja! Kako mogu da pomognem?“
Alternativni tok		
	4.1	Šalje odgovor: „Ništa specijalno, smišljam šta bih mogla sledeće da učim. Da li mogu da Vam dam neke informacije?“
	4.2	Šalje odgovor: “Razmišljam na temu Veštačke inteligencije u obrazovanju. Kako mogu da pomognem?”
	4.3	Šalje odgovor: “Učim da odgovaram na nova pitanja studenata. Kako mogu da pomognem?”

6.2. IZBOR I SETOVANJE NLU SERVISA I KOMUNIKACIONIH KANALA

U toku ove aktivnosti bira se NLU servis, jezik i komunikacioni kanali. Neophodno je setovati konkretne NLU service, komunikacione kanale i generisati njihovu nezavisnost. U slučaju da ne postoje željeni NLU servisi ili komunikacioni kanali, oni se dodatno prave u fazi projektnog inženjeringa. Slika 50 prikazuje izgled *Weaver* stranice za izbor NLU platforme i komunikacionih kanala.



Slika 50. Izbor NLP servisa i komunikacionih kanala

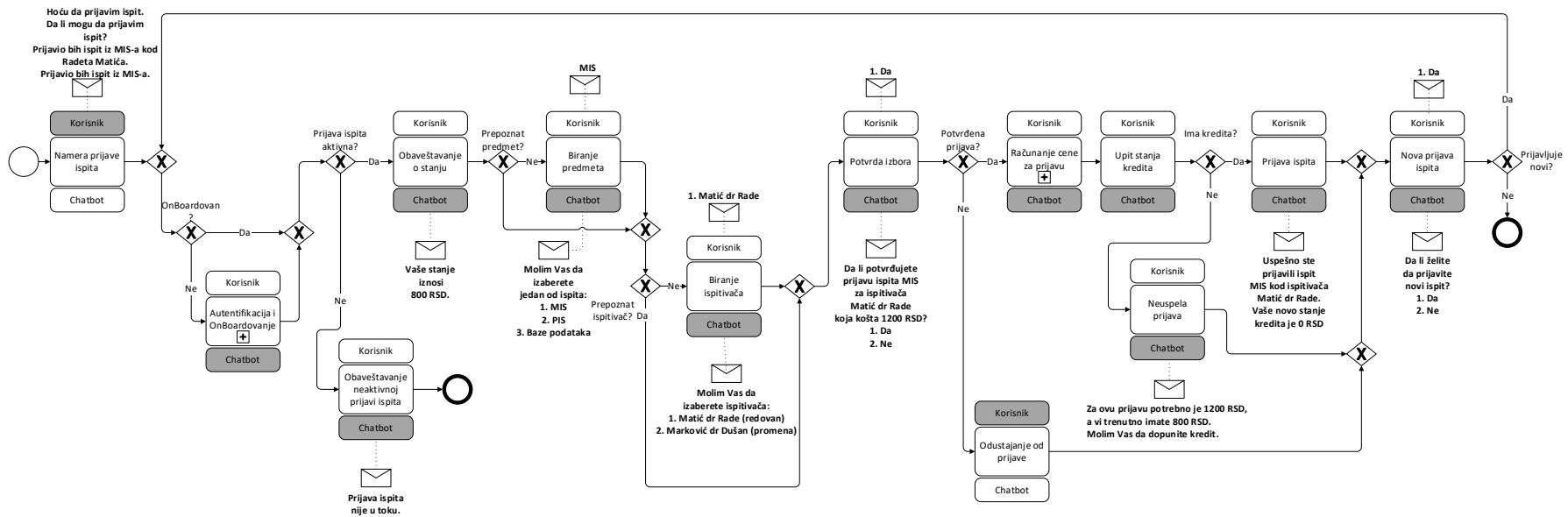
6.3. MODELOVANJE SCENARIJA KROZ KONVERZACIONI DIZAJNER

Ova aktivnost podrazumeva sledeće zadatke:

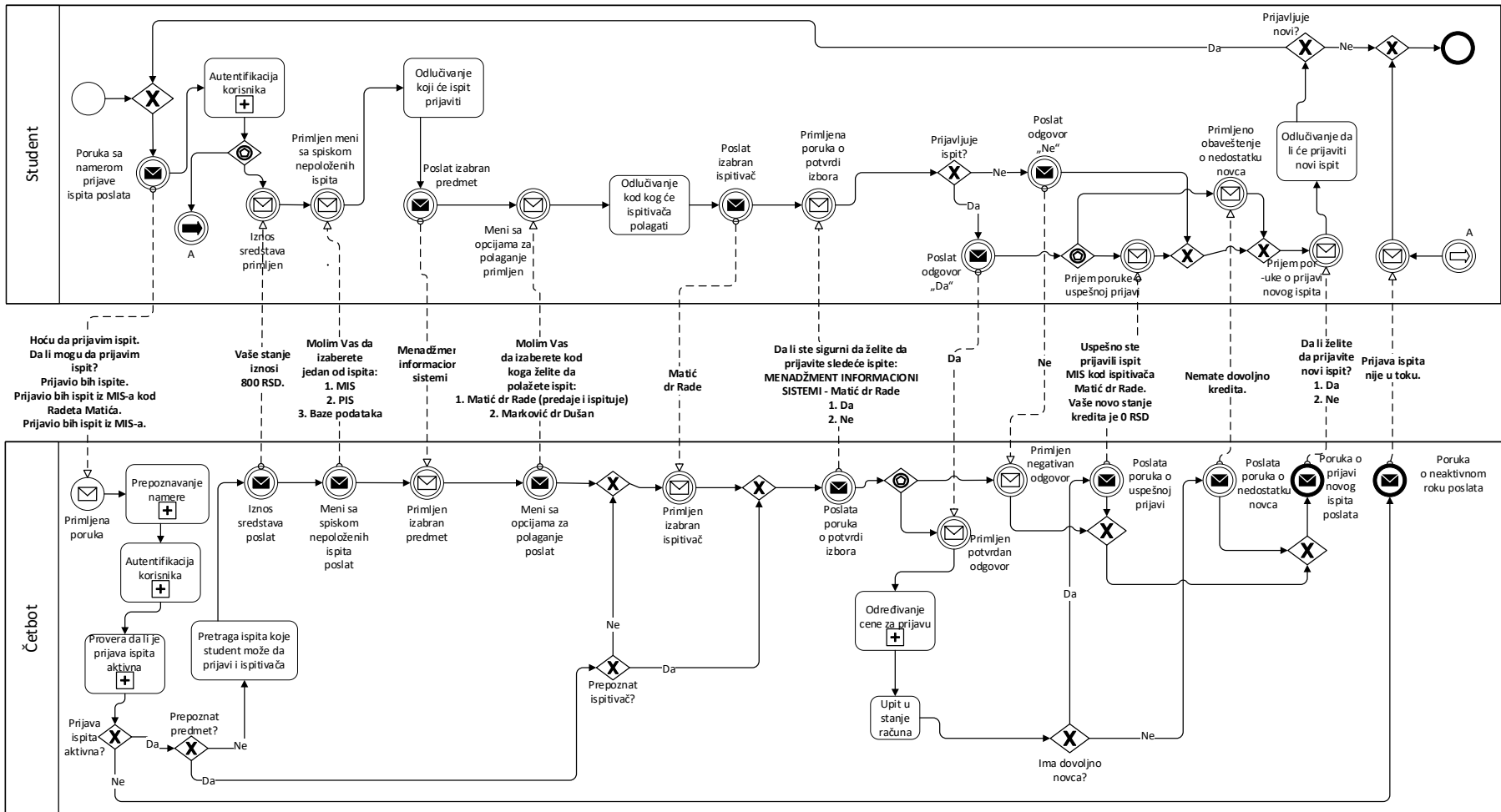
- Analiza slučajeva korišćenja i projektne dokumentacije iz prve faze aplikacionog inženjeringa.

- Precizno identifikovanje i specificiranje svih mogućih scenarija, njihovih tokova konverzacije kao i odgovora (akcija) od strane konkretnog četbota.
- Za svaki scenario je potrebno definisati odgovarajuća (ne)obavezna pitanja i odgovore koji će se implementirati, kao i početak i kraj konverzacije.
- Definisanje precizne IT specifikacije u slučaju izbora scenarija integracije sa aplikacijama/bazama podataka eksternih sistema. Ovo podrazumeva odgovore na pitanja: kada, gde i koje informacije treba uzeti od eksternih servisa.

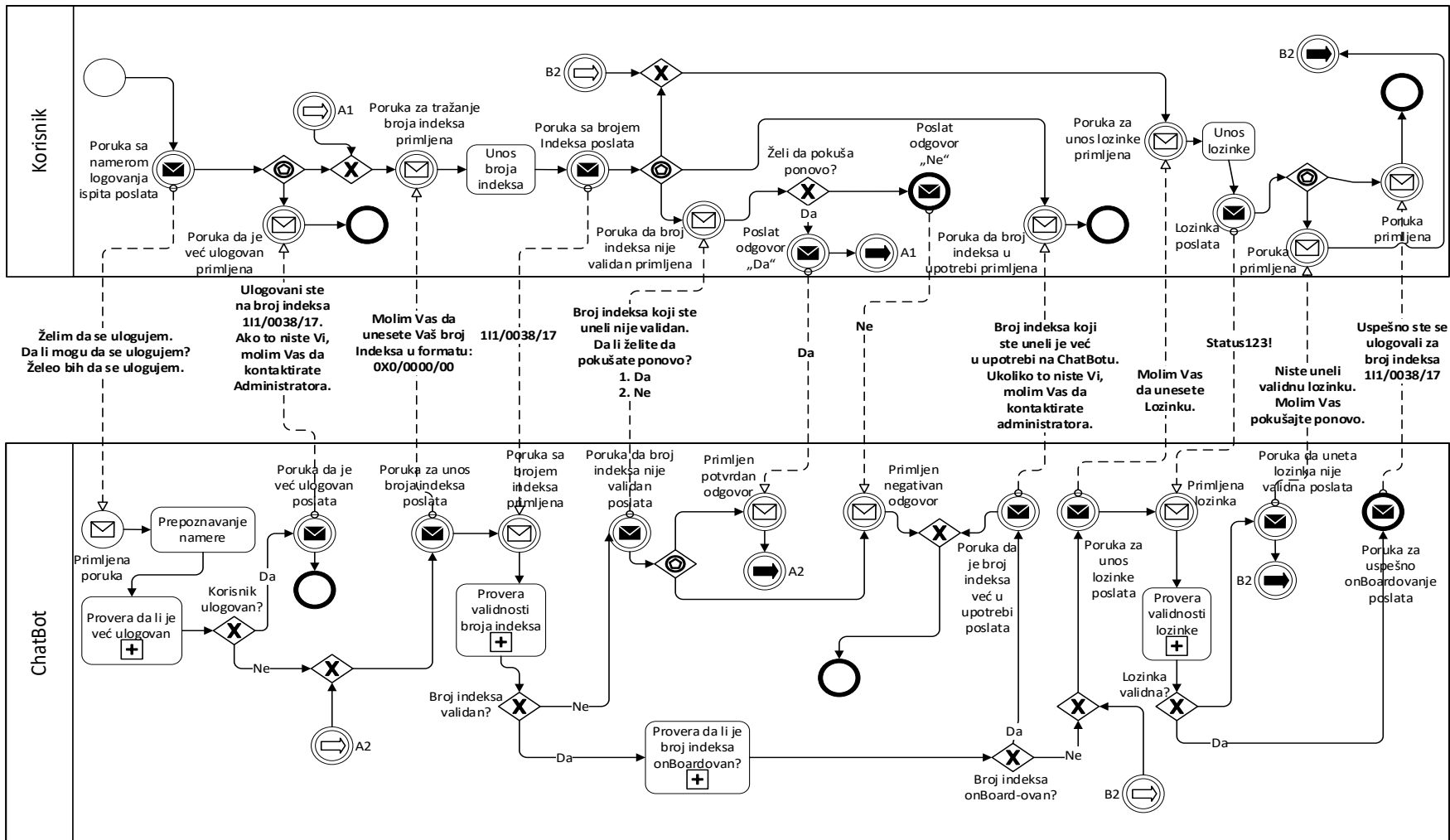
Modelovanje toka scenarija izabranih slučajeva korišćenja se realizuje korišćenjem standarda za modelovanje poslovnih procesa (engl. Business Process Modeling and Notation - BPMN) u konverzacionom dizajneru. U nastavku će biti prikazano samo nekoliko primera BPMN standarda odnosno njemu pripadajući dijagrami koreografije i dijagrami kolaboracije. Na slici 51. i 52. su prikazani dijagrami koreografije i kolaboracije za prijavu ispita, dok slika 53. prikazuje dijagram kolaboracije za verifikaciju studenata.



Slika 51. Dijagram koreografije za prijavu ispita u BOT dizajneru



Slika 52. Dijagram kolaboracije za prijava ispita u BOT dizajneru



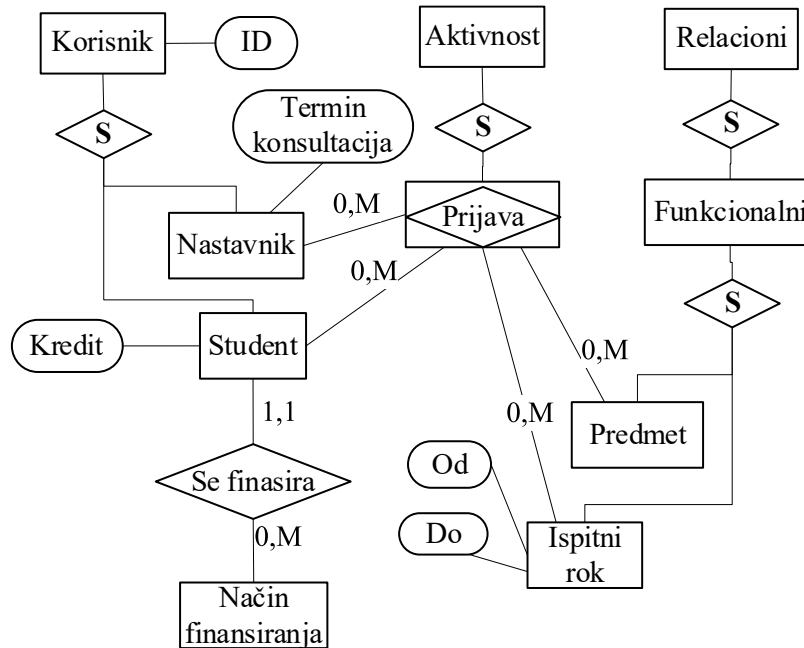
Slika 53. Dijagram kolaboracije za verifikaciju korisnika u BOT dizajneru

6.4. MODELOVANJE KONTEKSTA

U aktivnosti modelovanje konteksta, dizajner pravi kontekstni model. Uz pomoć dizajnerskog alata i posmatranjem opšteg kontekstnog modela iz projektnog inženjeringa, kontekstni dizajner treba da za svaku moguću definiciju scenarija (slučaja korišćenja) definiše lokalni kontekst odnosno kontekstni model. Ovakav lokalni kontekstni model mora da bude u skladu sa predloženim referentnim opštim kontekstnim modelom kao podrška u određivanju konkretnog konteksta, čime se želi postići odgovarajuća granularnost. Za interpretaciju scenarija ove kategorije konteksta su veoma bitne. Konkretni kontekst je rezultat koji je dobijen iz kontekstnih kategorija. Uopšteno gledajući u vreme izvršavanja, adaptivna platforma za razvoj četbota održava stanje kontekstnog modela u kom se čuvaju kontekstne informacije u svakom trenutku. Stanje ovakvog konteksta se u ovom rešenju realizuje kroz bazu podataka koja je definisana u vreme projektovanja ili kroz memorijski keš. Baza podataka, odnosno memorijski keš se ažurira i stanje konteksta se održava zahvaljujući konverzacijama i/ili komponentama platforme. Kada dođe do relevantnih promena konteksta, u vreme izvršavanja se okidaju određene komponente koje iniciraju proces adaptacije scenarija. Pojednostavljeni primer kontekstnog modela za tri slučaja korišćenja definisan od strane kontekst dizajnera je dat na slici 54. Da bi se adekvatno izvršili slučajevi korišćenja: provera konsultacija određenog nastavnika, prijava ispita i provera kredita studenta, razvojna platforma mora da čuva kontekstne informacije o studentu i nastavniku. Student može da proveri kad su konsultacije nastavnika, koliko ima kredita i može da prijavi ispit u vreme ispitnog roka kod odgovarajućeg nastavnika. Student pripada kategoriji konteksta „Korisnik“, Prijava ispita pripada kategoriji „Aktivnost“, Predmet i Nastavnik pripadaju kategoriji „funktionalnog“ konteksta.

Ovaj primer kontekstnog modela obuhvata informacije o studentu, njegovom načinu finansiranja, kreditu, ispitnom roku, predmetu i prijavi ispita kod određenog nastavnika. Ali takođe čuva informacije o terminu konsultacija nastavnika. Kada korisnik hoće da prijavi ispit ili proveri stanje kredita ili proveri kad su konsultacije, treba da se proverava da li u kontekstu ulogovanog korisnika stoji informacija da je u pitanju student. A zatim se proverava njegov način finansiranja i dostupan kredit. Ako su ove kontekstne

informacije o profilu studenta dostupne onda se proverava da li postoje informacije o ispitnom roku i da li je u poruci studenta unešena informacija koji predmet i kod kog nastavnika student želi da prijavi ispit. Za proveru termina konsultacija proverava se da li postoji nastavnik u kontekstu. U slučaju da neke od ovih informacija nedostaju u kontekstu, četbot treba da postavi pitanja i da dobije nedostajuće informacije potrebne za ovaj kontekst.



Slika 54. Primer kontekstnog modela

Iz ovih slučajeva korišćenja se vidi da različita scenarija mogu imati više istih ili različitih kontekstnih informacija za realizaciju odgovarajućih slučajeva korišćenja.

6.5. NLU UČENJE

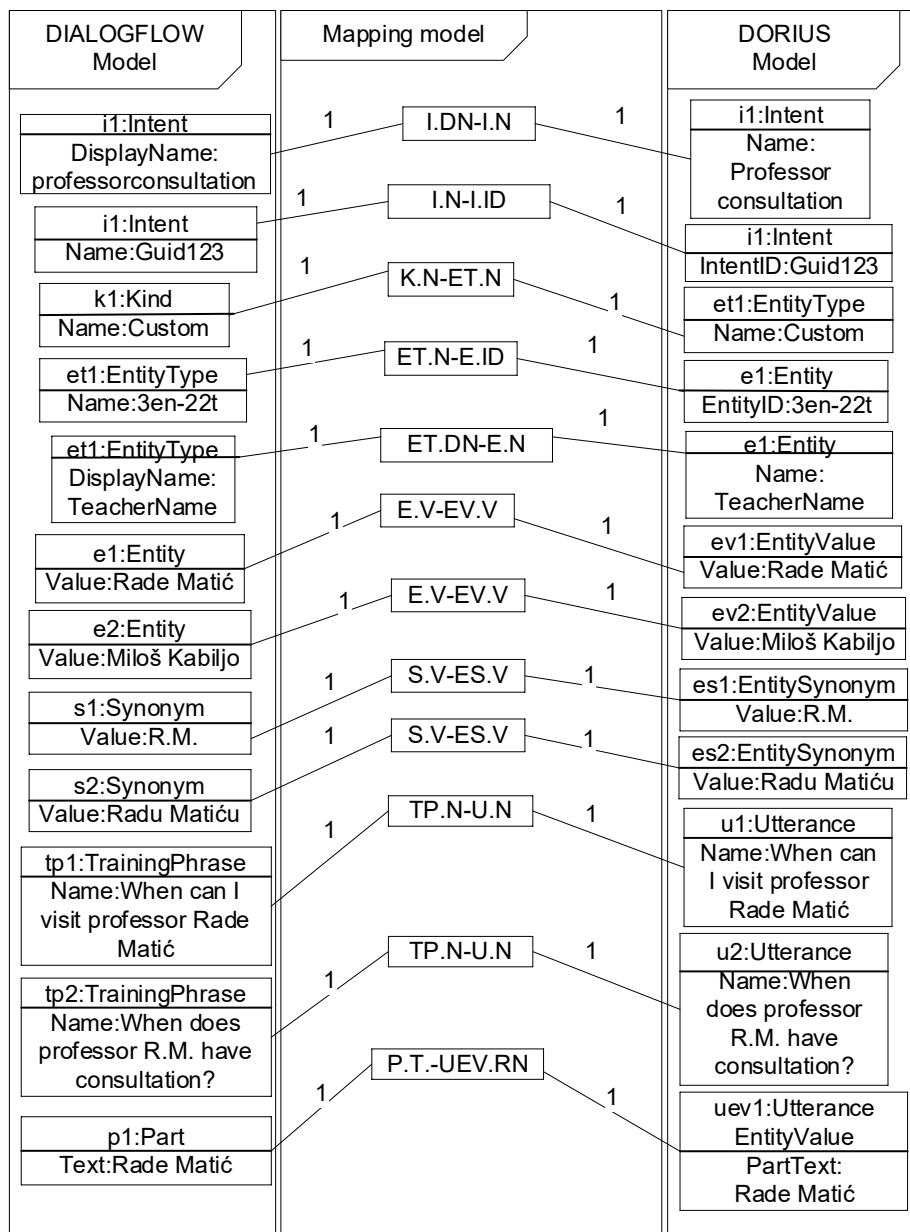
U ovoj aktivnosti dizajner bi po pravilu trebao da koristi alat za NLU učenje na izabranoj NLU platformi. Međutim, u ovde predloženom pristupu NLU trener treba za svaki mogući scenario da unese namere (engl. intent), entitete, sinonime i ubaci fraze (upite, rečenice) u opšti DORIUS NLU model. Kada se unesu ovi bitni koncepti za sva scenarija na DORIUS NLU modelu, jednostavno je metodom preslikavanja prebaciti sve koncepte u izabrane NLU platforme iz druge aktivnosti aplikacionog inženjeringa. Ovo se postiže uz pomoć odgovarajućih modela mapiranja i pravila mapiranja definisanih u projektnom inženjeringu. Nakon mapiranja vrši se iterativno validiranje scenarija (učenje i testiranje).

Primer dela odgovarajućeg modela mapiranja između DORIUS modela i *Dialogflow* modela je prikazan na slici 55. Paket DORIUS model sadrži instance DORIUS metamodela, dok paket *Dialogflow* model sadrži instance *Dialogflow* metamodela. Paket modela mapiranja sadrži instance odgovarajućih pravila mapiranja po kojima su koncepti *Dialogflow* mapirani sa konceptata *DORIUS* modela.

- Namera “i1” (ProfessorConsultation) je mapirana iz odgovarajuće namere istog imena koristeći pravilo I.DN-I.N.
- Vrsta (engl. Kind) “k1” je mapirana iz “et1:EntityType” istog imena koristeći pravilo mapiranja K.N-ET.N.
- Sa druge strane, atribut Name (3en-22t) koji pripada et1:EntityType, je mapiran iz atributa EntityID (3en-22t) koji pripada “e1:Entity” po pravilu ET.N-E.ID. Takođe, atribut DisplayName (TeacherName) koji pripada “et1:EntityType” je mapiran iz atributa Name (TeacherName) koji pripada “e1:Entity” po pravilu mapiranja ET.DN-E.N.
- Dalje, entiteti “e1 (Rade Matić)” i “e2 (Miloš Kabiljo)” su mapirani iz odgovarajućih vrednosti entiteta “ev1” i “ev2” iste vrednosti koristeći pravilo mapiranja E.V-E.VV.

- Sinonimi “s1 (R.M.)” i “s2 (Radu Maticu)” su mapirani ka odgovorajućim sinonimima entiteta “es1” i “es2” iste vrednosti koristeći pravilo “S.V-E.SV”.
- Fraze za trening (engl. Training phrases) “tp1” i “tp2” su mapirane iz odgovarajućih fraza (engl. Utterances) “u1” i “u2” istog imena po pravilu mapiranja TP.N-U.N.
- Atribut “Text” koji pripada “p1:Part” je mapiran iz atributa PartText koji pripada “uev1:UtteranceEntityValue” po pravilou P.T-UEV.RN. UtteranceEntityValue predstavlja vrednost entiteta u jednoj ili više fraza. Na primer, u frazi: „Kada profesor Rade Matic ima konsultacije?“, vrednost entiteta „Rade Matic” je PartText.

Ovde je dat primer modela mapiranja DORIUS modela sa jednim NLU modelom (*Dialogflow* model). Sličan primer mapiranja se može prikazati sa RASA modelom.



Slika 55. Deo modela mapiranja između *Dialogflow* i DORIUS model

U nastavku su dati delovi korisničkog interfejsa platforme *Weaver* za unos svih koncepata DORIUS modela koji su potrebni za učenje četbota na bilo kojoj NLU platformi. Kada se u DORIUS model unesu svi ovi koncepti oni se mapiraju na izabrane NLU platforme i pušta se učenje i svakodnevno testiranje. Svaka poruka (fraza, rečenica, upit) koja dolazi od strane korisnika analizira se u cilju detekcije značenja i ostalih značajnih informacija, na osnovu kojih četbot pruža odgovarajući odgovor. Podaci koje četbot pronalazi u korisničkom upitu nazivaju se Entiteti. Panel „Entiteti“ sa slike 56.

sadrži prikaz i administraciju svih entiteta (*Intent*, *Keyword*). Postoje sledeći tipovi entiteta:

1. Sistemski entiteti – vrednosti kao što su broj, datum, vreme, itd.
2. *Keyword* (srp. ključne reči) – fraze koje imaju posebno značenje i utiču na tok scenarija;
3. *Intent* (srp. namera) – kontekst odnosno značenje koje nosi ceo upit korisnika.

Weaver / NLP / Entitet

NLP entiteti

Pretraga 🔍 🔗 Izvezi 🔄 Osveži + Dodaj NLP objekat

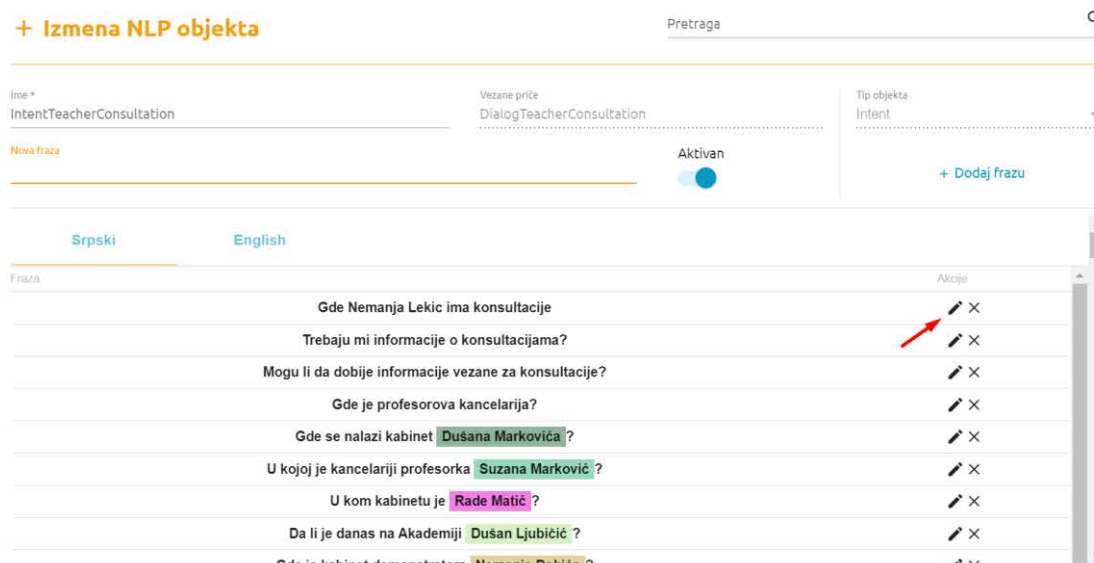
Ime	Tip objekta	Aktivan	Vezane priče	
IntentLibraryLocation	Intent	<input checked="" type="checkbox"/>	DialogLibrary	⋮
IntentStudentsMistakes	Intent	<input checked="" type="checkbox"/>	DialogStudentsMistakes	⋮
IntentChristmas	Intent	<input checked="" type="checkbox"/>	DialogNewYearAndChristmas	⋮
IntentSubjectByTeacher	Intent	<input checked="" type="checkbox"/>	DialogSubjectByTeacher	⋮
IntentGratitude	Intent	<input checked="" type="checkbox"/>	DialogGratitude	⋮
IntentTeacherConsultation	Intent	<input checked="" type="checkbox"/>	DialogTeacherConsultation	👁️ ✎ ✕
IntentWomanDay	Intent	<input checked="" type="checkbox"/>	DialogEightMarth	⋮

Slika 56. Panel Entiteti

Korisnik dodaje novi entitet klikom na dugme “Dodaj NLP objekat” gde se unose:

1. Ime: naziv entiteta;
2. Tip objekta: potrebno je odabrati tip objekta – *Intent* ili *Keyword*.

Izgled jednog entita koji je tipa *Intent* je dat na slici 57. i na njoj se vide sve rečenice (frazе) koje su ušle u učenje ove namere koja ima za cilj razumevanja namere konsultacija nastavnika (engl. *IntentTeacherConsultation*).



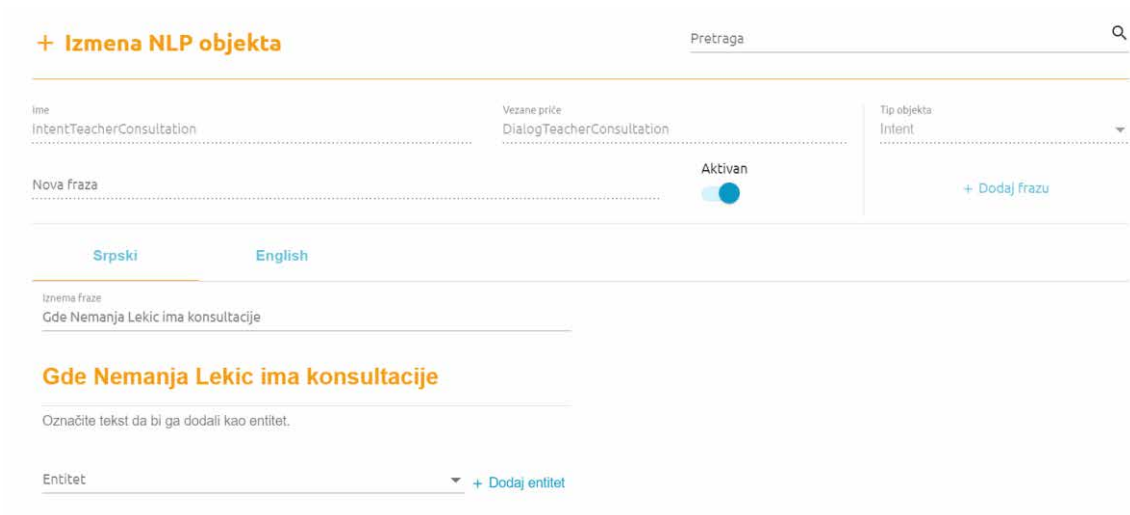
Slika 57. Prikaz entiteta *IntentTeacherConsultation* koji je tipa *Intent*

Na dobijenom pregledu entiteta moguće je videti sve unete rečenice (frazе) u *Intent*. Ovaj skup rečenica za učenje se najčešće naziva trening set. Entiteti, odnosno ključne reči su označene u tim rečenicama. Nove rečenice moguće je dodati klikom na dugme „Dodaj frazu“ i unošenjem u polje Nova vrednost.

Rečenice koje postoje u okviru ovog trening seta moguće je:

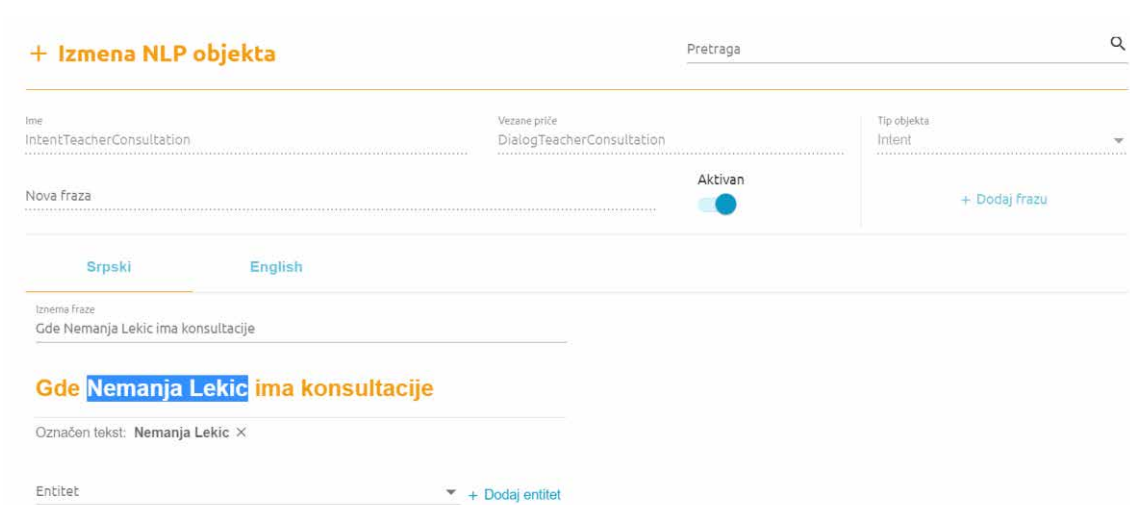
1. Izmeniti (klikom na akciju izmeni): obrisati deo, izmeniti rečenicu ili označiti entitet (ključnu reč);
2. Izbrisati (klikom na akciju „X“): izbrisati rečenicu iz trening seta.

Izmena fraze započinje otvaranjem panela Entiteti, pretragom odgovarajućeg entiteta i klikom na akciju izmeni. Klikom na izmenu odgovarajuće rečenice otvara se pregled kao na slici 58.



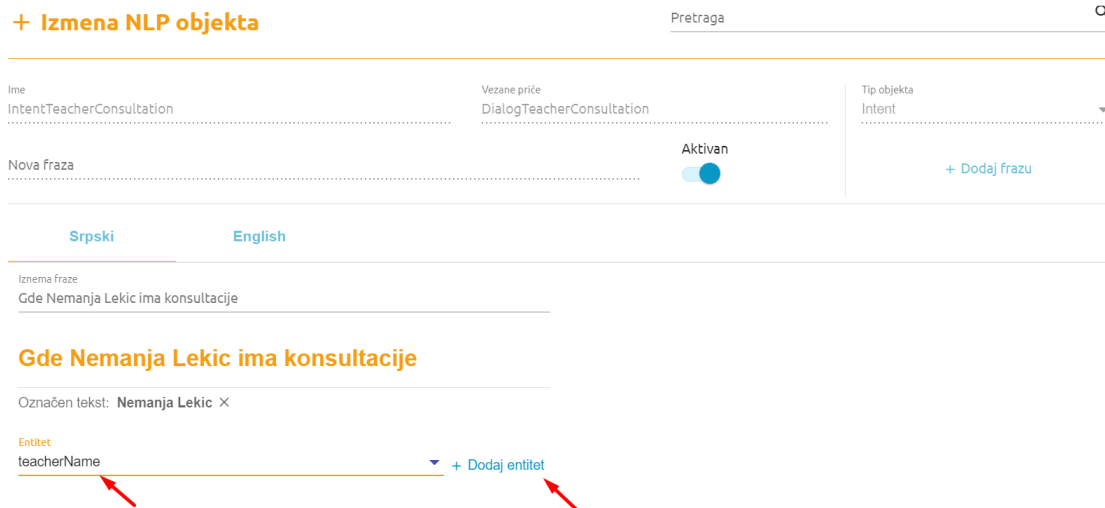
Slika 58. Izmena odgovarajuće rečenice

U narednom koraku potrebno je označiti ključnu reč obeležavanjem teksta kao na slici 59. Treba imati u vidu da će se označen tekst prikazati ispod rečenice, što je takođe prikazano na slici 59.



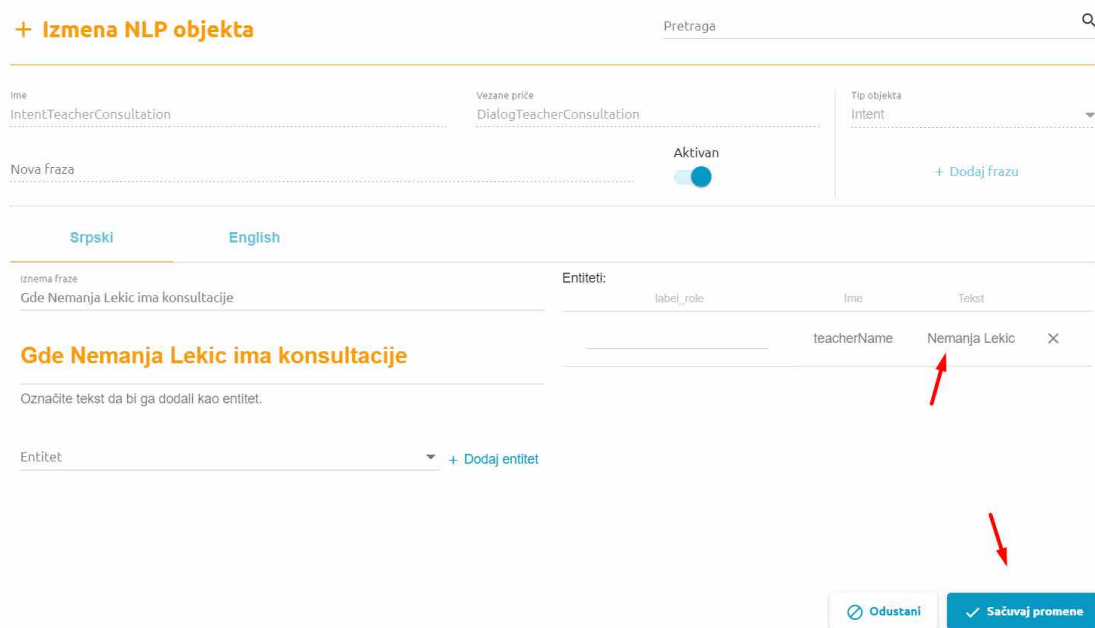
Slika 59. Označavanje ključne reči na rečenici

Nakon toga potrebno je kliknuti na padajuću listu Entitet (opcija ispod označenog teksta) i izabrati odgovarajući entitet, odnosno ključnu reč, a zatim izabrati opciju „dodaj entitet“ koja se nalazi pored padajuće liste kao na slici 60.



Slika 60. Dodavanje označenog teksta kao entiteta

Zatim se pojavljuje prikaz sa desne strane gde je prikazan označen entitet, nakon čega je potrebno sačuvati promene kao na slici 61.



Slika 61. Čuvanje rečenice sa označenim entitetom

Nakon sačuvane rečenice, potrebno je sačuvati promene i na *Intent* NLP objektu. Rečenica sa obeleženim entitetom će se uspešno ubaciti u trening set. A zatim se taj trening set pušta na konkretno izabrane platforme za učenje.

6.6. TESTIRANJE KONKRETNOG ČETBOTA

Ovde je potrebno uspostaviti testno okruženje sa početnim podešavanjima četbota. U ovoj aktivnosti se vrši iterativno validiranje slučaja korišćenja (testiranje, učenje). Testiranje se vrši na osnovu specifikacije scenarija iz treće aktivnosti aplikacionog inženjeringa koja predstavljaju osnovu za predtestiranje kao i pisanja scenarija za alfa test. Alfa testiranje podrazumeva interno testiranje tima stručnjaka koji treba da otkriju sve greške. Neke greške se ispravljaju odmah tokom samog testiranja. U ovom alfa testu se svakako dodatno ubacuju nova učenja i promene odgovora nakon čega se radi regresivno testiranje. U ovom testiranju dodatno se ispravljaju greške, ubacuju nova učenja, prilagođava i/ili menjaju odgovori i sve što je potrebno za sledeću fazu beta testiranja. Na osnovu rezultata alfa testiranja prilagođavaju se scenarija testiranja poslovnim korisnicima iz domenskog okruženja za potrebe beta testiranja.

Beta testiranje podrazumeva vrstu testiranja koje je omogućeno poslovnim korisnicima (nosiocima domenskog znanja) u realnom testnom okruženju kako bi eksperimentisali i prijavili probleme koji se javljaju pri upotrebi u realnim uslovima. Tokom ovih testiranja, potrebno je pratiti sve interakcije između četbota i korisnika. Ovde se takođe ponavlja ciklus učenja sa novim frazama. Nakon toga zajedno sa korisnicima se ponovo radi dodatni regresivni test gde se konačno potvrđuje učenje i eventualno ubacuju dodatne izmene i nova učenja. Regresivno testiranje podrazumeva ponavljanje testova koji su prošli u prethodnim iteracijama čime se obezbeđuje da funkcionalnosti i dalje rade bez grešaka i da su u skladu sa specifikacijom.

Odmah posle sprovedenog alfa i beta testiranja od strane poslovnih korisnika, pristupa se izradi konačne specifikacije scenarija koja odgovaraju implementiranim scenarijima za sve slučajeve korišćenja.

6.7. PUŠTANJE ČETBOTA NA PRODUKCIJU I VALIDACIJA FRAZA

U ovoj aktivnosti se četbot se konačno pušta na produkciju. Nakon ove aktivnosti potrebno je obavezno praćenje konverzacije korisnika sa četbotom i dodatno unapređenje postojećih scenarija. To podrazumeva validiranje i učenje novih fraza (upita, rečenica) čime četbot postaje pametniji. Ideja je da se znanje instance četbota konstantno povećava. Osim validacije ponovo se radi novo učenje i testiranje četbota. Validacija je komponenta namenjena validiranju fraza (upita, rečenica) dospelih od korisnika do četbota. Svrha validacije je povezivanje SVAKE korisnikove poruke (frazе, upita, rečenice) sa postojećim ili potencijalno novim scenarijima ili odbacivanje istih. Validiranje poruka (frazа, upita, rečenice) podrazumeva prepoznavanje fraza koje četbot ne zna, dodatno treniranje četbota tim frazama, kao i prepoznavanje novih scenarija i dolazak do subjektivnog zaključka koliko četbot dobro odgovara. Validator mora da prođe svaku poruku (frazu, upit, rečenicu) ručno i da označi da li je validna ili ne. Veoma bitan preduslov za validaciju je da osoba koja vrši validaciju mora da bude neko ko odlično poznaje sva scenarija i sve odgovore koje četbot zna.

Slika 62. prikazuje NLU validaciju fraza za nameru “provera konsultacija nastavnika” koje su studenti tražili, kao i njihov procenat tačnosti (poverenja) na RASA NLU servisu. Korišćenjem NLU validacije moguće je direktno podučavati NLU servis kako bi se izbegla zavisnost od NLU interfejsa dobavljača.

Funkcionalnost procesa validacije je prikazana kroz nekoliko sledećih slika. Klikom na opciju pregled sa slike 63. se otvara pregled konverzacije dat na slici 64. Ako odgovor koji je četbot dao korisniku odgovara frazi, kao što je to na ovom primeru prikazano, potrebno je izbaciti rečenicu iz NLP validacije i to se radi klikom na opciju izbriši, kao što je prikazano na slici 65.

✓ NLP Validation Refresh RASA

Utterance	Values	Valid	Accuracy
kako ide mail profesora Radeta	IntentTeacherContact "null",teacherName "Rade Matic"	✓	86%
koji je mail radeta matica	IntentTeacherContact "null",teacherName "Rade Matic"	✓	92%
Kada su konsultacije kod Radeta Matica?	IntentTeacherConsultation "null",teacherName "Rade Matic"	✓	100%
Kako mogu da kontaktiram nastavnika R. M.	IntentTeacherContact "null",teacherName "Rade Matic"	✓	90%
Koji je kontakt nastavnika Radeta Matica	IntentTeacherContact "null",teacherName "Rade Matic"	✓	95%
Gde je kabinet Rada Matica?	IntentTeacherConsultation "null",teacherName "Rade Matic"	✓	100%

Slika 62. NLU validacija

Weaver / NLP / NLP Validacija

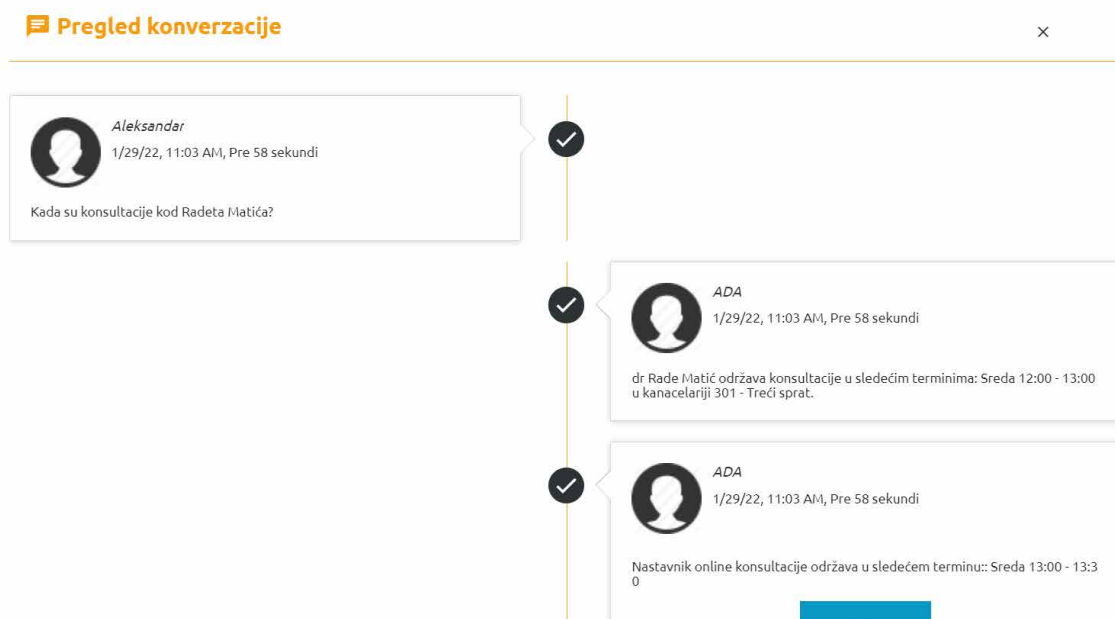
✓ NLP Validacija

Multi intent
 Validno / Nevalidno
 Pretraga
 Osveži
 ✓ Selektuj stranicu
 ✓ Mark All
 ✗ Uklanjanje selektovanog

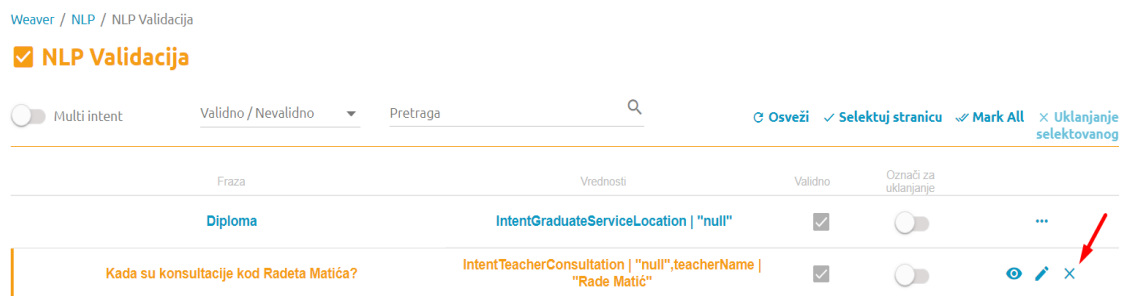
Fraza	Vrednosti	Validno	Označi za uklanjanje
Diploma	IntentGraduateServiceLocation "null"	✓	<input type="checkbox"/>
Kada su konsultacije kod Radeta Matica?	IntentTeacherConsultation "null",teacherName "Rade Matic"	✓	<input checked="" type="checkbox"/>

Rezultata po strani 10 221 - 222 od 222 < >

Slika 63. Pregled validacije



Slika 64. Pregled Konverzacije



Slika 65. Brisanje NLP Validacije

Proces validacije korisnikove poruke (fraze, upita, rečenice) se vrši za svaku poruku. Administrator ili validator ručno validira svaku poruku i ima mogućnost izbora odluke jednog od sledeća četiri različita izbora validacije:

1. Razumljiva i naučena fraza vodi do PREPOZNAVANJA dovoljno TAČNE namere i tačnog scenarija.

U ovom slučaju četbot je tačno odgovorio i potrebno je validirati odgovor kroz interfejs platforme.

2. Razumljiva i naučena fraza vodi do PREPOZNAVANJA dovoljno NETAČNE namere koja ukazuje na netačan scenario.

U ovom slučaju četbot je netačno odgovorio i potrebno je validirati odgovor kroz interfejs platforme.

3. Razumljiva i naučena fraza vodi do NEPREPOZNAVANJA dovoljno TAČNE namere i ukazuje na scenario “nisam razumela”.

U ovom slučaju četbot je odgovorio da ne zna. Međutim, četbot bi trebalo da nauči frazu tj. potrebno je ubaciti frazu u učenje za odgovarajuću nameru.

4. Nerazumljiva ili nenaučena fraza vodi do NEPREPOZNAVANJA dovoljno TAČNE namere i ukazuje na scenario “nisam razumela”.

U ovom slučaju četbot je odgovorio da ne zna. I to je sasvim u redu, jer je stigla poruka koju četbot još ne treba da uči tj. ne može da prepozna nameru ILI je stigla nerazumljiva fraza i nemoguće je prepoznati nameru korisnika.

Okidač za pokretanje ponovnog treninga je ubacivanje (dodavanje) nove poruke (frazе, upita, rečenice) u učenje, što je opisano validacijom. Ubacivanje se realizuje u trenutku dodavanja nove fraze u odgovarajuću nameru.

7. EVALUACIJA ČETBOTA ADA

U uslovima pandemije KOVID-19 i zbog zatvaranja obrazovnog sistema, Beogradska akademija poslovnih i umetničkih strukovnih studija (BAPUSS) je bila u obavezi da omogući onlajn obrazovni sadržaj i interakciju sa svojim studentima. Da bi poboljšala, modernizovala i digitalizovala obrazovne usluge, BAPUSS je razvila ADA četbot koristeći *Weaver* platformu kako bi svojim studentima pružila poboljšanu uslugu i neophodne informacije tokom studiranja. ADA četbot spada u zatvorene domene jer pruža lakše i brže informacije sadašnjim i budućim studentima akademije. Lako je komunicirati sa ADA-om jer imate osećaj da razgovarate sa prijateljicom koja može da obavi složene zadatke kao što je prijava ispita. ADA prima poruke sa različitih komunikacionih kanala (Viber, FB mesindžer i vebčeta preko sajta BAPUSS). Platforma obavlja veliki broj važnih zadataka pre nego što pošalje poruku NLU. NLU analizira tekst i na osnovu tehnika mašinskog učenja upoređuje, a zatim generiše nameru i njene entitete. Generisani rezultat se šalje nazad u platformu, koja određenim algoritmom koristi nameru, entitete, kontekst i šalje odgovor korisniku u odgovarajućem formatu za prikazivanje poruke.

ADA je dostupna 24 sata i daje odgovore na česta pitanja studenata i budućih studenata. ADA zna nešto više od 250 scenarija u vezi sa pitanjima na koje studenti i ostali korisnici žele da dobiju odgovor ili da izvrše odgovarajuću uslugu. Iako možemo da kategorišemo da je ADA informativnog i transakcionog karaktera, tipovi scenarija koje je ADA zna su kategorisani na sledeći način:

- Kratak razgovor (engl. small talk) – glavna svrha ove vrste razgovora je da zadrži korisnike na kanalu i doprinese razgovoru sličan ljudima. Kratak razgovor se naziva neobaveznim razgovorom. Uglavnom obuhvata: pozdrave, zabavne fraze, pozitivna osećanja, negativna osećanja, najnovije emisije i pesme, pitanja za testiranje četbota.
- Česta pitanja i odgovori (engl. Frequent QA) - scenarija orijentisana na potrebe studenata (nastavnik, predmet, predavanja, literatura, obrazovanje itd.)

- Različite vrste transakcija - prijavljivanje ispita, upis i overa semestra, otkazivanje ispita, promena ispitivača, promena izbornog predmeta, kupovina knjige, resetovanje lozinke itd.

Neki od scenarija iz kategorije čestih pitanja i odgovora su informacije o [84]:

- konsultacijama nastavnika,
- ispitnim rokovima,
- rangiranju akademije,
- rasporedu predavanja,
- rangiranju studenata za budžetsku listu,
- događajima,
- dostupnost usluga,
- akademskom okruženju,
- terminima predavanja i ispita,
- aktivnostima koje se odvijaju u okviru Akademije i
- druge korisne akademske informacije.

ADA koristi NLU servise da razume nameru, a onda tek pronalazi adekvatne odgovore na sva studentska pitanja i reši njihove probleme. Na taj način se povećava zadovoljstvo studenata, a oni ostvaruju bolju interakciju sa BAPUSS koristeći više komunikacionih platformi i NLU servisa. Ono što najbolje govori o uspehu ove implementacije je to što je BAPUSS celu kampanju imidža brenda posvetila ADA četbotu.

7.1. TESTIRANJE ADA ČETBOTA

Celokupno testiranje je obavljeno u tri faze. Prva faza je podrazumevala interno testiranje namera zbog specifičnosti srpskog jezika. U ovoj fazi su potvrđeni nedostaci *Dialogflow* iz rada [59] i eksperimentalno se prešlo i na RASA NLU. Jedan od dodatnih

razloga prelaska na RASA NLU je bio i cena korišćenja *Dialogflow*. Druga faza testiranja obavljena je u kontrolisanom okruženju koristeći *Dialogflow* i RASA NLU sa grupom od 60 studenata sa smera Informacioni sistemi i tehnologije. Ovim studentima nije rečeno šta ADA zna, a takođe im nije bilo rečeno kako da pitaju. Cilj je bio da se postigne tačnost prepoznavanja namere i entiteta od minimum 75%. Nakon prvog testa sa različitim komunikacionih platformi (Viber, FB mesindžer i vebčeta preko sajta BAPUSS), kao što je prikazano u tabeli, sa 1516 pitanja studenata, procenat tačnosti prepoznavanja namere je bio 52,7 %. Ovo je bilo zadovoljavajuće bez obzira na željeni veći procenat tačnosti, jer studentima nije otkriveno šta je ADA sve naučila. Oni su koristili različite kanale i postavljali su pitanja iz različitih domena, a dobijali su odgovore koje je platforma generisala analizirajući namere, entitete koristeće različite NLU servise.

Tabela 6. Statistika pitanja i odgovora

ADA	Broj	Procenat
Ukupan broj pitanja (poruka)	1516	100%
Neprihvatljiva pitanja (Besmislena pitanja, nema potrebe za odgovorom)	358	23,6% (358 of 1516)
Prihvatljiva pitanja (bez besmislenih pitanja)	1158	76,4% (1158 of 1516)
Prihvatljivi odgovori (od prihvatljivih pitanja)	610	52,7% (610 of 1158)
Neprihvatljivi odgovori koje treba ADA da nauči (pogrešna namera ili bez odgovora)	548	47,3% (558 of 1158)

Kao što se vidi iz druge faze, model učenja je trebalo pojačati učenjem sa izrazima koje ADA u drugoj fazi nije mogla da prepozna. Nakon toga, model učenja je postao mnogo bogatiji novim frazama. U trećoj i poslednjoj fazi testiranja uključeni su svi studenti IT smera na Akademiji. Oni su dali mnoge fraze koje su vremenom ubacivane u trening podatke, odnosno u učenje sve dok se nije došlo do toga da ADA odgovori na 85% tačnosti prepoznavanja namere. Neka pitanja isključena su iz treninga i ocene poverenja jer su te fraze bile van okvira naučenih scenarija. Međutim, neka od njih su ušli u krug potencijalno novih scenarija koja će se učiti u nekoj budućoj fazi unapređenja ADE. Da bi izvukli maksimum iz podataka sa testa, model učenja je unapređen i korišćen na različitim kanalima, različitim količinama podataka o obuci i različitim NLU servisima. Za podršku srpskog jezika nije korišćena biblioteka *spaCy* [85]. Za engleski i

druge popularne jezike *spaCy* je veoma korisno uključiti. Bez obzira koji jezik koristili, veoma je važno izvući maksimum iz treniranih podataka. Bilo je potrebno da se trenira i proceni model učenja na različitim planovima izvršavanja i različitim količinama trening podataka.

Nakon uspešnih faza testiranja ADA je puštena na produkciju. ADA sada pomaže budućim studentima da saznaju više o Akademiji. ADA je obučena da studentima pruži najkvalitetnije obrazovne usluge sa širokim spektrom informacija o nastavnicima, učionicama, nastavnom planu i programu, literaturi, radnom vremenu, biblioteci, predmetima, obaveštenjima, sadržajima obrazovanja, rasporedu predavanja, cenovniku, itd. Studenti takođe traže od ADE različite informacije kao što su: visina školarine, koja dokumenta su potrebna za upis u novu školsku godinu, raspored predavanja ili datum konsultacija određenog profesora. ADA može da obavi neke veoma bitne usluge i procese kao što su registracija ili otkazivanje ispita, promena nastavnika na ispitu, resetovanje lozinke ili kupovina knjige. Zahvaljujući tome, bez obzira što postoji veb orijentisani informacioni sistem BAPUSS-a, uspeh je taj što skoro 30% studenata koriste ADA kao primarni kanal komunikacije sa Akademijom. ADA je prvi četbot u obrazovanju koji obezbeđuje lakšu i bolju komunikaciju, štedi na vremenu jer pruža efikasniju realizaciju usluga u obrazovanju. Četbot tehnologija može značajno da utiče na studentsku percepciju obrazovne institucije i načina edukacije. Cilj platforme, pored toga što je na raspolaganju velikom broju studenata za sve potrebne informacije u vezi sa akademijom, jeste i dobijanje zvaničnih informacija bez čekanja, kao i da rastereti kontakt centre i nastavnike da svoju pažnju usmere na kreativniji zadatak. Četbot je besplatno dostupan na srpskom jeziku preko Vibera, FB mesindžera i veb sajta akademije, a trenutno je u fazi testiranja i engleski jezik kao i glasovni tip poruke.

7.2. STATISTIKA KORIŠĆENJA ADA ČETBOTA

Od septembra 2020. godine, kada je ADA počela da se koristi, ona beleži rast broja korisnika, broja primljenih poruka i odgovora. Statistike ukazuju na sledeću podatke date u tabeli 7. Može se primetiti da je broj verifikovanih korisnika preko sajta jednak

nuli, jer je nemoguće izvršiti dvofaktorsku verifikaciju korisnika preko sajta. Verifikovani studenti su oni koji su prošli dvofaktorsku autentifikaciju neophodnu za neku vrstu plaćanja ili drugu vrstu osetljivih usluga.

Table 7. Broj verifikovanih korisnika i primljenih poruka

	Viber	FB mesindžer	Veb sajt	Total
Broj korisnika	4.251	623	8.574	13.448
Broj verifikovanih korisnika	2.336	158	0	2.494
Broj primljenih poruka poslatih od korisnika	180.966	11.544	26.683	219.193

Izvor: ADA baza podataka

S obzirom da je srpski jezik veoma težak, bio je potreban veoma veliki trening set sa raznim frazama da bi model učenja bio što efikasniji. Pored fraza, vodilo se računa o padežima, entitetima, njihovim vrednostima kao i njihovim sinonimima zbog složenosti srpskog jezika što može predstavljati problem prilikom prepoznavanja namere. ADA ima mnogo entiteta koji imaju istu vrednost, ali različito značenje. U tom slučaju koristio se koncept uloge entiteta, tako da isti entitet u rečenici može imati dva značenja. Primer ovog entiteta je „menadžment“, što može biti naziv predmeta na jednom studijskom programu, ali takođe predstavlja i naziv određenog studijskog programa. U nastavku je data tabela 8. koja pokazuje broj namera, fraza učenja, entiteta i sinonima koji omogućavaju da ADA bude funkcionalna na srpskom jeziku. Ovi brojevi rastu jer se stalno radi na učenju i poboljšanju modela.

Table 8. Broj namera, fraza, entiteta, vrednosti entiteta i sinonima

	ADA
Intent	208
Utterance	8279
Entity	20
Entity values	525
Entity synonyms	3546

Izvor: ADA baza podataka

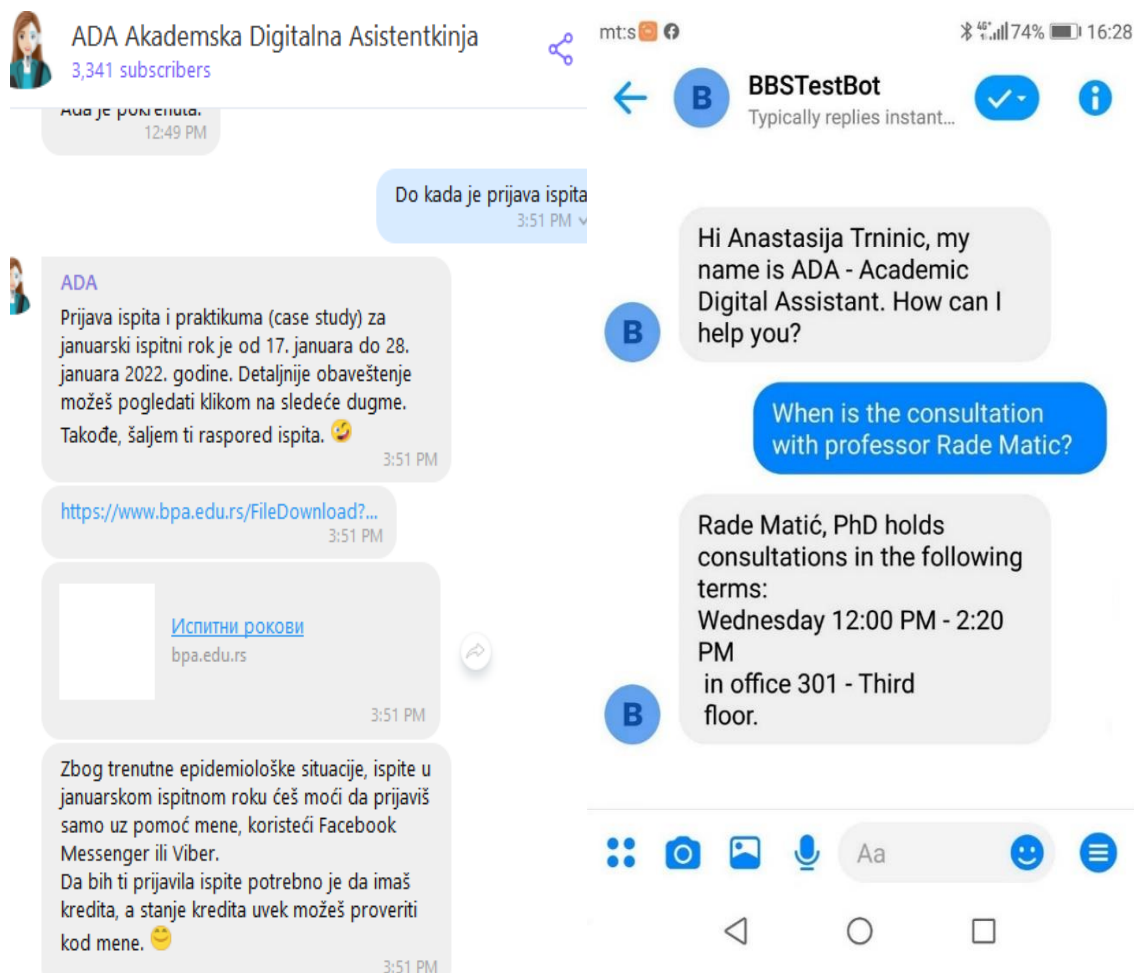
Analiza statistike razgovora po kanalima u periodu September 2020 – Decembar 2020 / Januar 2021 – Decembar 2021 / Januar 2022 data je u tabeli 9. U analiziranim mesecima dominantan kanal preko kojeg je poruka upućena ADA četbotu je bio Viber. U 2020. godini, 70,43% od ukupnih poruka. U 2021. godini kroz Viber je prošlo 79.60% od ukupnog broja poruka, a u januaru 2022. čak 85.96% od ukupnog broja poruka tog meseca.

Table 9. Broj konverzacija po kanalima u različitim periodima

	September 2020 – Decembar 2020	Januar 2021 – Decembar 2021	Januar 2022
Konverzacija preko FB mesindžera	71,25	406,92 (+471.12%)	972 (+138.87%)
Konverzacija preko vibera	3.970,00	5.899,42 (+48.60%)	14.222 (+141.08%)
Konverzacija preko veb sajta	1.595,25	1.104,25 (-30.77%)	1.350 (+22.25%)
Ukupno	5.636,5	7.410,59 (+31.48%)	16.544 (123.25%)

Izvor: ADA baza podataka

ADA komunicira koristeći srpski jezik, ali treba napomenuti da je lako prebaciti i naučiti ADU da komunicira na engleskom jeziku. Na levoj strani slike 66. prikazana je komunikacija na srpskom jeziku na Viber kanalu, a na istoj slici sa desne strane je prikazan primer testne komunikacije na engleskom jeziku na FB mesindžer kanalu. Slika 66 sa leve strane takođe pokazuje broj trenutnih korisnika ADE četbota na Viber kanalu koji iznosi 3341.



Slika 66. Viber i FB messenger kanali komunikacije

Studenti od ADE takođe traže informacije kao što su: visina školarine, koja dokumenta su potrebna za upis u novu godinu studija, raspored predavanja ili datum konsultacija određenog profesora. U tabeli 10 su prikazani podaci analize na početku eksploatacije ADE, odnosno poslednja tri meseca iz 2020. godine. U ovoj tabeli se vidi da je interesovanje studenata bilo u skladu sa školskim kalendarom i obavezama. Tako vidimo da je u septembru, mesecu polaganja ispita, studente najviše zanimalo kada i šta od predmeta nastavnik predaje i ispituje. U oktobru, mesecu upisa specijalističkih i master studija u novu školsku godinu, na vrhu liste upita se našla potreba za informacijama o visini školarine i vrsti dokumenata potrebnih za upis. U novembru studenti se raspituju o profesorskim konsultacijama i kancelariji, radnom vremenu akademije itd.

Tabela 10. Prvih 10 scenarija i broja konverzacija (bez kratkih razgovora) u periodu od septembra do decembra 2020. godine

Septembar 2020.		Octobar 2020.		Novembar 2020.	
Scenarija	Broj konver	Scenarija	Broj konver	Scenarija	Broj konver
Koje predmete profesor predaje i ispituje	178	Potrebna dokumentacija za upis određene školske godine	513	U kom kabinetu i u koje vreme određeni profesor održava konsultacije	918
Raspored predavanja	176	Iznos školarine	472	Lokacija, radno vreme i koje knjige se nude u skriptarnici Akademije	789
Kada u datom semestru počnu predavanja	140	U kom kabinetu i u koje vreme određeni profesor održava konsultacije	470	Potrebna dokumentacija za upis određene školske godine	719
Iznos školarine	137	Lokacija, radno vreme i koje knjige se nude u skriptarnici Akademije	459	Iznos školarine	630
Kada počinje upis studenata	134	Informacije o rangiranju budžeta	298	Na kom spratu se nalazi određeni kabinet	540
Potrebna dokumentacija za upis određene školske godine	128	Na kom spratu se nalazi određeni kabinet	269	Koje predmete profesor predaje i ispituje	480

Lokacija, radno vreme i koje knjige se nude u skriptarnici Akademije	113	Koje predmete profesor predaje i ispituje	267	Informacije o rangiranju budžeta	459
Na kom spratu se nalazi određeni kabinet	109	Kada počinje upis studenata	258	Kada počinje upis studenata	404
Informacije o radnom kalendar	81	Brojevi telefona šaltera za pojedine studijske programe	138	Brojevi telefona šaltera za pojedine studijske programe	173

Izvor: ADA baza podataka

Slična tabela, ali sa većim periodom analize pokrenutih scenarija je data u nastavku rada.

Tabela 11. TOP 10 scenarija i broja konverzacija (bez kratkih razgovora) u periodima: Septembar 2020 - Decembar 2020 / Januar 2021 - Decembar 2021 / Januar 2022

September 2020 – Decembar 2020		Januar 2021 – Decembar 2021		Januar 2022	
Scenarija	Broj konver.	Scenarija	Broj konver.	Scenarija	Broj konver.
Konsultacije profesora	1129	Prijava ispita	6610	Informisanje o načinima overe semestral	1743
Provera iznosa sa cenovnik Akademije	1109	Verifikacija studenta	3994	Provera iznosa sa cenovnik Akademije	1461
Lokacija, radno vreme i koje knjige se nude u skriptarnici Akademije	752	Provera iznosa kredita studenta	3927	Prijava ispita	1321

Potrebna dokumentacija za upis određene školske godine	719	Provera iznosa sa cenovnik Akademije	2761	Overa semestra	1064
Na kom spratu se nalazi određeni kabinet	550	Provera personalizovanih termina ispita	2655	Provera iznosa kredita studenta	984
Koji profesori predaju određen predmet	523	Informisanje o terminima konsultacija	2620	Provera personalizovanih termina ispita	912
Kako se student može upisati na studije	445	Kako se student može upisati na studije	2515	Informacije o rangiranju budžeta	984
Informacije o radnom kalendaru	354	Provera lokacije i radnog vremena skriptarnice Akademije	2429	Automatska verifikacija	586
Informacije o rang listi za budžet	351	Informisanje o ispitnom rokovima	1869	Verifikacija studenta	509

Izvor: ADA baza podataka

Verovatno najveće dostignuće ADA je nezavisnost od NLU eksternih usluga i komunikacionih platformi (kanala). ADA koristi više komunikacionih kanala i više NLU servisa da razume pitanja studenata integrišući moderno iskustvo razmene poruka kroz veb sajt akademije putem veb časkanja. ADA trenutno može da koristi dva eksterna NLU servisa: RASA i *Dialogflow*. Zahvaljujući rešenju koje je ovde predloženo, bilo je lako preći sa jednog NLU servisa na drugi ili koristiti oba. U tabeli 12 je dat prikaz korišćenja NLU servisa od strane ADA. Iz priloženog se vidi da je najveći broj korisnika i prepoznatih namera urađen koristeći RASA NLU. Razlog za ovako veliku razliku je jednostavan. Naime, RASA je još uvek otvorenog koda i veoma dobro radi u poređenju sa *Dialogflow*.

Table 12. Broj korisnika i broj prepoznatih namera u odnosu na NLU servis

NLU Servis	RASA	Dialogflow	Total
Broj korisnika	3173	168	3341
Broj prepoznatih namera	81420	5554	86974

Izvor: ADA baza podataka

ADA četbot uči i postaje pametnija zahvaljujući veštačkoj inteligenciji. S obzirom da je učenje dugotrajan i težak process, softverski tim koji je implementirao ovaj projekat mora neprestano da monitoriše interakcije studenata sa četbotom, odnosno da prati šta i kako studenti pitaju. Tim NLU trenera konstantno uči ADA tako što se ubacuju nova scenarija i implementira nove usluge. Neke od novih usluga koje treba da se realizuju su: mogućnost promene izbornog predmeta, evidencija prisustva, lakše prenošenje znanja od strane nastavnika itd.

Na kraju ovog poglavlja treba navesti prednosti ADA četbota implementiranog u BAPUSS-u:

- Dostupnost 24/7/365 – Visoka dostupnost je obezbeđena svakim danom, jer svima odgovara da se pruži usluga van normalnog radnog vremena. Časkanje je dostupno dvadeset i četiri sata tokom cele godine;
- Ušteda vremena – Mogućnost istovremenog opsluživanja hiljada studenata i korisnika. Korak odlaska na šalter se preskače radi dobijanja određenih informacija. Četbot ceni i štedi vreme studenata i oslobađa više vremena koje mogu da iskoriste za učenje;
- Usluga je brža – Našim studentima je omogućeno da manje koriste standardne načine komunikacije (telefon i mejl) jer ih to prisiljava da više čekaju. ADA rešava mnogo brže jednostavne zahteve studenata;
- Akademija je u koraku sa trendovima – Zadatak ADA nije samo da pruži informacije, da olakša svakodnevne zadatke, već da uvede novu digitalnu strategiju koristeći elemente veštačke inteligencije;
- Laka komunikacija - Svako ko je korisnik Viber platforme ili FB Mesindžera, može veoma lako i jednostavno da komunicira putem časkanja;

- Istovremeno obavljanje više zadataka – Potreba za informacijama i uslugama od strane većeg broja studenata u odnosu na ograničenje ljudskih resursa BAPUSS-a je zadovoljena i optimizovana;
- Niži troškovi i veća efikasnost rada Akademije – savremena komunikacija sa studentima za Akademiju znači mogućnost bržeg i efikasnijeg sprovođenja njenih usluga po nižim troškovima.

8. POREĐENJE SA OSTALIM PRISTUPIMA

Tokom razvoja četbota evidentirani su specifični pristupi, arhitekture i rešenja koje su predložile različite organizacije kao što su: velike softverske kompanije, akademske zajednice, pojedine državne institucije razvijenih zemalja i međunarodna tela za standardizaciju. Predloženo je nekoliko pristupa da se pojednostavi proces razvoja četbota i da se poboljša njihov mehanizam odgovora [86, 87]. ALICE sistem i njegov AIML jezik su poboljšani u fazi usklađivanja uzorka i vidljivosti generisanja odgovora [38]. Jedan od ciljeva četbota zasnovanog na AIML jeziku bio je da odgovori na pitanja studenata u vezi sa naučenim materijalom [88]. Nijedan od njih nije labavo povezan, što znači da se njihove komponente ne mogu jednostavno zameniti ako se pojave bolja rešenja. Slabo povezana arhitektura je način upravljanja rizikom, posebno sa komponentama otvorenog koda.

U domenu proučavanja četbot tehnologija, poseban pravac interesovanja je vezan za okvir (engl. framework) modelovanja četbota. Okviri postojećih četbotova za časkanje pružaju specifične arhitekture i alate za časkanje. Neki su veoma jednostavni, a drugi koriste masivne lingvističke skupove podataka veb servisa koji imaju veoma precizne modele dobijene zahvaljujući dugogodišnjem iskustvu i ogromnim količinama podataka. Okvir za četbot je uglavnom realizovan definisanjem entiteta, namera i odgovora u okviru određene platforme, obezbeđujući interfejs i velike kapacitete razumevanja prirodnog jezika. Dostupne su mnoge komercijalne opcije i opcije otvorenog koda za razvoj četbota. Broj tehnologija vezanih za četbot je već ogroman i svakim danom raste. Četbotovi se razvijaju na dva načina: korišćenjem bilo kog programskog jezika kao što su *C#*, *Java*, *Clojure*, *Python*, *C++*, *PHP*, *Rubi* i *Lisp* ili korišćenjem najsavremenijih platformi. Iako su prethodno već navedeni treba napomenuti da se trenutno izdvajaju šest vodećih NLU platformi u oblaku koje programeri mogu da koriste za kreiranje aplikacija koje mogu da razumeju prirodne jezike: *Google-ov DialogFlow*, *Facebook-ov WIT.ai*, *Microsoft-ov LUIS*, *IBM-ov Watson Conversation*, *RASA*, *Amazon Lex* i *SAP Conversation AI* [89]. Sve ove platforme podržavaju mašinsko učenje. Oni dele neku standardnu funkcionalnost (bazirani su na oblaku, podržavaju različite programske i prirodne jezike), ali se značajno

razlikuju u drugim aspektima [90]. Druge poznate platforme za razvoj četbota su Botsify [91], Chatfuel [92], Manychat [93], FlowXO [94], Chatterbot [95], Pandorabots [96], Botkit [97] i Botlytics [98].

Do skoro pre desetak godina se razvoj četbota uglavnom odvijao putem direktnog definisanja namera i odgovora četbota u okviru određene četbot platforme. Svi oni pružaju onlajn interfejs koji omogućava da se specificiraju namere korisnika, tok razgovora i kontekst informacija koje treba održavati tokom razgovora i nude odlične mogućnosti obrade prirodnog jezika. Međutim, svi oni imaju ograničene mogućnosti integracije sa drugim platformama. Svaki složeni odgovor četbota (osim samo davanja odgovora zasnovanog na tekstu) zahteva ručno kodiranje i upravljanje API, što ih čini neprikladnim za neprofesionalne programere. Okviri za kodiranje četbota kao što je pružaju skup programskih biblioteka za implementaciju i primenu četbot aplikacija. Oni obično olakšavaju integraciju sa mašinama za prepoznavanje namera i nekim platformama za razmenu poruka, ali i dalje zahtevaju ručnu integraciju tih i bilo koje druge eksterne usluge. Druge platforme kao što je Botkit ili neke platforme sa niskim kodiranjem [92, 99, 100] unapred definišu brojne integracije sa glavnom platformom za razmenu poruka. Iako ovo pomaže ako je cilj napraviti jednostavne konverzacione četbotove, oni i dalje moraju da budu povezani na jednu od gore navedenih platformi (za moćne jezičke veštine, npr. za prepoznavanje namere) i zahtevaju, kao i ranije, ručno kodiranje naprednih četbot akcija. Nekoliko platformi sa niskim kodiranjem kao što je FlowXO [94] takođe pružaju podršku za pokretanje akcija unutar konverzacija. Međutim, oni su tipično definisane kao zatvoreno okruženje koje dizajner ne može lako da proširi novim akcijama i/ili platformama za prepoznavanje namera.

Velika studija o vrstama arhitektura četbota je napisana u [1]. U ovom i drugim sličnim radovima [6, 33, 57, 87, 101-108] niko ne objašnjava kako prevazići problem integracije sa različitim NLU servisima ili sa različitim komunikacionim platformama. Veoma je teško integrisati arhitekturu četbota sa drugim eksternim servisima bez ručnog kodiranja. Konkretno, veoma je teško proširiti četbot arhitekturu sa novim komunikacionim platformama ili novim NLU servisima [101]. Većina četbot arhitektura samo nudi API za ispitivanje rezultata namere i integraciju sa NLU servisom. Slično tome, iako svaki od njih podržava različite kanale komunikacije, obično ne nude nikakve

opcije proširenja. Takođe, ne objašnjava se način prelaska sa jednog NLU na drugi NLU servis ili sa jednog na drugi komunikacioni kanal.

Međutim, predložena su svega dva slična pristupa da se pojednostavi proces razvoja četbota i da se poboljša njihov mehanizam odgovora [101, 109, 110]. Xatkit [109], ranije poznat kao Jarvis [101], i Conga [110] su delimično slični predloženom rešenju. Oni koriste arhitekturu vođenu modelom (engl. Model Driven Development - MDA) za razvoj četbota. Iako se naši pristupi i arhitekture četbota razlikuju od ova dva rada, oni iznose realizaciju ideje o razvoju četbota koji su nezavisni od NLU servisa. Međutim, ovde predložena arhitektura ne koristi MDA za razvoj četbota poput njih, već koristi jedan od osnovnih koncepata MDA pristupa (modele). Da bi se izbegla zavisnost od dobavljača NLU servisa, predložena arhitektura četbota se sastoji od metamodela za NLU usluge, i oni se mapiraju na opšta pravila mapiranja NLU metamodela. Ovde je lako primetiti da se koristi samo NLU, koji je jedan deo obrade prirodnog jezika, i da metodologija koja se predlaže u ovom radu ima drugačiju logiku upotrebe konteksta i akcije izvršavanja. Njihovi metamodeli su veći sa različitim svrhom od našeg, jer uključuju koncepte kao što su kontekst, akcija, platforme itd. U predloženom rešenju se koristi jednostavna tehnika koja omogućava da se obezbedi proširljiva NLU nezavisna arhitektura četbota. U ovde predloženoj arhitekturi, sva logika koja se odnosi na kontekst koji utiče na tok razgovora ili na izvršenje određene radnje nalazi se u komponentama koje su već objašnjene. Međutim, osim detaljnog objašnjenja mikroservisne arhitekture četbota, glavni doprinos i razlika ove arhitekture je laka integracija sa NLU eksternim servisom jer je to verovatno najvažniji deo svake arhitekture četbota koji je teško implementirati samostalno. Zbog toga je dobro koristiti eksterne usluge. Ali predložena arhitektura ne isključuje mogućnost stvaranja sopstvenog NLU servisa za koju su potrebni ogromni resursi i znanje. Svaki novi NLU servis se može integrisati na isti jednostavan način kao što je prikazano u ovoj tezi, bez ručnog programiranja zahvaljujući metamodelima i predloženim pravilima mapiranja.

Xatkit predstavlja multi-platformski okvir za modelovanje četbota koji koristi metamodele i tekstualni jezik specifičnog domena (DSL). Ovaj okvir razdvaja deo modelovanje četbota od aspekata specifičnih za platformu, povećavajući mogućnost ponovne upotrebe. DSL obezbeđuje primitive za dizajniranje namera korisnika, logike

izvršenja i platforme za primenu. Trenutno podržavaju samo *Dialogflow* kao NLU servis. Prikazana implementacija u ovoj tezi je pokazala nezavisnost od usluga *Dialogflow* i RASA NLU. U ovde predloženom pristupu takođe se daje uslovno grananje i podržavaju se generički događaji tzv. vebhuk (engl. webhooks) koji omogućavaju da se napravi proaktivni četbot koji može aktivno da sluša i odgovara. Predloženi okvir poboljšava proširljivost i prilagodljivost obezbeđujući eksplicitne metode vebhuka i elemente arhitekture koji mu omogućavaju da proširi svoj stabilan interfejs. S druge strane, na njihovom metamodelu se ne vidi kako oni označavaju uloge entiteta i grupisanja entiteta. Pomoću uloga entiteta mogu se definisati entiteti sa specifičnim ulogama u frazi. Grupisanje entiteta omogućava da se entiteti grupišu zajedno sa posebnom oznakom grupe koja definiše različite redoslede. Iako je Xatkit zasnovan na modelima, on ne smanjuje rizik od zavisnosti proizvođača i ne daje podršku za migraciju četbota.

CONGA predstavlja veoma apstraktno rešenje zasnovano na modelu za reverzibilan inženjeringu četbota, koji sadrži sistem preporuka koji pomaže u odabiru najprikladnijih alata za razvoj četbota. Sastoji se od neutralnog metamodela i DSL-a za generisanje koda i parsera za nekoliko četbot platformi. Glavna i najvažnija razlika u odnosu na CONGA je u tome što je u ovoj tezi predložena arhitektura izvršna, tako što obezbeđuju mehanizam za izvršavanje. CONGA takođe ne podržava koncepte specifične za platformu, kao što su dugmići koji prouzrokuju neku akcije. Njihov editor nije namenjen za veb okruženje. Iako u celom njihovom metamodelu takođe nedostaju elementi uloge i grupe, teško je uporediti ostatak metamodela jer je namena njihovih metamodela drugačija od ovde predloženih metamodela. Iako MDA ima mnogo prednosti u razvoju softvera, takav razvoj je često ograničen vrstom alata koji koriste. Conga već ima 15 alata za generisanje. Oni su fleksibilni samo u delovima okvira koje pokriva korišćeni DSL.

U tabeli 13 se daje poređenje opšte poznate arhitekture razvoja četbota sa posebnim osvrtom na pristup adaptivnog četbota koji je lako prilagodljiv različitim NLU servisima i komunikacionim kanalima. Poređenje je urađeno prvenstveno imajući u vidu opšte karakteristike i komponente arhitekture za razvoj četbota.

Tabela 13. Komparacija relevantnih radova

Referenca	Apstraktna arhitektura	Detaljna arhitektura	Odgovr baziran na pravilima	Odgovr baziran na pronalazenju	Odgovr baziran na generisanju	Komponenta za konektora poruka	Analizu poruke (NLU)	Provera pravonisa	Analiza osecanja	Upravljanje dijalogom	Obrada dvosmislenosti	Obrada grešaka	Eksterni API konektori	Komponenta za generisanje odgovora	Podrška za migraciju	Fleksibilnost NLU i komunikacionih kanala
[111]	+	-	+	-	-	+	-	-	-	-	+	+	+	-	-	-
[112]	-	+	-	+	-	-	-	-	-	-	-	-	-	+	-	-
[113]	-	+	-	+	-	-	+	-	-	+	-	-	+	+	-	-
[108]	-	+	+	+	+	+	+	-	+	+	-	-	+	+	-	-
[57]	+	-	+	+	+	-	+	-	+	+	+	+	-	+	-	-
[114]	-	+	-	-	+	-	-	-	-	-	-	-	-	+	-	-
[115]	-	+	+	-	-	-	+	-	-	+	-	-	+	+	-	-
[116]	-	+	+	-	-	+	+	-	-	+	-	-	+	+	-	-
[117]	-	+	-	+	+	+	+	-	-	+	-	-	+	+	-	-
[118]	-	+	-	+	+	+	+	+	+	+	+	+	+	+	-	-
[119]	-	+	-	+	+	+	+	-	-	+	-	-	+	+	-	-
[109]	-	+	-	+	+	-	+	-	-	+	+	+	-	-	-	+
[110]	-	+	-	+	+	-	+	-	-	+	+	+	+	-	+	+
[Predložen pristup]	-	+	-	+	+	+	+	+	-	+	+	+	+	+	+	+

U [111], autori predlažu arhitekturu, ali je specifična za četbote zasnovane na pravilima. Slično, [112] predstavljaju arhitekturu koja je specifična za četbote zasnovane na pronalazenju. U [113], baza znanja četbota je povezana sa drugim bazama podataka i informacionim sistemima koji daju odgovore na upite korisnika. Međutim, nedostaju vitalne usluge kao što su analiza osećanja i rukovanje dvosmislenosti. Zanimljiv dizajn [108] predstavlja arhitekturu strukturiranu u slojevima. [57] ne prikazuje detaljnu arhitekturu iako ima dosta komponentu neophodnih za razvoj četbota. Međutim, prilično je apstraktan jer ne pruža bitne detalje za svaki sloj. Takođe, ne govori o tome kako se analizira unos korisnika i koje su komponente upravljanja dijalogom. U [114], predlaže se da četbot generiše emocionalno dosledne odgovore, ali dizajn arhitekture se fokusira samo na modul za generisanje odgovora. Ham takođe radi na arhitekturi [115], ali ne daje nikakav arhitekturni dizajn. Arhitektura u [116] služi integraciji velikih podataka sa bazom znanja četbota. Međutim, to se odnosi samo na modele zasnovane na pravilima implementiranim sa AImL-om. U [117], predstavljen je detaljan arhitekturni projekat za četbot koji podržava studente. Ovom dizajnu nedostaje komponenta analize osećanja. U [118], predstavljena je detaljna arhitektura *Microsoft XiaoIce*. Ona uključuje mnoge

zanimljive karakteristike, ali je specifična za *XiaoIce*. U [119], istraživanje se fokusira na slojeve kampusa i kako su oni povezani sa četbotom, ali ne daje arhitekturni dizajn četbota. Xatkit i CONGA koriste arhitekturu vođenu modelom za razvoj četbota i slična su ovde predloženom rešenju razvoja četbota. Iako se predloženi pristup i arhitektura četbota razlikuju od ova dva rada, oni iznose realizaciju ideje o razvoju četbota koji su nezavisni od NLU servisa. Pristup predložen u ovoj tezi je drugačiji jer je zasnovan na metamodelima koristeći mikroservisnu arhitekturu zajedno sa modelovanjem kontekstom. Korišćenje okvira metamodela predloženih u ovoj tezi je dovoljno proširljivo i prilagodljivo da uključi nove NLU servise i komunikacione kanale. Predložena arhitektura pruža interaktivno učenje koje olakšava programerima razvoj tokova razgovora.

9. ZAKLJUČAK

Mnogi korisnici digitalnih komunikacija najviše vremena provode u korišćenju komunikacionih platformi za razmenu poruka. Četbot je softver koji služi kao korisnički interfejs koji koristi prirodni jezik za potrebe dobijanja podataka ili obavljanja usluga. KOVID-19 je ubrzao potrebu za rešenjima za časkanje. Nakon pandemije, upotreba četbota za informativne i poslovne aplikacije nastaviće da raste. On predstavlja računarski program koji simulira konverzaciju koristeći veštačku inteligenciju i učestvuje u dijalogu sa čovekom koristeći prirodni jezik. Četbotovi mogu da dosegnu široku publiku kroz komunikacione platforme i da budu efikasniji od ljudi. U isto vreme, oni se mogu razviti u sposoban alat za prikupljanje i davanje informacija pri tome omogućavajući značajne uštede najviše u radu korisničke podrške, a zatim i u ostalim segmentima poslovanja.

Tehnološki giganti su masovno počeli da integrišu četbota unutar komunikacionih platformi kako bi pomogli korisniku da jednostavno dobije odgovor ili da mu se ispune odgovarajuće osnovne usluge. Na osnovu prirodne obrade rečenice uz pomoć veštačke inteligencije i primenu odgovarajućeg algoritma, četbot dolazi do adekvatnog odgovora koji se prosleđuje korisniku. Sa daljim razvojem veštačke inteligencije i mašinskog učenja, vrlo je verovatno da korisnici neće moći biti u stanju da prepoznaju da li razgovaraju sa četbotom ili agentom iz stvarnog života. A to je u krajnjem slučaju i nepotrebno, jer je cilj da se uspešno ostvari zahtev krajnjeg korisnika. Minimalno ljudska interakcija i mešanje u upotrebu pametnih uređaja je cilj u ovom digitalnom svetu.

Četbot postaje sve više interesantan jer postoje sve veća očekivanja inteligentnih digitalnih asistenata da će uvek biti na raspolaganju za brzo i efikasno rešavanje zahteva klijenata. Četbot je odličan alat da sve organizacije ostanu povezane sa klijentima i da sa više pažnje osluškuju i rešavaju sve njihove zahteve, od dobijanja jednostavnih odgovora do izvršavanja komplikovanih usluga. Četbotovi su kao nove aplikacije, a digitalni pomoćnici su meta aplikacije ili kao novi pregledači. Osnovna prednost implementacije četbota je uvođenje nove digitalne strategije koristeći elemente veštačke inteligencije kroz uštedu, automatizaciju i optimizaciju.

Polazeći od ovih konstatacija, disertacija je imala hipotezu da se razvojem adaptivnog četbota zasnovanog na veštačkoj inteligenciji može postići automatizacija velikog broja poslovnih procesa i brža obrada velikog broja zahteva i na taj način se povećati produktivnost kompanija. U radu je predložena originalna metodologija zajedno sa metamodelima kao preduslova za adaptacioni mehanizam. Predloženi pristup razvoja četbota daje mogućnost adaptivnog ponašanja četbota u odnosu na stare, nove i modifikovane NLU servise i komunikacione platforme bez intervencije korisnika. Na taj način se umanjuju njihove interakcije i prilagođeno programiranje, ali se daje nezavisnost od proizvođača ovih servisa i platformi. Istaknuti su osnovni logički delovi opšte četbot arhitekture, kao i način adaptacije različitih konkretnih četbotova. Takođe je prezentovano da predloženi dvostruki životni ciklus mora da pruži podršku i u vreme izvršavanja. Ovaj pristup je omogućio da se u vreme projektovanja koriste apstraktne tehnike modelovanja. U skladu sa zahtevima je moguće jednostavno adaptirati automatske transformacije modela kako bi se izgenerisao izvršni kôd i omogućeno je brže prilagođavanje promenama kao jednim od najbitnijih zahteva savremenih informacionih sistema. Na osnovu svega prethodno navedenog, ovde predloženi originalni pristup je imao cilj da se omogući visok stepen prilagođavanja četbota, veća produktivnost i efikasnost tokom generisanja i održavanja četbota.

U ovom radu upoznali smo se bliže sa četbotovima, predstavili njihove tipove i generičku arhitekturu za četbot frejmwork. Predložena platforma i arhitektura je fleksibilna, sama po sebi skalabilna i podržava različite medije za interakcije korisnika. Teza prvenstveno teži ka naučnom doprinosu u oblasti razvoja adaptivnog četbota. Ključni naučni doprinosi su:

- Razvoj metodološkog pristupa za izgradnju adaptivnog četbota zasnovanog na veštačkoj inteligenciji.
- Specifikacija jedinstvene logičke arhitekture za praktičnu primenu predloženog pristupa u implementaciji složenog konverzionog četbota.
- Definisane aplikativnog okvira (engl. framework) koji treba da omogući specificiranja bilo koje definicije poslovnog procesa (engl. workflow) zajedno sa jasno definisanim modelom konteksta.

- Definisani opšti postupak i pravila mapiranja modela između elemenata metamodela opšteg (generičkog) i konkretnog metamodela servisa za razumevanje prirodnog jezika koji četbot u vreme njenog izvršavanja koristi, u cilju prilagođavanja četbota novim servisima, odnosno pružanju nezavisnosti od samo jednog servisa. Na ovaj način se stiče nezavisnost od bilo kog servisa za razumevanje prirodnog jezika, čime se rešenje neće vezati za jedan servis već će se predloženim mehanizmom obezbediti lako mapiranje i prebacivanje sa jednog na drugi novi i/ili stari servis.
- Pregled najpopularnijih servisa (platformi) za razumevanje prirodnog jezika, kao i opis detaljne specifikacije procesa obrade i komunikacije sa drugim sistemima.
- Nezavisna komunikacija sa eksternim poslovnim servisima kao što su npr.: poslovni ili bankarski sistemi, kontakt centri, itd.
- Proces dinamičke adaptacije generisanjem modela mapiranja na osnovu metamodela i pravila mapiranja.
- Date su originalne verzije metamodela opšteg (generičkog) NLU servisa kao i dva najpoznatija NLU servisa svetski poznatih NLP platformi. Glavni razlog za razvoj sopstvene verzije metamodela je da se omogući jednostavna veza između koncepata najpoznatijih NLU platformi.
- U radu je definisana metodologija razvoja četbota ilustrovana kroz slučajeve korišćenja.

Stručni i praktični doprinosi:

- Za lakše prevazilaženje nezavisnosti četbota od NLU servisa urađen je model preslikavanja između opšteg i specifičnog metamodela NLU servisa koji prikazuje detaljnije mapiranje između elemenata ova dva metamodela.
- Ilustrovana je primena predloženog pristupa kroz fazu aplikacionog inženjeringa u izabranom domenu poslovanja. Da bi ostala konkurentna, Beogradska akademija odlučila je da stvori takvo okruženje za učenje za svoje studente kako bi digitalizacija postala alat koji proširuje i dopunjuje odnos student-profesor. Zahvaljujući brzim odgovorima dobijenim od ADE, trenutni i budući studenti postaju dobro informisani i spremniji da se uključe u interaktivnu komunikaciju sa profesorima i u procesu zajedničkog rada, učenja, istraživanja i saradnje. S druge strane, prateći njihova pitanja, rezultate ispita, odabrane sadržaje, BAPUSS bolje upoznaje svoje krajnje

korisnike i dolazi u poziciju da im ponudi sisteme učenja koji će im više odgovarati i koji su bolje prilagođeni ličnim stilovima učenja sa odličnim preciznostima (personalizacija obrazovnih sadržaja).

- Rešenje je implementirano u konkretnu studiju slučaja na srpskom jeziku prikazujući kreiranje četbota koje je nezavisno od jednog NLU servisa. Lekcije naučene iz studije slučaja su pokazale da je veb API bio dobar izbor između ostalog jer omogućava RESTfull servis i JSON objekte i jer može biti hostovan unutar IIS ili same aplikacije.

Vođeni obećanjem inteligentnih digitalnih asistenata da će uvek biti na raspolaganju za brzo i dosledno rešavanje zahteva klijenata, četbotovi postaju sve korisniji. Jedan od glavnih ciljeva je bio da se napravi agilna i proširljiva mikroservisna arhitektura četbota, koja će upravljati veoma rastućem i promenljivim okruženjem modernih platformi i tehnologija. Prednosti predložene arhitekture četbota:

- Može se proširiti i daje podršku za svaki novi NLU servis i komunikacioni kanal za razmenu poruka.
- Oslanja se na metamodele kao glavne proširive mehanizme. Za ovu karakteristiku obezbeđeni su:
 - Opšti (generički) NLU metamodel.
 - Metamodeli za dva specifična NLU servisa (*Dialogflow* i RASA).
 - Odgovarajući metamodeli kao i pravila za mapiranje između generičkog i specifičnih NLU metamodela.
- Krojena je i prilagodljiva u odnosu na sposobnost arhitekture da njome upravlja i prilagođava projektant (modelator). Predložena softverska arhitektura se može proširiti, adaptirati, upravljati i prilagoditi kako bi se zadovoljile buduće promene i tehnologije četbota.
- U osnovi je skalabilna zahvaljujući mikroservisima. Predstavljena je napredna arhitektura četbot okvira izgrađena na mikroservisima, odnosno API servisnoj arhitekturi.
- Pruža zrelu funkcionalnost u vreme izvršavanja unutar domena četbotova.
- Obezbeđuje radni tok koji predstavlja detaljne i dinamične poslovne procese (scenarije) koji se tiču redosleda aktivnosti da bi se izvršile namere.

- Korisnički interfejs za NLU učenje je definisan na jednom mestu bez obzira na različite eksterne NLU servise.
- Definicije scenarija su takođe definisane na jednom mestu bez obzira na različite komunikacione platforme. Prelazak sa jednog na više NLU-a ili uvođenje novog NLU-a je lako postići u nekoliko klikova zbog metamodela i niske povezanosti osnovnih delova arhitekture. Dodatna prednost je što četbot platforma ne mora da brine o korisničkom interfejsu komunikacionih platformi. Održavanje je mnogo lakše jer korisnički interfejs održava komunikaciona platforma (Viber, FB messenger, itd.).

Sve prethodno navedene prednosti smanjuju krivu učenja za korisnika i pomaže u boljem razumevanju korisničke poruke.

Koliko su uslovi pandemije ugrožavali obrazovni sistem, toliko su ga menjali i otvarali mogućnosti za drugačije, naprednije i efikasnije metode učenja i prenošenja znanja. Digitalne tehnologije nude potpuno nove odgovore na pitanje šta ljudi uče, gde, kada i na koji način uče. Na osnovu prikazane statističke analize uočava se opravdanost uvođenja četbota kao pomoćnog alata za pružanje usluga i komunikacije sa studentima. Statistički parametri korišćenja ADE i analiza statistike konverzacije po kanalima pokazuju rastući trend upotrebe četbota u BAPUSS-u. Iz ovoga se može zaključiti da bi svaka obrazovna ustanova, pored privrede, trebalo da ima i svog četbtoa.

9.1. PRAVCI DALJEG ISTRAŽIVANJA

Iako je dosta problema rešeno, postoji još detalja koji treba da se predlože i implementiraju. Postupak treninga zahteva više strpljenja, vremena i znanja iz domena poslovanja. Bez obzira na prednosti uvođenja četbota uvek postoji potreba za ljudskom intervencijom. Teza je otvorila nove pravce istraživanja u smeru daljeg formalnog usavršavanja metodologije. Istraživački i praktičan rad se nastavlja u nekoliko pravaca:

- U budućim istraživanjima potrebno je da se detaljno potvrdi mogućnost istovremenog korišćenja NLU servisa i prihvatanja najboljeg rezultata prepoznavanja namere.
- Treba nastaviti u pravcu potpune automatizacije i proširenja četbota sa novim mogućnostima, kao na primer automatsko izvlačenje osećanja korisnika iz komunikacije.
- Trebalo bi se omogućiti glasovne poruke i potrebno je obezbediti dodatna merenja performansi kako bi se identifikovala uska grla korišćenjem tehnika evaluacije kao što su [102, 103]. Predložen pristup i arhitektura četbota predstavljena u ovom radu je odlična polazna tačka za ove istraživačke pravce.
- Potrebno je po istom metodološkom pristupu definisati opšti postupak i pravila mapiranja modela između elemenata metamodela opšteg (generičkog) i konkretnog metamodela komunikacione platforme za razmenu poruka (kanal) koji četbot u vreme njenog izvršavanja koristi, u cilju prilagođavanja četbota novim kanalima, odnosno pružanju nezavisnosti od samo jednog kanala. Na ovaj način se stiče nezavisnost od bilo kog komunikacionog kanala, čime se rešenje neće vezati za jedan kanal već će se predloženim mehanizmom obezbediti lako mapiranje i prebacivanje sa jednog na drugi novi i/ili stari kanal. Nezavisnost od komunikacionih kanala treba da se definiše kroz tri glavna dela: (1) generički metamodel koji definiše opšte koncepte svih komunikacionih kanala i njihovih odnosa, (2) generički metamodel specifičnog komunikacionog kanala (npr. FB mesindžer, Viber) i (3) skupa pravila mapiranja koji preslikavaju svaki koncept specifičnog komunikacionog kanala u koncept generičkog metamodela.

- Potrebno je ostvarite potpunu personalizaciju kroz profilisanje korisnika i mehanizme svesti o kontekstu. Pojava nedavnih tehnologija zasnovanih na veštačkoj inteligenciji kao što su rekurentne neuronske mreže i nedavni primeri četbotova koji su svesni konteksta kroz proširenje obima razgovora su osnova za buduća istraživanja za postizanje potpune personalizacije.
- U praktičnom smislu u toku je nekoliko dodatnih implementacija u oblasti obrazovanja, kao i za podršku rada državnih organa. Jedan od pravaca daljeg praktičnog rada je da se ADA četbot osposobi i nauči da bude pomoćno sredstvo koje će studentima omogućiti proučavanje gradiva predviđenog nastavnim planom i programom predmeta.
- Iako trenutno postoji jednostavno rešenje reagovanja četbota na nepredviđene promene situacije u okruženju, jedan od pravaca budućeg istraživanja razvoja četbota su upravo te adaptacije na nepoznate i nepredviđene događaje.
- S obzirom da je ovde predloženi pristup baziran na kreiranju kategorija četbotova potrebno je proširiti arhitekturu tako da ona može da obezbedi komunikaciju između njih. Ideja da se napravi četbot za svaki predmet dovodi do potrebe njihove međusobne interakcije. ADA bi trebala da bude nadređeni četbot koja zna kog četbota da pozove kad student odluči da sprema (uči) određeni predmet. Svaki nastavnik kreira sopstveni sadržaj i samostalno obučava svoj četbot. Ovi nezavisni četbotovi bi bili povezani preko ADE zajedno sa dodatnim inteligentnim slojem. Potrebna je saradnja između nastavnika, ali mnogo manja nego ako bi se celokupno kreiranje, učenje i obuka odvijala u jednom četbotu.
- Četbot je dobar izbor za one sisteme koji imaju veliki broj upita, ali zbog troškova učenja i uvođenja četbota nije prikladan za manje organizacione sisteme. S tim u vezi potrebno je definisati i generisati kategoriju tzv. lajt verzije četbota sa manje funkcionalnosti, koja može da pokrije potrebe i zahteve manjih organizacija i koja je finansijski dostupnija.
- S obzirom da broj kategorija četbota eksponencijalno raste dodavanjem novih slučajeva korišćenja, postavlja se pitanje performansi i skalabilnosti. Takođe, u razvoju četbota treba razmotriti problem učestalog učenja koje se okida u vreme izvršavanja različitih

konkretnih četbotova. Sve ovo ima uticaj na nefunkcionalne zahteve i performanse konkretnog četbota i zato ovaj problem može biti jedan od budućih pravaca daljeg istraživanja.

Rezultati koji su dosada postignuti u razvoju četbota pokazuju da je obezbeđena potvrda dualnog životnog ciklusa. Prateći buduće pravce istraživanja apsolutno rastuće paradigme četbota očekivanja su optimistična. Međutim, kvalitet koji korisnici percipiraju je ključni izazov istraživanja. Navedeni istraživački izazovi su vezani za stepen do kog korisnici percipiraju četbotove kao korisne alate. Budućnost evolucije prodornosti četbotova uglavnom je uslovljena uvidom i doprinosima budućih istraživanja u smislu poboljšanja korisničkog iskustva kroz napredan, efikasan i precizno pametan proces konverzacije.

10. LITERATURA

- [1] Adamopoulou, E.; Moussiades, L. Chatbots: History, technology, and applications. *Machine Learning with Applications*, Volume 2, 100006, ISSN 2666-8270, <https://doi.org/10.1016/j.mlwa.2020.100006>. (2020).
- [2] Rodríguez Cardona, D.; Werth, O.; Schönborn, S.; Breitner, M.H. A mixed methods analysis of the adoption and diffusion of Chatbot Technology in the German insurance sector (2019)
- [3] Dale, R. The return of the chatbots. *Natural Language Engineering*, 22, 811–817. (2016)
- [4] J. Weizenbaum, ELIZA—a computer program for the study of natural language communication between man and machine, *Communications of the ACM*, v.9 n.1, p.36-45, (Jan. 1966)
- [5] Salla Syvänen and Chiara Valentini. 2020. Conversational agents in online organization–stakeholder interactions: a state-of-the-art analysis and implications for further research. *Int. J. Inf. Commun. Technol. Educ.* 24, 4 (Jan. 2020)
- [6] Motger, Q., Franch, X., & Marco, J. *Conversational Agents in Software Engineering: Survey, Taxonomy and Challenges*. 2021. ArXiv, abs/2106.10901.
- [7] Robert Dale. 2016. The return of the chatbots. *Natural Language Engineering* 22 (Sept. 2016)
- [8] Gina Neff and Peter Nagy. 2016. Talking to Bots: Symbiotic Agency and the Case of Tay. *International Journal of Communication* 10 (Okt. 2016)
- [9] Asbjørn Følstad and Petter Brandtzaeg. 2017. Chatbots and the new world of HCI. *interactions* 24 (Juni 2017)
- [10] David Griol, Araceli Sanchis, José Manuel Molina, and Zoraida Callejas. 2019. Developing enhanced conversational agents for social virtual worlds. *Neurocomputing* 354 (2019)
- [11] Mikko Riikkinen, Hannu Saarijärvi, Peter Sarlin, and Ilkka Lähteenmäki. 2018. Using artificial intelligence to create value in insurance. *International Journal of Bank Marketing* 36 (06 2018)

- [12] D. Niculescu and N. Badri, Ad hoc positioning system (APS) using AOA, Proc. 22th Joint IEEE Computer Communications Soc. (INFOCOM'03), vol. 3, pp. 1734-1743, Mar. 30-Apr. 3, (2003)
- [13] Zhao Yan, Nan Duan, Peng Chen, Ming Zhou, Jianshe Zhou, and Zhoujun Li. 2017. Building Task-Oriented Dialogue Systems for Online Shopping. AAAI 31, 1 (Feb. 2017)
- [14] Eva Bittner, Sarah Oeste-Reiß, and Jan Marco Leimeister. Where is the bot in our team? Toward a taxonomy of design option combinations for conversational agents in collaborative work. In Proceedings of the 52nd Hawaii international conference on system sciences. (2019)
- [15] Chun Hsieh and Daniel Buehrer. The Implementation of an Artificially Intelligent Personal Assistant for a Personal Computer. Applied Mechanics and Materials 627 (Sept. 2014)
- [16] Allan de Barcelos Silva, Marcio Miguel Gomes, Cristiano André da Costa, Rodrigo da Rosa Righi, Jorge Luis Victoria Barbosa, Gustavo Pessin, Geert De Doncker, and Gustavo Federizzi. Intelligent personal assistants: A systematic literature review. Expert Syst. Appl. 147 (June 2020)
- [17] João Santos, Joel J. P. C. Rodrigues, Bruno M. C. Silva, João Casal, Kashif Saleem, and Victor Denisov. An IoT-based mobile gateway for intelligent personal assistants on mobile health environments. Journal of Network and Computer Applications 71 (Aug. 2016)
- [18] Anna A. Allen, Howard C. Shane, and RalfW Schlosser. The Echo™ as a Speaker-Independent Speech Recognition Device to Support Children with Autism: an Exploratory Study. Advances in Neurodevelopmental Disorders 2, 1 (2018)
- [19] J. Todorov, S. Stoyanov, V. Valkanov, B. Daskalov, and I. Popchev. Learning Intelligent System for Student Assistance - LISSA. In 2016 IEEE 8th International Conference on Intelligent Systems (IS). (2016)
- [20] Raja M Suleman, Riichiro Mizoguchi, and Mitsuru Ikeda. 2016. A New Perspective of Negotiation-Based Dialog to Enhance Metacognitive Skills in the Context of Open Learner Models. International Journal of Artificial Intelligence in Education 26, 4 (Dec. 2016)

- [21] Miloš Kabiljo, M. Vidas-Bubanja, Rade Matić, Miodrag Živković, Education System in the Republic of Serbia under COVID-19 Conditions: Chatbot-Academic Digital Assistant of the Belgrade Business and Arts Academy for Applied Studies, Knowledge International Journal, 43(1), pg. 25 - 30. Retrieved from <http://ikm.mk/ojs/index.php/KIJ/article/view/4855> ISSN: 187-923X (print) i ISSN 2545-4439 (2020)
- [22] Jean-Emmanuel Bibault, Benjamin Chaix, Arthur Guillemassé, Sophie Cousin, Alexandre Escande, Morgane Perrin, Arthur Pienkowski, Guillaume Delamon, Pierre Nectoux, and Benoît Brouard. 2019. A Chatbot Versus Physicians to Provide Information for Patients With Breast Cancer: Blind, Randomized Controlled Noninferiority Trial. *J. Med. Internet Res.* 21, 11 (Nov. 2019)
- [23] Luis Fernando D'Haro, Seokhwan Kim, and Rafael E. Banchs. A robust spoken Q&A system with scarce in-domain resources. In 2015 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA). (2015)
- [24] Ela Elsholz, Jon Chamberlain, and Udo Kruschwitz. Exploring Language Style in Chatbots to Increase Perceived Product Value and User Engagement. (2019)
- [25] S. Jusoh, "Intelligent Conversational Agent for Online Sales", 2018 10th International Conference on Electronics Computers and Artificial Intelligence (ECAI), (2018)
- [26] José Quiroga Pérez, Thanasis Daradoumis, and Joan Manuel Marquès Puig. 2020. Rediscovering the use of chatbots in education: A systematic literature review. *Comput. Appl. Eng. Educ.* 28, 6 (Nov. 2020)
- [27] Hyekyun Rhee, James Allen, Jennifer Mammen, and Mary Swift. 2014. Mobile phone-based asthma self-management aid for adolescents (mASMAA): A feasibility study. *Patient preference and adherence* 8 (Jan. 2014)
- [28] IMF. (2020). World Economic Outlook: A Long and Difficult Ascent, Washington, (October 2020)
- [29] Gartner: <https://www.gartner.com/en/newsroom/press-releases/2018-02-19-gartner-says-25-percent-of-customer-service-operations-will-use-virtual-customer-assistants-by-2020> (datum pristupa 01.02.2018. godine).

- [30] Chatbot Report 2019: Global Trends and Analysis | by BRAIN [BRN.AI] CODE FOR EQUITY | Chatbots Magazine. (<https://chatbotsmagazine.com/chatbot-report-2019-global-trends-and-analysis-a487afec05b>). (datum pristupa 05.06.2021. godine)
- [31] Chatbot Market Size, Share | Industry Trends and Analysis by 2027. <https://www.alliedmarketresearch.com/chatbot-market>. (datum pristupa 05.06.2021. godine)
- [32] Turing, A. M. Computing machinery and intelligence. *Mind*, LIX(236), 433–460. <http://dx.doi.org/10.1093/mind/LIX.236.433>. (1950)
- [33] Smutny, P.; Schreiberova, P. Chatbots for learning: A review of educational chatbots for the Facebook Messenger. *Computers & Education*, 151, 103862. (2020)
- [34] Bansal, H., Khan, R.: A review paper on human computer interaction. *Int. J. Adv. Res. Comput. Sci. Softw. Eng.* 8, 53. <https://doi.org/10.23956/ijarsse.v8i4.630> (2018)
- [36] Colby, K.M., Weber, S., Hilf, F.D.: Artificial paranoia. *Artif. Intell.* 2, 1–25. [https://doi.org/10.1016/0004-3702\(71\)90002-6](https://doi.org/10.1016/0004-3702(71)90002-6) (1971)
- [37] Wallace, R.S.: The anatomy of A.L.I.C.E. In: Epstein, R., Roberts, G., Beber, G. (eds.) *Parsing the Turing Test: Philosophical and Methodological Issues in the Quest for the Thinking Computer*, pp. 181–210. Springer, Cham. https://doi.org/10.1007/978-1-4020-6710-5_13 (2009)
- [38] Marietto, M.d.G.B.; de Aguiar, R.V.; Barbosa, G.d.O.; Botelho, W.T.; Pimentel, E.; França, R.d.S.; da Silva, V.L. Artificial intelligence markup language: a brief tutorial. *arXiv preprint arXiv:1307.3091* (2013)
- [39] Molnár, G., Zoltán, S.: The role of chatbots in formal education. Presented at the 15 (September 2018)
- [40] Siri. <https://www.apple.com/siri/> (datum pristupa 25.10.2021. godine)
- [41] Google Assistant, your own personal Google. <https://assistant.google.com/> (datum pristupa 21.11.2021. godine)
- [42] What is Amazon's Alexa, and what it can do? <https://www.digitaltrends.com/home/what-is-amazons-alexa-and-what-can-it-do/> (datum pristupa 05.12.2021. godine)

- [43] Personal Digital Assistant - Cortana Home Assistant – Microsoft. <https://www.microsoft.com/en-us/cortana> (datum pristupa 14.12.2021. godine)
- [44] IBM Watson. <https://www.ibm.com/watson> (datum pristupa 15.12.2021. godine)
- [45] Scopus - Document search. <https://www.scopus.com/search/form.uri?display=basic> (datum pristupa 10.01.2018. godine)
- [46] Colace, F., De Santo, M., Lombardi, M., Pascale, F., Pietrosanto, A., Lemma, S.: Chatbot for e-learning: a case of study. *Int. J. Mech. Eng. Robot. Res.* 7, 528–533. <https://doi.org/10.18178/ijmerr.7.5.528-533> (2018)
- [47] E. Ferrara, O. Varol, C. Davis, F. Menczer, and A. Flammini, The rise of social bots, *Communications of the ACM*, vol. 59, no. 7, pp. 96–104, (2016)
- [48] Facebook Messenger Swelly. <https://www.messenger.com/t/swell.bot> (datum pristupa 12.03.2021. godine)
- [49] Shum, Hy., He, Xd. & Li, D. From Eliza to XiaoIce: challenges and opportunities with social chatbots. *Frontiers Inf Technol Electronic Eng* 19, 10–26. <https://doi.org/10.1631/FITEE.1700826> (2018)
- [50] Raiffeisen banka, „Ja sam REA“, <https://www.raiffeisenbank.rs/digitalne-usluge/rea-elektronska-asistentkinja/>, (datum pristupa 12.08.2021. godine)
- [51] Rade Matić, Miloš Kabiljo, Milen Janjić, Miodrag Živković, Milan Čabarkapa: ČETBOT – Digitalni asistent zasnovan na WEAVER platformi, *Info M* 70/2020, Serbia, Belgrade, (Januar 2020)
- [52] Weaver. <https://weaverbot.ai/>. (datum pristupa 30.01.2022. godine)
- [53] Ramesh, K., et al.: A Survey of Design Techniques for Conversational Agents. In: *International Conference on Information, Communication and Computing Technology*. Springer, Singapore (2017)
- [54] M. Nuruzzaman and O K Hussain. A Survey on Chatbot Implementation in Customer Service Industry through Deep Neural Networks. In *2018 IEEE 15th International Conference on e-Business Engineering (ICEBE)*. (2018)
- [55] Hien, H.T., Cuong, P.-N., Nam, L.N.H., Nhung, H.L.T.K., Thang, L.D.: Intelligent assistants in higher-education environments: the FIT-EBot, a chatbot

- for administrative and learning support. In: Proceedings of the Ninth International Symposium on Information and Communication Technology, pp. 69–76. ACM, New York (2018)
- [56] Mathur, S., & Lopez, D. A scaled-down neural conversational model for chatbots. *Concurrency Computations: Practice and Experience*, 31(10), Article e4761. <http://dx.doi.org/10.1002/cpe.4761>. (2019)
- [57] Nimavat, K., & Champaneria, T. Chatbots: An overview types, architecture, tools and future possibilities. *International Journal for Scientific Research and Development*, 5(7), 1019–1024, (2017)
- [58] Fayad, M.E.; Hamza, H.S.; Sanchez, H.A. Towards scalable and adaptable software architectures. IRI-2005 IEEE International Conference on Information Reuse and Integration, Conf. IEEE, 2005, pp. 102–107. (2005)
- [59] Abdellatif, A.; Badran, K.; Costa, D.; Shihab, E. A Comparison of Natural Language Understanding Platforms for Chatbots in Software Engineering. *IEEE Trans. Softw. Eng.* (2021)
- [60] R.Matić, Kontekstno orijentisani samoadaptivni sistemi zasnovani na dinamičkim softverskim proizvodnim linijama, doktorska disertacija, Beograd (Maj 2016)
- [61] S. Nešković, B. Lazarević, “Konceptija ostvarenja funkcionalnog i tehnološkog jedinstva informacionog sistema organa i organizacija SRJ”, studija, Savezni zavod za informatiku, Beograd (2001)
- [62] M. Karupiah, S. Kumari, X. Li, F. Wu, A.K. Das, M. K. Khan, R.Saravanan and S. Basu, A dynamic id-based generic framework for anonymous authentication scheme for roaming service in global mobility networks, *Wireless Personal Communications*, Vol.93, No.2, pp.383–407. (2016)
- [63] S. Cohen, D. Dori, U. de Haan, “A Software System Development Life Cycle Model for Improved Stakeholders’ Communication and Collaboration”, *International Journal of Computers, Communications & Control*, vol. 5, no. 1, (2010)
- [64] CH. Lin., W. Wolf, X. Koutsoukos, S. Neema, J. Sandeep,: “System and Software Architectures of Distributed Smart Cameras”, *ACM Transactions on Embedded Computing Systems (TECS)*. Vol. 9. No. 4, (2010)

- [65] Fayad, M.E.; Hamu, D.S.; Brugali, D. Enterprise frameworks characteristics, criteria, and challenges. *Communications of the ACM*, 43, 39–46. (2000)
- [66] Dialogflow, natural language understanding platform. <https://cloud.google.com/dialogflow/docs/>. (datum pristupa 30.01.2022. godine).
- [67] Open source conversational AI | RASA. <https://rasa.com/>. (datum pristupa 30.01.2022. godine)
- [68] LUIS (Language Understanding) – Cognitive Services – Microsoft Azure. <https://www.luis.ai/home> (datum pristupa 12.01.2022. godine)
- [69] Wit.ai. <https://wit.ai/> (datum pristupa 12.01.2022. godine)
- [70] C. Bolchini, C. A. Curino, E. Quintarelli, F. A. Schreiber, L. Tanca, “A data-oriented survey of context models”, *ACM SIGMOD Record*, vol.36 no.4, (Dec. 2007)
- [71] A.K. Dey, G.D. Abowd, “Towards a better understanding of context and contextawareness,” In the Workshop on the What, Who, Where, When and How of Context-Awareness, affiliated with the 2000 ACM Conference on Human Factors in Computer Systems, (2000)
- [72] C. Bettini, O. Brdiczka, K. Henriksen, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni, “A survey of context modeling and reasoning techniques”. *Pervasive and Mobile Computing*, vol.6 no.2, p.161-180, (2010)
- [73] A. Devaraju, S. Hoh, “Ontology-based Context Modeling for User-Centered Context-Aware Services Platform”, 3rd International Symposium on Information Technology, Kuala Lumpur, (2008)
- [74] A. Zimmermann, A. Lorenz, R. Oppermann, “An operational definition of context”, In the Proceedings of the Sixth International and Interdisciplinary Conference on Modeling and Using Context (CONTEXT 07), LNCS 4635:558–572, (2007)
- [75] V. Devedžić, “Understanding ontological engineering”, *Communications of the ACM*, vol.45 no.4, (April 2002)
- [76] M. Jarrar, J. Demey, R. Meersman, “On Using Conceptual Data Modeling for Ontology Engineering“, In Spaccapietra et al, (Eds.): *Journal on Data Semantics (Special issue on Best papers from the ER, ODBASE, and COOPIS 2002 Conferences)*. LNCS, vol. 2800, Springer, pp.:185-207. (October 2003)

- [77] K. K. Rajiv, Z. Hong, R. Ramesh, "A Helix-Spindle model for ontological engineering", *Communications of the ACM*, vol.47 no.2, p.69-75, (Feb. 2004)
- [78] D.M. Sanchez, J.M. Cavero, E.M. Martinez, "The road toward ontologies", In *ONTOLOGIES : A handbook of principles, concepts and applications in information systems*, London: Springer, pp. 3-20, (2006)
- [79] M. D. Fabro, J. Bézivin, F. Jouault, E. Breton, G. Guillaume. "AMW: A Generic Model Weaver". In *Ingénierie Dirigée par les Modèles (IDM 2005)*, (2005)
- [80] Neškovic, S.; Matic, R. Context modeling based on feature models expressed as views on ontologies via mappings. *Computer Science and Information Systems*, 12, 961–977. (2015)
- [81] Matic, R.; Kabiljo, M.; Zivkovic, M.; Cabarkapa, M. Extensible Chatbot Architecture Using Metamodels of Natural Language Understanding. *Electronics*, 10, 2300. <https://doi.org/10.3390/electronics10182300> (2021)
- [82] BAPUSS - Beogradska akademija poslovnih i umetničkih strukovnih studija. <https://www.bpa.edu.rs/>. (datum pristupa 04.12.2021. godine)
- [83] ADA - About me. <https://www.bpa.edu.rs/en-us/about-ada-chatbot> (datum pristupa 04.12.2021. godine)
- [84] ADA chatbot. <https://chatbot.bpa.edu.rs/> (datum pristupa 04.12.2021. godine)
- [85] Spacy. <https://spacy.io/> (datum pristupa 23.01.2022. godine)
- [86] A. Augello, G. Pilato, G. Vassallo, and S. Gaglio, "Chatbots as interface to ontologies." In *Advances onto the Internet of Things*, Springer International Publishing, pp. 285-299, (2014)
- [87] S. A. Abdul-Kader, and J. C. Woods, Survey on chatbot design techniques in speech conversation systems, *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 7, pp. 72-80, (2015)
- [88] F. A. Mikic, J. C. Llamas Burguillo, D. A. M. Rodríguez, and E. Rodríguez, "CHARLIE: An AIML-based Chatterbot which Works as an Interface among INES and Humans." *IEEE EAEEIE annual conference*, pp. 1-6, (2009)
- [89] SAP Conversational AI | Automate Customer Service With AI Chatbots. <https://cai.tools.sap> (datum pristupa 21.02.2022. godine)

- [90] Canonico, M., Russis, L.D.: A Comparison and Critique of Natural Language Understanding Tools. (2018)
- [91] Botsify: Botsify - Create Automated Chatbots Online for Facebook Messenger or Website. <https://botsify.com> (datum pristupa 11.02.2022. godine)
- [92] Chatfuel. <https://chatfuel.com/> (datum pristupa 11.02.2022. godine)
- [93] ManyChat – Chat Marketing Made Easy. <https://manychat.com/> (datum pristupa 11.02.2022. godine)
- [94] AI Online Chatbot Software, Live Chat on Websites. <https://flowxo.com/> (datum pristupa 11.02.2022. godine)
- [95] About ChatterBot — ChatterBot 1.0.2 documentation. <https://chatterbot.readthedocs.io/en/stable/> (datum pristupa 11.02.2022. godine)
- [96] Pandorabots: Home. <https://home.pandorabots.com/home.html> (datum pristupa 11.02.2022. godine)
- [97] Botkit: Building Blocks for Building Bots. <https://botkit.ai/> (datum pristupa 11.02.2022. godine)
- [98] Botlytics: <https://www.botlytics.co/> (datum pristupa 11.02.2022. godine)
- [99] Mendix: <https://www.mendix.com/>. (datum pristupa 14.02.2022. godine)
- [100] Smartloop: <https://smartloop.ai/>. (datum pristupa 14.02.2022. godine)
- [101] Daniel, G.; Cabot, J.; Deruelle, L.; Derras, M. Multi-platform chatbot modeling and deployment with the Jarvis framework. *International Conference on Advanced Information Systems Engineering*. Springer, pp. 177–193. (2019)
- [102] Radziwill, N.M.; Benton, M.C. Evaluating quality of chatbots and intelligent conversational agents. *arXiv preprint arXiv:1704.04579* (2017).
- [103] Pereira, J.; Díaz, O. A quality analysis of facebook messenger’s most popular chatbots. *Proceedings of the 33rd annual ACM symposium on applied computing*, pp. 2144–2150. (2018)
- [104] Jwala K., Sirisha G.N.V.G, Padma Raju G.V. Developing a Chatbot using Machine Learning. *International Journal of Recent Technology and Engineering (IJRTE) ISSN: 2277-3878, Vol. 8, pp. 89-92.* (2019)
- [105] Swanson, K., Yu, L., Fox, C., Wohlwend, J., Lei, T. Building a production model for retrieval-based chatbots. In: *Proceedings of the First Workshop on NLP for Conversational AI* pp. 32–41. (2019)

- [106] Peng, Z., Ma, X. A survey on construction and enhancement methods in service chatbots design. *CCF Transactions on Pervasive Computing and Interaction* pp. 204-223. (2019)
- [107] Hussain, S., Sianaki, O. A., Ababneh, N. A survey on conversational agents/chatbots classification and design techniques. *Web, Artificial Intelligence And Network Applications: Proceedings Of The Workshops Of The 33Rd International Conference On Advanced Information Networking And Applications*, Matsue, Japan, pp. 946-956. https://doi.org/10.1007/978-3-030-15035-8_93 (27-29 Mart 2019)
- [108] Khan, R. Standardized architecture for conversational agents a.k.a. chatbots. *International Journal of Computer Trends and Technology*, vol. 50, pp 114–121. doi.org/10.14445/22312803/IJCTT-V50P120. (2017)
- [109] Daniel G., Cabot J., Deruelle L., Derras M., Xatkit: A multimodal low-code chatbot development framework, *IEEE Access*, vol. 8, pp. 15332–15346. (2020)
- [110] Pérez-Soler S., Guerra E., de Lara J., Model-driven chatbot development, in *ER*, ser. LNCS, vol. 12400. Springer, pp. 207–222. (2020)
- [111] Khanna, A., Pandey, B., Vashishta, K., Kalia, K., Bhale, P., & Das, T. A study of today's A.I. through chatbots and rediscovery of machine intelligence. *International Journal of U- and e-Service, Science and Technology*, 8, 277–284. <http://dx.doi.org/10.14257/ijunesst.2015.8.7.28>. (2015)
- [112] Wu, Y., Wu, W., Xing, C., Zhou, M., & Li, Z. Sequential matching network: A new architecture for multi-turn response selection in retrieval-based chatbots. *ArXiv:1612.01627 [Cs]*. <http://arxiv.org/abs/1612.01627>. (2016)
- [113] Zumstein, D., & Hundertmark, S. Chatbots – an interactive technology for personalized communication. *Transactions and Services. IADIS International Journal on WWW/Internet*, 15, 96–109. (2017)
- [114] Zhou, H., Huang, M., Zhang, T., Zhu, X., & Liu, B. Emotional chatting machine: emotional conversation generation with internal and external memory. (2017)
- [115] Hahm, Y., Kim, J., An, S., Lee, M., & Choi, K.-S. Chatbot who wants to learn the knowledge: kb-agent. *Semdeep/NLIWoD@ISWC*, 4. (2018)

- [116] S., R., & Balakrishnan, K. Empowering chatbots with business intelligence by big data integration. *International Journal of Advanced Research in Computer Science*, 9(1), 627–631. <http://dx.doi.org/10.26483/ijarcs.v9i1.5398>. (2018)
- [117] Mislevics, A., Grundspen, k, is, J., & Rollande, R. A systematic approach to implementing chatbots in organizations—RTU leo showcase. *BIR workshops*. (2018)
- [118] Zhou, L., Gao, J., Li, D., & Shum, H.-Y. The design and implementation of xiaoice, an empathetic social chatbot. *ArXiv:1812.08989 [Cs]*. Retrieved from <http://arxiv.org/abs/1812.08989>. (2019)
- [119] Villegas, W., Arias-Navarrete, A., & Palacios, X. Proposal of an architecture for the integration of a chatbot with artificial intelligence in a smart campus for the improvement of learning. *Sustainability*, 12(1500), <http://dx.doi.org/10.3390/su12041500>. (2020)

11. INDEKS SLIKA

Slika 1. Dijagram sekvenci interakcije korisnika sa komponentama četbota	15
Slika 2. Domena primene četbota	17
Slika 3. Ciljevi četbota	19
Slika 4. Rezultati pretrage reči <i>chatbot</i> i <i>conversation agent</i> po godinama	27
Slika 5. Klasifikacija četbotova	30
Slika 6. Metode, tehnike i tehnologije razvoja četbota	35
Slika 7. Odnos AI, ML, DL i NLP	39
Slika 8. Komponente obrade prirodnog jezika (NLP)	40
Slika 9. Koraci u NLU planu izvršavanja	45
Slika 10. Fraza podeljena na manje delove teksta - tokene	46
Slika 11. Razlika razdvajanja karaktera u zavisnosti od teknizera	47
Slika 12. Dijagram koji prikazuje kodiranje rečenice	48
Slika 13. Korišćenje DIET klasifikatora	49
Slika 14. Ekstracija entiteta u NLU planu izvršavanja	49
Slika 15. Fajl „config.yml“ (redosled komponente u fajlu)	50
Slika 16. Dijagram obrade poruke	51
Slika 17. Dodavanje informacija u poruku	52
Slika 18. Opšti metodološki okvir	57
Slika 19. Opšta referentna arhitektura	58
Slika 20. Referentna arhitektura adaptivne platforme za razvoja četbota	60
Slika 21. Dinamički model sloja definicija	62
Slika 22. Dualni životni ciklus razvoja adaptivne platforme	64
Slika 23. Logička arhitektura	68
Slika 24. Dijagram komponenti <i>BOT engine-a</i>	71
Slika 25. Referentni kontekstni model – kontekstne kategorije	79
Slika 26. Veze između kontekstnih kategorija	79
Slika 27. Arhitektura modela	82
Slika 28. Opšti postupak za kreiranje adaptivnog četbota	85
Slika 29. DORIUS Metamodel	87
Slika 30. Dialogflow metamodel	90
Slika 31. Model preslikavanja Dialogflow2DORIUS	92

Slika 32. RASA metamodel	93
Slika 33. Model preslikavanja RASA2DORIUS	94
Slika 34. BPMN dijagram koreografije za proveru prosečne ocene studenta	96
Slika 35. BPMN dijagram kolaboracije za proveru stanja kredita	97
Slika 36. Generalizacija uloga korisnika u ADA četbotu	102
Slika 37. Dijagram slučajeva korišćenja za pružanje opštih informacija	103
Slika 38. Dijagram slučajeva korišćenja za zabavne scenarije	104
Slika 39. Dijagram slučajeva korišćenja za pružanje informacija o Akademiji	105
Slika 40. Dijagram slučajeva korišćenja za informacije o studentskom servisu	105
Slika 41. Dijagram slučajeva korišćenja za dodatne informacije o Akademiji	106
Slika 42. Dijagram slučajeva korišćenja za pripremnu nastavu i polaganje ispita	107
Slika 43. Dijagram slučajeva korišćenja za kontakte, programe, rangiranje i potvrde	108
Slika 44. Dijagram slučajeva korišćenja za prijavu ispita, konsultacije i rokove	109
Slika 45. Dijagram slučajeva korišćenja za stručnu praksu	110
Slika 46. Dijagram slučajeva korišćenja za overu semestra i dodatne informacije	111
Slika 47. Dijagram slučajeva korišćenja za odjavu ispita, stanje kredita i predavanja	112
Slika 48. Dijagram slučajeva korišćenja za školarinu i izmenu izbornog	113
Slika 49. Dijagram slučajeva korišćenja za kupovinu knjiga i pregled predmeta	114
Slika 50. Izbor NLP servisa i komunikacionih kanala	119
Slika 51. Dijagram koreografije za prijavu ispita u BOT dizajneru	120
Slika 52. Dijagram kolaboracije za prijava ispita u BOT dizajneru	121
Slika 53. Dijagram kolaboracije za verifikaciju korisnika u BOT dizajneru	122
Slika 54. Primer kontekstnog modela	124
Slika 55. Deo modela mapiranja između <i>Dialogflow</i> i DORIUS model	127
Slika 56. Panel Entiteti	128
Slika 57. Prikaz entiteta <i>IntentTeacherConsultation</i> koji je tipa <i>Intent</i>	129
Slika 58. Izmena odgovarajuće rečenice	130
Slika 59. Označavanje ključne reči na rečenici	130
Slika 60. Dodavanje označenog teksta kao entiteta	131
Slika 61. Čuvanje rečenice sa označenim entitetom	131
Slika 62. NLU validacija	134
Slika 63. Pregled validacije	134
Slika 64. Pregled Konverzacije	135
Slika 65. Brisanje NLP Validacije	135

12. INDEKS TABELA

Tabela 1. Osnove razlike između Dialogflow i RASA	53
Tabela 2. Detaljan scenario „Kako si?“	115
Tabela 3. Detaljan scenario „Prijava ispita“	115
Tabela 4. Detaljan scenario „Srećna nova godina“	116
Tabela 5. Detaljan scenario „Šta radiš?“	117
Tabela 6. Statistika pitanja i odgovora	139
Tabela 7. Broj verifikovanih korisnika i primljenih poruka	141
Tabela 8. Broj namera, fraza, entiteta, vrednosti entiteta i sinonima	141
Tabela 9. Broj konverzacija po kanalima u različitim periodima	142
Tabela 10. Prvih 10 scenarija i broja konverzacija (bez kratkih razgovora) u periodu od septembra do decembra 2020. Godine	144
Tabela 11. TOP 10 scenarija i broja konverzacija (bez kratkih razgovora) u periodima: Septembar 2020 - Decembar 2020 / Januar 2021 - Decembar 2021 / Januar 2022	145
Tabela 12. Broj korisnika i broj prepoznatih namera u odnosu na NLU servis	147
Tabela 13. Komparacija relevantnih radova	153