



UNIVERZITET U BANJOJ LUCI
ELEKTROTEHNIČKI FAKULTET



**AUTOMATIZOVANO PROJEKTOVANJE BAZE
PODATAKA NA OSNOVU GOVORA**

Master rad

Mentor:
Prof. dr Dražen Brđanin

Kandidat:
Dejan Keserović

Banja Luka, 2025.



UNIVERSITY OF BANJA LUKA
FACULTY OF ELECTRICAL
ENGINEERING



AUTOMATED SPEECH-BASED DATABASE DESIGN

Master thesis

Mentor:
Assoc. Prof. Dražen Brđanin, PhD

Candidate:
Dejan Keserović

Banja Luka, 2025.

Tema: AUTOMATIZOVANO PROJEKTOVANJE
BAZE PODATAKA NA OSNOVU GOVORA

Mentor: Prof. dr Dražen Brđanin, vanredni profesor
Elektrotehnički fakultet, Univerzitet u Banjoj Luci

Naučna oblast: Inženjerstvo i tehnologija

Naučno polje: Elektrotehnika, elektronika i informaciono
inženjerstvo

Ključne riječi: *Speed, VOSK, baza podataka, dijagram klasa, direktni
inženjering, konceptualni model, relacioni model, UML*

**Klasifikaciona oznaka
(CERIF):** T120

**Tip licence
Kreativne zajednice:** CC BY-SA

Komisija: Prof. dr Zoran Đurić, predsjednik
Prof. dr Dražen Brđanin, mentor
Prof. dr Jovan Galić, član

Sažetak: Tema ovog rada je automatizovano projektovanje baze podataka na osnovu govora. Predložen je pristup koji omogućava automatsko prepoznavanje govora i generisanje teksta, nakon čega se na osnovu analize tog teksta generiše konceptualni model baze podataka. Konceptualni model se može konvertovati u relacioni model, nakon čega se može generisati odgovarajuća DDL skripta. Nakon generisanja DDL skripte, alat omogućava kreiranje fizičke baze podataka za ciljni DBMS. Predloženi pristup je implementiran kao proširenje postojećeg Speed alata, koji predstavlja prvi alat za generisanje konceptualnog modela na osnovu govora.

Topic: AUTOMATED SPEECH-BASED
DATABASE DESIGN

Mentor: Dražen Brđanin, Associate Professor
Faculty of Electrical Engineering,
University of Banja Luka

Scientific area: Engineering and technology

Scientific field: Electrical engineering, electronics and information
engineering

Keywords: *Speed, VOSK, database, class diagram, forward engineering,
conceptual model, relational model, UML*

**Classification label
(CERIF):** T120

**Creative Commons
license:** CC BY-SA

Evaluation Committee: Prof. Zoran Đurić, Chairperson
Assoc. Prof. Dražen Brđanin, Mentor
Assoc. Prof. Jovan Galić, Member

Abstract: The topic of this master thesis is the automated speech-based database design. A proposed approach enables automatic speech recognition and text generation, after which a conceptual database model is generated based on the analysis of that text. The conceptual model can be converted into a relational model, after which a corresponding DDL script can be generated. After generating the DDL script, the tool allows for the creation of a physical database for the target DBMS. The proposed approach has been implemented as an extension of the existing Speed tool, which represents the first tool for generating a conceptual model based on speech.

Spisak korištenih skraćenica

Skraćenica	Puni naziv
ANN	Artificial Neural Network
API	Application Programming Interface
ASR	Automatic Speech Recognition
AWS	Amazon Web Services
BPM	Business Process Modeling
BPMN	Business Process Modeling Notation
CI/CD	Continuous Integration And Continuous Delivery
CNN	Convolutional Neural Network
DBMS	Database Management System
DDL	Data Definition Language
DNN	Deep Neural Network
DTW	Dynamic Time Warping
E-R	Entity-Relationship
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HTML	HyperText Markup Language
IDEF1X	Integration DEFinition For Information Modeling
IE	Information Engineering
JSON	JavaScript Object Notation
LPC	Linear Predictive Coding
LSTM	Long Short-term Memory
MFCC	Mel-frequency Cepstral Coefficients
NLP	Natural Language Processing
ORM	Object-Role Modeling
POS	Part Of Speech
RNN	Recurrent Neural Network
SGMM	Subspace Gaussian Mixture Model
TDNN	Time Delay Neural Network
TDNN-F	Factorized Time Delay Neural Network
UML	Unified Modeling Language
XMI	XML Metadata Interchange
XML	Extensible Markup Language

Sadržaj

1. Uvod	1
1.1. Predmet istraživanja	1
1.2. Cilj istraživanja i ostvareni rezultati	4
1.3. Metodologija istraživanja	4
1.4. Struktura rada	5
1.5. Objavljeni rezultati istraživanja	5
2. Automatizovano projektovanje baze podataka	6
2.1. Faze projektovanja baze podataka	6
2.1.1. Konceptualni dizajn	8
2.1.2. Logički dizajn	8
2.1.3. Fizički dizajn	9
2.2. pristupi i metode za automatizovano projektovanje baze podataka	9
2.2.1. Tekstualna specifikacija	10
2.2.2. Model	14
2.2.3. Forma	17
2.2.4. Govor	17
2.3. Unapređenje razvoja baze podataka kroz automatizaciju	20
3. Prepoznavanje govora	22
3.1. Istorijski razvoj	22
3.2. Proces prepoznavanja govora	23
3.3. Primjena alata za prepoznavanje govora	24
3.4. Alati i algoritmi za prepoznavanje govora	25
3.5. Vosk	27
3.6. Kaldi	27
3.7. Izazovi alata za prepoznavanje govora	28
4. Konceptualno modelovanje na osnovu govora	30
4.1. Razvoj konceptualnog modela baze podataka na osnovu govora	30
4.2. Obrada audio podataka	33
4.3. Provjera gramatike	33

4.3.1. Proces provjere gramatike	34
4.3.2. Alati za provjeru gramatike i pravopisa	35
4.4. Analiza teksta upotrebom NLP tehnika	36
4.5. Generisanje konceptualnog modela na osnovu teksta	36
5. Implementacija pristupa	37
5.1. Prepoznavanje govora	38
5.1.1. Prilaganje audio datoteke	41
5.1.2. Registrovanje govora u realnom vremenu	41
5.2. Provjera gramatike	42
5.3. Integracija sa TexToData i AMADEOS alatima	47
5.4. Klijentska aplikacija	49
5.5. Ilustrativni primjer projektovanja baze podataka na osnovu govora	51
6. Eksperimentalni rezultati	56
6.1. Očekivani rezultati konceptualnog modelovanja na osnovu govora	57
6.1.1. Očekivani rezultati konceptualnog modelovanja na primjeru organizacione strukture zasnovanog na govoru na engleskom jeziku	57
6.1.2. Očekivani rezultati konceptualnog modelovanja na primjeru organizacione strukture zasnovanog na govoru na njemačkom jeziku	58
6.1.3. Očekivani rezultati konceptualnog modelovanja na primjeru mrežne infrastrukture zasnovanog na govoru na engleskom jeziku	59
6.1.4. Očekivani rezultati konceptualnog modelovanja na primjeru mrežne infrastrukture zasnovanog na govoru na njemačkom jeziku	60
6.2. Eksperimentalni rezultati konceptualnog modelovanja na osnovu govora	61
6.2.1. Eksperimentalni rezultati konceptualnog modelovanja zasnovanog na govoru na engleskom jeziku na primjeru organizacione strukture	65
6.2.2. Eksperimentalni rezultati konceptualnog modelovanja zasnovanog na govoru na njemačkom jeziku na primjeru organizacione strukture	67
6.2.3. Eksperimentalni rezultati konceptualnog modelovanja zasnovanog na govoru na engleskom jeziku na primjeru mrežne infrastrukture	68
6.2.4. Eksperimentalni rezultati konceptualnog modelovanja zasnovanog na govoru na njemačkom jeziku na primjeru mrežne infrastrukture	70
6.3. Sistematizacija rezultata	71
7. Zaključak	74
Literatura	75

Glava 1

Uvod

1.1. Predmet istraživanja

Automatizacija projektovanja baza podataka predmet je istraživanja dugi niz godina. Ova oblast istraživanja nastala je sa ciljem smanjenja potrebnog vremena i resursa za projektovanje baza podataka.

Proces projektovanja baze podataka obično započinje projektovanjem konceptualnog modela, koji predstavlja osnovu za dalji razvoj. Na osnovu konceptualnog modela, mogu se automatski generisati odgovarajući relacioni model i fizička šema ciljne baze podataka. Kreiranje konceptualnog modela je često složen proces koji zahtijeva pažljivo razmišljanje i iterativni pristup. Zbog toga je automatizacija konceptualnog modelovanja, odnosno automatsko generisanje konceptualnog modela, od izuzetnog značaja.

Još u 1980-im godinama javljaju se prve ideje o automatskom generisanju konceptualnog modela baze podataka [1]. Najčešći pristupi automatskom generisanju konceptualnog modela jesu na osnovu tekstualne specifikacije [2] ili na osnovu modela [3], kao ulaznih artefakata.

Pristupi zasnovani na tekstualnoj specifikaciji su najkorišćeniji, a neki od najznačajnijih alata u ovoj oblasti su: ER-Converter [4], CM-Builder [5], APSARA [6], CABSYDD [7], OMDDE [8] i HBT [9]. Iako ideja automatske sinteze konceptualnog modela baze podataka na osnovu teksta datira s početka 1980-ih godina, još uvijek nema alata koji omogućava automatsko generisanje kompletnog konceptualnog modela, a alati tipično podržavaju samo jedan prirodni jezik. Jedini alat koji podržava više jezika jeste TexToData [10].

U poređenju sa pristupima koji kao polazni osnov uzimaju tekstualnu specifikaciju, pristupi zasnovani na modelima kao polazni osnov imaju vizuelne reprezentacije (dijagrame) različitih aspekata sistema od interesa. Prema [3], kao polazni osnov za automatizovanu sintezu konceptualnog modela koristi se dvadesetak različitih grafičkih notacija, koje se mogu klasifikovati kao notacije orijentisane ka: (1) procesima, (2) funkcijama, (3) komunikaciji i (4) ciljevima. Najbrojniji su pristupi koji kao polazni osnov imaju procesne modele, pri čemu se za njihovu reprezentaciju najviše koriste UML (eng. *Unified Modeling Language*) dijagram aktivnosti i BPMN (eng. *Business Process Modeling Notation*). Iako ideja modelom vođene automatske sinteze konceptualnog

modela baze podataka datira s početka 1980-ih godina, još uvijek nema alata koji omogućava automatsko generisanje kompletnog konceptualnog modela. Među postojećim alatima izdvaja se AMADEOS [11], koji podržava različite notacije i omogućava automatizovano generisanje konceptualnog modela na osnovu kolekcije modela poslovnih procesa. Pored toga, AMADEOS omogućava automatizaciju i ostalih koraka u direktnom inženjeringu relacione baze podataka [12].

Rjeđe korišćeni pristupi automatskom generisanju konceptualnog modela imaju kao polazni osnov forme i govor. Iako je govor prirodan i čovjeku svojstven (tekst i modeli su vještački), ipak još uvijek ne postoji alat koji omogućava automatsko projektovanje baze podataka na osnovu govora.

Automatizacija projektovanja baze podataka na osnovu govora mora uključivati prepoznavanje govora. Prepoznavanje govora je proces u kojem računarski sistemi konvertuju govor u tekst. Još od 1950-ih godina naučnici su pokušavali napraviti alat koji omogućava prepoznavanje govora, što je rezultovalo prvim alatom zvanim Audrey, nastalim 1952. godine [13].

Značajan napredak postignut je 1980-ih godina, uvođenjem skrivenog Markovljevog modela (eng. *Hidden Markov Model* – HMM). U narednim decenijama uvedeni su novi i inovativniji pristupi za prepoznavanje govora, kao što su korišćenje n-gram modela, vještačke neuronske mreže (eng. *Artificial Neural Network* – ANN) pa sve do današnjih LSTM (eng. *Long-short Term Memory*) metoda [14]. Danas se sve više koriste metode vještačke inteligencije za prepoznavanje govora, među kojima su najznačajnije bazirane na RNN (eng. *Recurrent Neural Network*), CNN (eng. *Convolutional Neural Network*) i DNN (eng. *Deep Neural Network*).

Alati za automatsko prepoznavanje govora (eng. *Automatic Speech Recognition* – ASR) nalaze široku primjenu u različitim oblastima, uključujući kreiranje virtuelnih asistenata, edukaciju, autentifikaciju, transkripciju itd. Istraživanje [15] pokazuje da se ASR alati mogu koristiti i u procesu projektovanja baza podataka – iz snimljenog govora može da se ekstrahuje tekst, na osnovu kojeg se dalje može generisati konceptualni model baze podataka.

Proces prepoznavanja govora započinje prikupljanjem audio podataka koji se obrađuju kroz nekoliko faza. Prva faza obuhvata poboljšanje određenih aspekata audio podataka, uz filtriranje najvažnijih podataka potrebnih za dalju obradu. U procesu ekstrakcije obilježja izdvajaju se ključne karakteristike snimljenog govora, koristeći metode kao što su LPC (eng. *Linear Predictive Coding* [16]), MFCC (eng. *Mel-frequency Cepstral Coefficients* [17]) i DTW (eng. *Dynamic Time Warping* [18]) [19]. Nakon toga, podaci se prosljeđuju akustičkom modelu koji prepoznaje zvukove na osnovu zvukova koje je naučio iz trening podataka. Akustički model procjenjuje vjerovatnoću različitih jezičkih elemenata, generišući vjerovatnoće za svaki lingvistički entitet. U ovoj fazi, mogu se koristiti različite metode kao što su HMM ili tehnike dubokog učenja. Dobijene vjerovatnoće jezički model koristi za izračunavanje najvjerovatnijeg niza riječi, koristeći statističke metode kao što su n-gram ili RNN. Krajnji rezultat ovog procesa je generisani tekst od strane računarskog sistema.

Neki od najznačajnijih alata za prepoznavanje govora su Kaldi [20], DeepSpeech [21], LinTo [22] i CMU Sphinx [23]. U poređenju sa ostalim alatima, kao što su CMU Sphinx i DeepSpeech, Vosk [24] se izdvojio kao alat sa najboljim performansama i mogućnostima

[25]. Vosk je jedan od najpoznatijih *open-source* alata za prepoznavanje govora, a njegova efikasnost omogućava implementaciju funkcionalnosti prepoznavanja govora na visokom nivou.

Prepoznavanje govora podrazumijeva korišćenje istreniranih modela za svaki podržani jezik. Modeli se treniraju na velikoj količini podataka, najčešće koristeći čiste audio podatke. Vosk omogućava prepoznavanje govora za više od 20 jezika, među kojima su engleski, njemački, ruski, italijanski itd. Vosk podržava velike i male modele, koji se razlikuju u sposobnosti prepoznavanja govora i zauzimanju prostora na disku. Veliki modeli pružaju veću tačnost, koja najviše zavisi od količine i kvaliteta podataka koji su upotrijebljeni za treniranje modela. Vosk koristi Kaldi model koji je kompajliran iz tri izvora (rječnik, akustički i jezički model). Alat omogućava kreiranje sopstvenog modela i jednostavnu integraciju sa različitim programskim jezicima kao što su Java, C i Python.

Postoje brojni izazovi prepoznavanja govora, kao što su reverberacija, šum, prepoznavanje govora različitih govornika i različite brzine govora. Takođe, govor različitih akcenata i dijalekata je izazovan za prepoznavanje. Zbog toga je važno pažljivo odabrati podatke za treniranje modela, kako bi ASR alat generisao odgovore koji su u skladu sa audio podacima. ASR alati najčešće ne prepoznaju znakove interpunkcije, što umanjuje preciznost generisanja teksta. Nakon dobijanja teksta, važno je utvrditi njegovu tačnost i gramatičku ispravnost. Za analizu gramatike dobijenog teksta mogu da se koriste alati kao što su ProWritingAid [26], Jazzy [27], LanguageTool [28] i dr.

Predmet istraživanja u ovom radu upravo je analiza mogućnosti upotrebe postojećih ASR alata u konceptualnom projektovanju baza podataka, kroz unapređenje postojećeg Speed alata [29] koji predstavlja prvi alat koji omogućava generisanje konceptualnog modela baze podataka na osnovu snimljenog govora [15]. Speed prvo vrši konverziju govora u tekst, nakon čega se tekst analizira upotrebom TexToData [10] alata koji generiše konceptualni model reprezentovan UML dijagramom klasa. Prethodna verzija Speed alata je imala podršku samo za engleski jezik, što je omogućilo prostor za unapređenje alata dodavanjem podrške za dodatne jezike. Dodatno, prethodni Speed je omogućavao samo prepoznavanje govora u *offline* režimu, tj. ekstrakciju teksta iz ranije snimljenog govora, što predstavlja dodatni prostor za unapređenje alata omogućavanjem ekstrakcije teksta i u *online* režimu, tj. ekstrakcije teksta iz govora koji se registruje mikrofonom u realnom vremenu. Raniji Speed nije vršio provjeru gramatičke ispravnosti ekstrahovanog teksta, a preliminarni rezultati pokazuju da ekstrahovani tekst sadrži gramatičke greške koje smanjuju tačnost automatskog generisanja konceptualnog modela baze podataka, što predstavlja dodatni prostor za unapređenje alata dodavanjem podrške za automatizovanu provjeru gramatičke ispravnosti generisanog teksta i automatizovane korekcije gramatičkih grešaka. Konačno, raniji Speed je omogućavao samo automatizovano generisanje inicijalnog konceptualnog modela, dok naredne faze u direktnom inženjeringu relacione baze podataka nisu bile podržane.

1.2. Cilj istraživanja i ostvareni rezultati

Cilj istraživanja je implementacija alata za automatizovano projektovanje baze podataka na osnovu govora. Alat je implementiran kroz unapređenje postojećeg Speed alata.

Prvi dio rada opisuje različite metode automatizovanog projektovanja baze podataka, istorijski razvoj, tehnike i alate za prepoznavanje govora, provjeru gramatike i pravopisa. U okviru praktičnog dijela rada je implementirana nova verzija Speed alata, koja omogućava konverziju audio podataka u tekst u *online* i *offline* režimima za različite jezike, uz dodatne mogućnosti provjere i korekcije gramatike ekstrahovanog teksta. Pored toga, postojećem Speed alatu dodata je podrška za direktni inženjering relacione baze podataka, koja već postoji u drugim alatima za automatizovano projektovanje relacionih baza podataka, koji su razvijeni u okviru istraživačke laboratorije M-lab.

Nakon završene implementacije, analizom je utvrđena tačnost u prepoznavanju u različitim jezicima, kao i prednost korišćenja alata za provjeru gramatike i pravopisa.

1.3. Metodologija istraživanja

Tokom istraživanja, proučeni su: (1) pristupi i alati za automatizovano projektovanje baza podataka; (2) pristupi i alati za prepoznavanje govora; (3) uticaj veličine jezičkih modela na kvalitet generisanih modela; (4) uticaj korekcije gramatičkih grešaka u ekstrahovanom tekstu na kvalitet generisanih modela; (5) mogućnosti Vosk biblioteke u pogledu podrške za višejezičnost te prepoznavanje govora u *offline* i *online* režimima.

Teorijski dio je poslužio kao osnova za implementaciju alata za automatizovano projektovanje baze podataka na osnovu govora, koji je implementiran u novoj verziji Speed alata. Ranija verzija Speed alata je proširena dodatnim komponentama za prepoznavanje govora i provjeru gramatike. Komponenta za prepoznavanje govora je realizovana upotrebom *web socket*-a, kako bi se ostvario efikasan način prepoznavanja u realnom vremenu. Komponenta za provjeru gramatike je realizovana kao *web* servis (nakon slanja zahtjeva od strane klijenta za provjerom gramatike za određeni tekst, klijent treba da dobije odgovor od strane servisa, prikaže ga u *web* čitaču i pruži korisniku mogućnost ispravke greške). Pored ovih komponenata, dodate su komponente za direktni inženjering relacione baze podataka, odnosno komponente za generisanje relacionog modela, DDL skripte i fizičke baze podataka. Kako bi se implementirale navedene komponente, izvršen je reinženjering klijentske i serverske strane postojećeg Speed alata. Implementacijom je kompletiran čitav proces projektovanja relacione baze podataka, od generisanja konceptualnog modela na osnovu govora, do generisanja ciljne šeme za konkretan sistem za upravljanje bazama podataka.

Nakon završene implementacije sprovedena je evaluacija unapređenog Speed sistema vrednovanjem automatski generisanih modela na osnovu govora.

1.4. Struktura rada

Master rad se sastoji iz sedam glava.

Prva glava predstavlja uvod, gdje se detaljno opisuje predmet, cilj i metodologija istraživanja.

Druga glava opisuje različite pristupe i metode automatizovanog projektovanja konceptualnog modela, kao i potencijalna unapređenja i efikasnost automatizacije za razliku od ručnog projektovanja.

Treća glava prikazuje različite tehnike prepoznavanja govora, istorijski razvoj alata za prepoznavanje govora, proces i algoritme za prepoznavanje. U ovoj glavi je objašnjeno na koji način ovi alati rade, posebno Vosk koji je u fokusu istraživanja.

Četvrta glava opisuje proces automatizovanog generisanja konceptualnog modela baze podataka na osnovu govora, počevši od prikupljanja audio podataka.

U narednoj glavi su opisane komponente novog Speed-a i implementacija.

Šesta glava prikazuje eksperimentalne rezultate generisanja entiteta, atributa, asocijacija i generalizacija u konceptualnom modelu. Izračunati su preciznost, odziv i efektivnost za svaki element.

Zaključak opisuje automatizovano projektovanje baze podataka na osnovu govora, kao i proširenje Speed alata kroz razvoj novih komponenata i integraciju sa TexToData i AMADEOS alatima.

1.5. Objavljeni rezultati istraživanja

Dijelovi istraživanja, koji su sprovedeni u okviru izrade master rada, objavljeni su u radu:

D. Brdjanin, G. Banjac, D. Keserovic, N. Babic, N. Golubovic, "Combining speech processing and text processing in conceptual database design," Telfor Journal, vol. 16, no. 1, pp. 8–13, 2024.

Glava 2

Automatizovano projektovanje baze podataka

Automatizovano projektovanje baze podataka uključuje korišćenje različitih softverskih alata za automatizaciju jednog dijela ili kompletnog procesa projektovanja baze podataka. Ovaj proces obuhvata generisanje konceptualnog modela, relacionog modela, DDL skripte i fizičke šeme baze podataka. U konceptualnom modelu definišu se entiteti, atributi i povezanosti između entiteta, koji predstavljaju osnov za dalje generisanje konkretne šeme baze podataka. Automatizacijom se postiže efikasnije projektovanje baze podataka, uz smanjenje mogućnosti nastanka grešaka u poređenju sa ručnim projektovanjem baze podataka. Kako projektovanje baze podataka podrazumijeva prolazak kroz više faza, prilikom automatizacije ovog procesa potrebno je obratiti posebnu pažnju na svaku od faza, jer se greška u jednoj fazi propagira u ostalim.

2.1. Faze projektovanja baze podataka

Projektovanje baze podataka je složen proces, koji uključuje nekoliko koraka potrebnih kako bi se implementirala baza podataka koja odgovara potrebama korisnika. Svaka od faza ima važnu ulogu u projektovanju i omogućava implementaciju baze podataka koja odgovara zahtjevima korisnika, pridržavajući se standarda i najboljih praksi u oblasti baza podataka. Cilj ovih faza je da se obezbijedi efikasnost, skalabilnost i integritet podataka.

Prvi korak u projektovanju baze podataka je planiranje i analiza zahtjeva. Planiranje baze podataka uključuje korake kao što su identifikacija ciljeva, korisnika i potrebnih resursa. U ovoj fazi se određuju planovi za budući razvoj baze podataka, kao i procjena rizika kako bi se osigurala stabilnost baze podataka na promjene u organizaciji. Cilj ove faze je da se prikupe i analiziraju heterogeni opisi zahtjeva i transformišu u homogene opise aplikacije [30]. Analiza zahtjeva počinje prikupljanjem i dokumentovanjem korisničkih zahtjeva. Prikupljanjem podataka se dolazi do zahtjeva, kao što su način čuvanja podataka, količina podataka i ograničenja. U ovoj fazi razvojni tim analizira zahtjeve u cilju identifikacije entiteta, atributa, ograničenja i povezanosti između entiteta. Ova faza je ključna za uspješno projektovanje baze podataka, jer se nastale

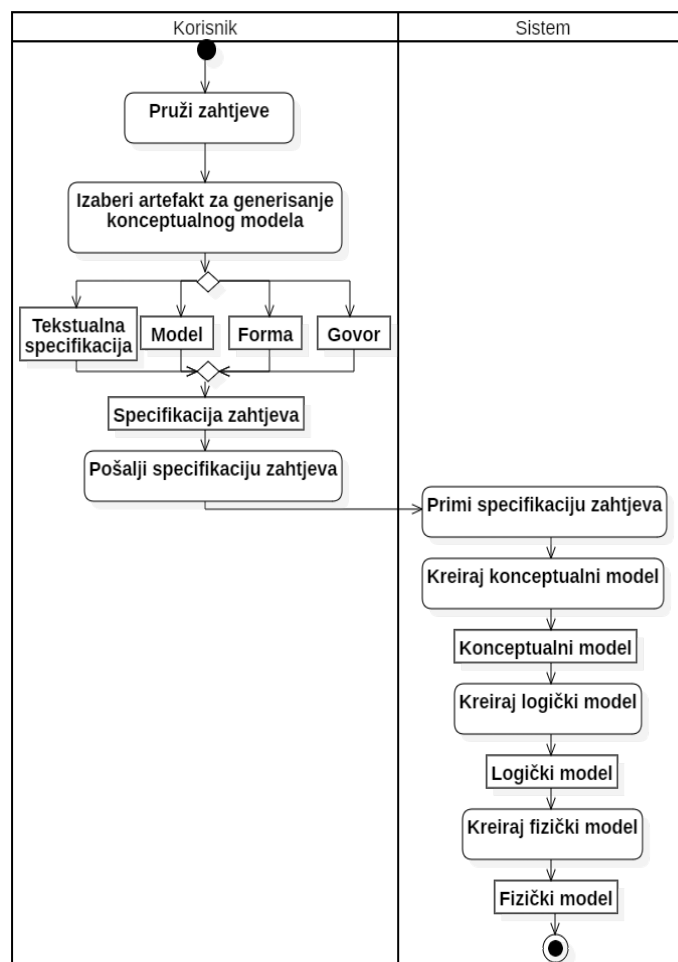
greške u ovoj fazi propagiraju u ostalim fazama, što dovodi do pogrešno generisane šeme baze podataka.

Analizom se utvrđuju elementi baze podataka kao što su entiteti i njihove veze, koji se prikazuju kroz upotrebu različitih vrsta dokumenata. Ispravnim i preciznim dokumentovanjem se postiže bolje razumijevanje korisničkih zahtjeva, što pruža jasnu sliku o cilju koji će se postići implementacijom baze podataka. Proces analize zahtjeva se može automatizovati tako što projektni tim dokumentuje zahtjeve u formatu koji može biti korišćen u alatu za analizu zahtjeva. Najčešće se ova faza realizuje kroz upotrebu teksta kao ulaznog artefakta ovih alata, što znači da projektni tim treba biti oprezan prilikom definisanja teksta.

Projektovanje baze podataka se može podijeliti u tri faze:

- konceptualni dizajn;
- logički dizajn;
- fizički dizajn.

Automatizacijom ovih faza se postiže automatizacija čitavog procesa projektovanja baze podataka. Faze projektovanja baze podataka prikazane su na sl. 2.1.



Slika 2.1: Faze automatizovanog projektovanja baze podataka

2.1.1. Konceptualni dizajn

Nakon dokumentovanja zahtjeva, analizom se utvrđuju elementi baze podataka kao što su entiteti i njihove veze, koji se najčešće vizuelno predstavljaju E-R (*eng. Entity-Relationship*) dijagramom. Konceptualnim modelom predstavljaju se potrebni podaci, nezavisno od aplikacije, softverskih i hardverskih sistema [30].

Vizuelnom reprezentacijom se postiže bolja preglednost osnovnih elemenata i omogućava jasnije razumijevanje strukture baze podataka. E-R dijagrami prikazuju entitetske tipove i veze između njih. Za svaki entitetski tip prikazuju se atributi, uključujući primarni ključ, kao i informacije o odnosu sa drugim entitetima i druge karakteristike. Postoje različite grafičke notacije za reprezentaciju konceptualnog modela, a neke od njih su:

- UML dijagram klasa;
- IDEF1X (*eng. Integration DEFINition for information modeling*) notacija;
- IE (*eng. Information Engineering*) notacija;
- Chen notacija;
- ptičje stopalo (*eng. Crow's Foot*) notacija.

Konceptualni model baze podataka predstavlja apstraktni pogled na strukturu podataka i njihove veze. Proces konceptualnog dizajna se može automatizovati korišćenjem alata za prepoznavanje entitetskih i veznih tipova. Prepoznati elementi se prikazuju odgovarajućim grafičkim konceptima u E-R dijagramu.

2.1.2. Logički dizajn

Logički dizajn pretvara model kreiran u konceptualnom dizajnu u strukturu baze podataka na logičkom nivou. Kao dominantan model za reprezentaciju logičkog dizajna koristi se relacioni model. Prema [31], postoji jasno definisan skup pravila za transformaciju konceptualnog modela u odgovarajući relacioni model.

U ovom koraku se vrše dodatne analize konceptualnog modela i unapređuje njegova struktura upotrebom normalizacije. Kroz proces normalizacije se smanjuje redundancija i osigurava ispravna povezanost između podataka. Ovaj proces se izvršava dekompozicijom koristeći normalne forme:

- prva normalna forma (1NF);
- druga normalna forma (2NF);
- treća normalna forma (3NF);
- Boyce-Codd normalna forma (BCNF);
- četvrta normalna forma (4NF).

Normalizacija omogućava efikasno manipulisanje podacima, tako što sprečava čuvanje istih podataka u više entiteta, te na ovaj način održava integritet podataka. Koristeći ove tehnike, relacije se dijele na više manjih kako bi se poboljšala organizacija podataka i osigurao kvalitet.

U ovoj fazi se vrši identifikacija tabela i definisanje primarnih i stranih ključeva, kao i ostalih ograničenja. Ova faza obezbjeđuje pravilno definisanje relacionog modela uz

optimizacione tehnike i definisanje pogleda sa ciljem generisanja efikasnog relacionog modela.

2.1.3. Fizički dizajn

Proces se završava fazom implementacije, koja uključuje korišćenje sistema za upravljanje bazom podataka (eng. *Database Management System* – DBMS) kako bi se implementirali identifikovani entiteti, ograničenja i veze između njih. Fizički dizajn podrazumijeva generisanje fizičke šeme baze podataka na osnovu logičke šeme koja je definisana u prethodnom koraku. U ovom koraku je potrebno odabrati ciljni DBMS kao što je MySQL, SQL Server, Oracle, PostgreSQL itd. U zavisnosti od potreba i dostupnih resursa potrebno je odabrati DBMS koji najbolje odgovara potrebama.

Postoje razni alati, kao što su Oracle SQL Developer Data Modeler i MySQL Workbench, koji omogućavaju konverziju relacionog modela u fizičku šemu baze podataka i obrnuto za različite DBMS. Fizička šema baze podataka se generiše izvršavanjem DDL (eng. *Data Definition Language*) skripte u konkretnom DBMS-u.

Nakon izvršavanja skripte u odgovarajućem DBMS-u, generiše se fizička šema baze podataka. U ovom koraku se mogu definisati prava pristupa bazi podataka za različite vrste korisnika. Takođe, mogu se koristiti tehnike za zaštitu u komunikaciji i čuvanju podataka kroz upotrebu enkripcije.

2.2. Pristupi i metode za automatizovano projektovanje baze podataka

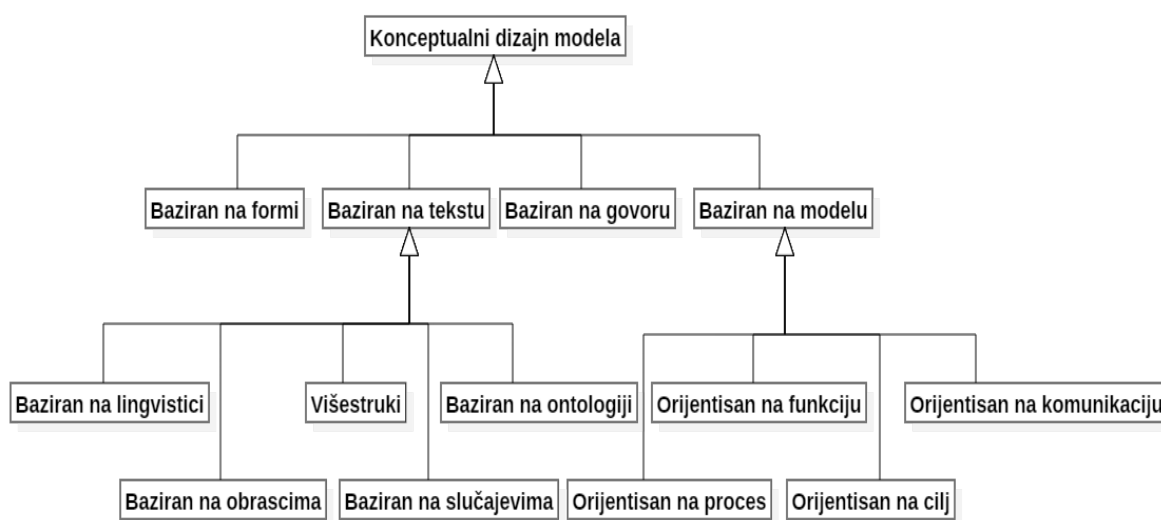
Tradicionalno, kreiranje baze podataka podrazumijeva složen proces ručnog generisanja šeme baze podataka, koji zahtijeva opsežno znanje i stručnost kako bi se stvorila šema koja najbolje odgovara specifikaciji zahtjeva. Ručno kreiranje šeme baze podataka podrazumijeva primjenu najboljih praksi, kao što su proces normalizacije, očuvanje integriteta i optimizacione tehnike. Razumijevanje ovih praksi ključno je za pravilno projektovanje koje ispunjava zahtjeve i osigurava integritet i sigurnost podataka.

S obzirom na konstantno napredovanje tehnologije, postoji sve veća potreba za razvojem kompleksnijih baza podataka. Ručno generisanje baze podataka postaje sve složenije, što dovodi do češće pojave grešaka. Međutim, ove poteškoće se mogu izbjeći korišćenjem alata za automatizovano projektovanje baza podataka. Automatizacijom ovog procesa, za kratko vrijeme se omogućava generisanje šeme baze podataka uz pridržavanje osnovnih principa projektovanja baze podataka, uključujući normalizaciju i optimizaciju.

Prvi korak u automatizovanom projektovanju baze podataka je dizajn konceptualnog modela kroz upotrebu E-R ili UML alata za vizuelno predstavljanje. Nakon ovog koraka, logički model se mapira u odgovarajući fizički model, odnosno bazu podataka u odabranom DBMS-u.

Na sl. 2.2 prikazana je taksonomija pristupa za automatizovani dizajn konceptualnog modela. Prema [15], pristupi za automatsko generisanje konceptualnog modela zasnivaju se na:

1. tekstualnoj specifikaciji;
2. modelu;
3. formi;
4. govoru.



Slika 2.2: Taksonomija pristupa za automatizovani dizajn konceptualnog modela [15]

Metode koje se oslanjaju na tekstualnu specifikaciju i model kao ulazne artefakte češće se primjenjuju u odnosu na metode koje se oslanjaju na formu i govor. Cilj ovog istraživanja je kreiranje alata koji koristi govor kao ulazni artefakt za automatsko generisanje konceptualnog modela, odnosno konkretne šeme baze podataka.

Različite tehnike se mogu koristiti u ovim pristupima, kao što su tehnike mašinskog učenja i metode vještačke inteligencije. U budućnosti se može očekivati poboljšanje ovih procesa kako se ove tehnologije budu razvijale.

2.2.1. Tekstualna specifikacija

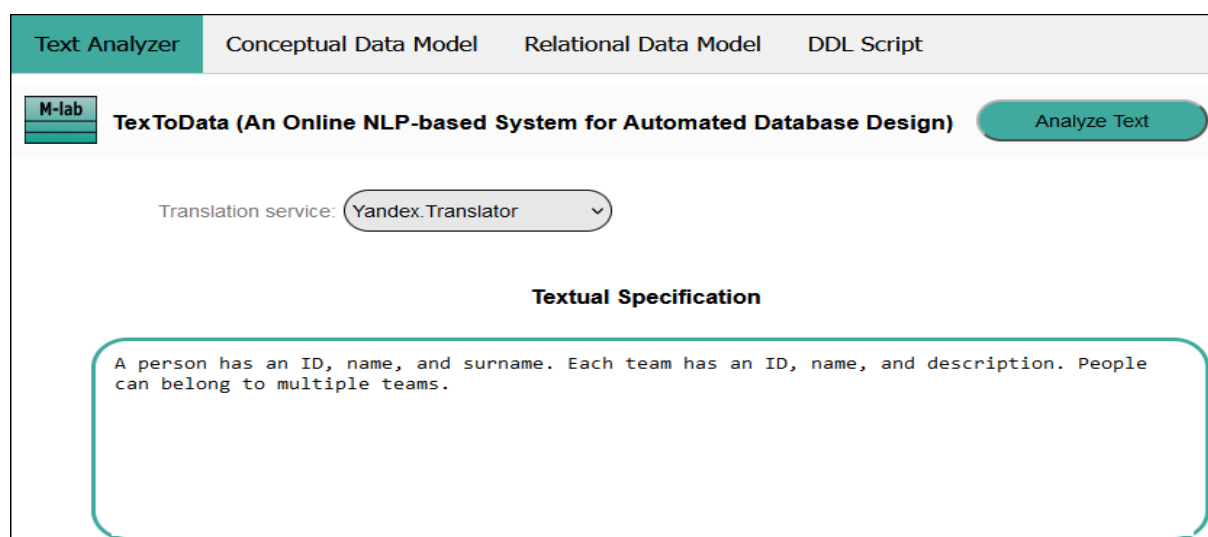
Korišćenje tekstualne specifikacije predstavlja najpopularniji način za automatsko generisanje konceptualnog modela. Najznačajniji predstavnici pristupa zasnovanog na tekstualnoj specifikaciji kao ulaznom artefaktu su TexToData, DBomnia [32], ER-Converter [4] i CM-Builder [5]. DBomnia omogućava generisanje konceptualnog modela koristeći različite ulazne artefakte. Ona omogućava kombinovanje tekstualnih specifikacija i procesnih modela kao osnovu za automatsko generisanje konceptualnog modela. Automatsko generisanje konceptualnog modela na osnovu tekstualne specifikacije je kompleksan proces koji podrazumijeva procesiranje i analizu teksta na prirodnom jeziku, kako bi se ustanovili entiteti, atributi, ograničenja i veze između entiteta. Procesiranje teksta podrazumijeva tokenizaciju, odnosno izdvajanje

pojedinačnih riječi i njihovo dodjeljivanje u gramatičke kategorije kao što su imenice, pridjevi i glagoli. Imenice u tekstu mogu označavati entitete u konceptualnom modelu. Identifikacija veza između entiteta i njihovih atributa je otežana jer je potrebno dublje razumijevanje konteksta kako bi se definisale relacije. Nakon identifikacije, vizuelno se predstavljaju svi elementi konceptualnog modela. Na primjer, identifikacija imenica iz tekstualne specifikacije može predstavljati entitete, dok je analizom glagola moguće izvući informacije o vezama između entiteta.

Potencijalni izazovi koji mogu nastati prilikom generisanja konceptualnog modela na osnovu tekstualne specifikacije su [2]:

1. znanje – Nedostatak znanja i iskustva mogu uticati na generisani konceptualni model, posebno kod dizajnera početnika.
2. kompleksnost veza – Veze između entiteta nekada znaju biti kompleksne tako da ih je teško identifikovati na osnovu teksta.
3. različito rješenje istog problema – Projektanti baza podataka na različite načine mogu riješiti isti problem, te generisani konceptualni model predstavlja gledište projektanata na zahtjeve krajnjih korisnika. Zbog ovog razloga je teško na osnovu teksta odabrati ispravan način, jer različita rješenja mogu biti podjednako ispravna.

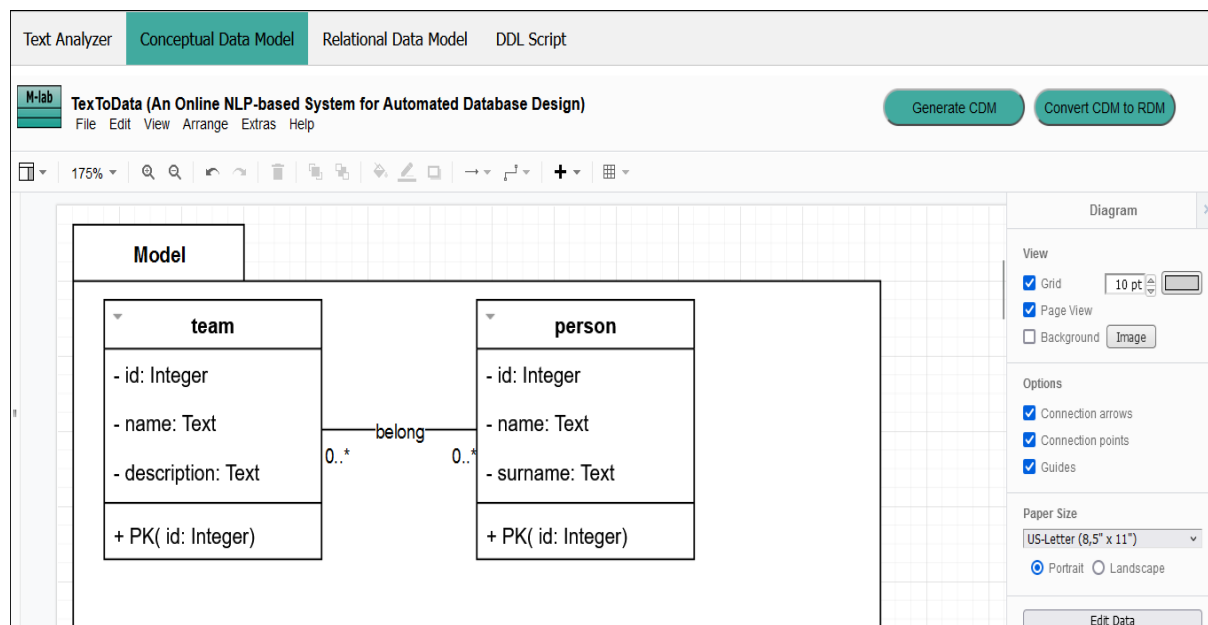
Potrebno je adresirati navedene izazove prilikom automatizovanog generisanja konceptualnog modela na osnovu tekstualne specifikacije kako bi se dobio odgovarajući model. Nedostaci ovog pristupa su što oni najčešće podržavaju samo jedan jezik, najčešće engleski [33]. *TexToData* je prvi višezjezički *online web* bazirani alat za automatsko izvođenje konceptualnog modela baze podataka na osnovu prirodnog jezika [33]. Na sl. 2.3 prikazana je početna stranica *TexToData* alata i uneseni tekst koji će biti analiziran u narednim koracima.



Slika 2.3: Početna stranica *TexToData* alata

Na sl. 2.4 prikazan je konceptualni model koji se dobija na osnovu ulaznog teksta nakon odabira opcije *Analyze Text*. Alat je uspješno identifikovao entitete, attribute i vezu između

entiteta. Alat omogućava mnogobrojne opcije za manipulaciju kreiranim konceptualnim modelom, snimanje, štampanje, eksportovanje i druge opcije.



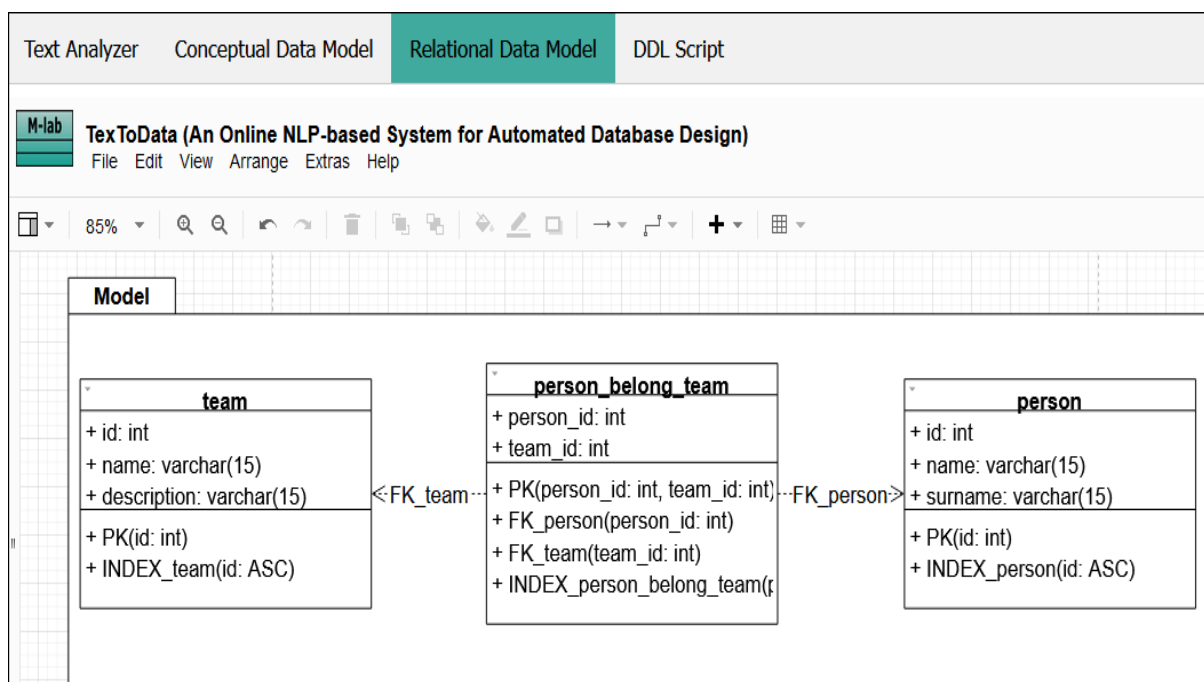
Slika 2.4: Konceptualni model kreiran TexToData alatom

Naredni korak je kreiranje relacionog modela na osnovu konceptualnog modela. Odabirom opcije *Convert CDM to RDM* alat prikazuje prozor sa sl. 2.5 u kojem je potrebno odabrati vrstu DBMS-a i načine mapiranja tipova atributa. Alat podržava pet DBMS-ova, a to su MySQL, Microsoft SQL, Oracle, PostgreSQL i IBM DB2.

Slika 2.5: Parametri za kreiranje relacionog modela u TexToData alatu

Odabirom opcije *Convert*, na osnovu odabranih parametara kreiraće se odgovarajući relacioni model kao što je prikazano na sl. 2.6. Alat omogućava manipulaciju relacionim

modelom kroz izmjene, snimanje, štampanje, i druge opcije. U navedenom primjeru alat je uspješno konvertovao konceptualni u relacioni model i izvršio mapiranje veznog tipa (asocijacije) u dodatnu klasu koja predstavlja veznu klasu, zajedno sa odgovarajućim stranim ključevima koji definišu veze između klasa koje reprezentuju relacije (tabele).



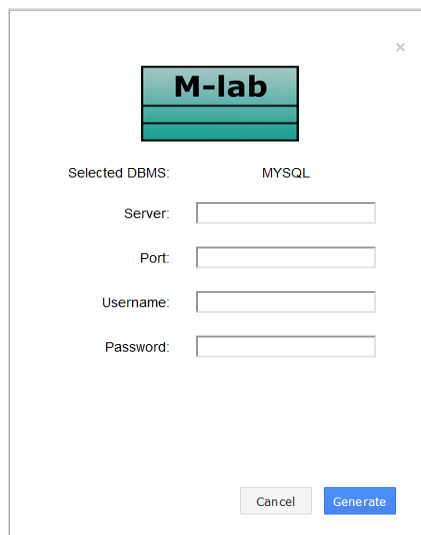
Slika 2.6: Relacioni model u TexToData alatu

Odabirom opcije *Generate DDL* kreira se DDL skripta na osnovu relacionog modela, što je prikazano na sl. 2.7. Alat omogućava izmjenu DDL skripte i generisanje fizičke baze podataka.

Slika 2.7: DDL skripta u TexToData alatu

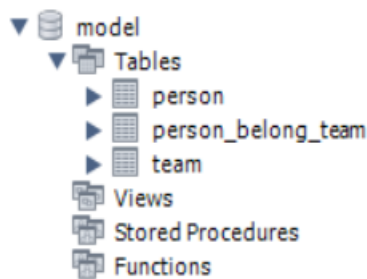
Odabirom opcije *Generate physical database* otvara se prozor sa sl. 2.8 u kojem je potrebno odabrati server, port, korisničko ime i lozinku DBMS-a. U navedenom primjeru

je odabrana MySQL baza podataka. Odabirom opcije *Generate* automatski će se kreirati odgovarajuća baza podataka.



Slika 2.8: Parametri za generisanje fizičke šeme baze podataka

Izvršavanjem skripte, generiše se odgovarajuća fizička baza podataka, što je prikazano na sl. 2.9.



Slika 2.9: Fizička šema baze podataka

Ovim korakom se završava proces generisanja konkretne šeme baze podataka na osnovu tekstualne specifikacije upotrebom TexToData alata.

2.2.2. Model

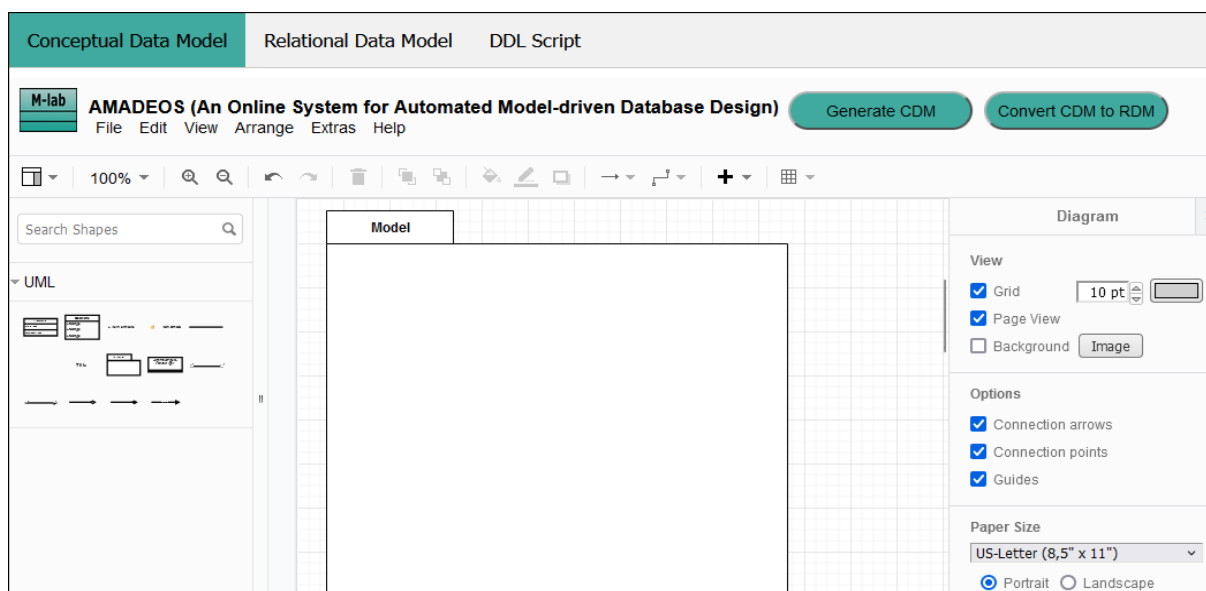
Automatsko generisanje konceptualnog modela na osnovu modela se odnosi na proces u kojem se koriste drugi prethodno definisani modeli, kao što su modeli poslovnih procesa, kako bi se automatski generisao konceptualni model baze podataka. Postoje mnoge grafičke notacije koje se mogu koristiti u ovom pristupu, dok se one po [3] dijele na one orijentisane ka: 1) procesima, 2) funkcijama, 3) komunikaciji i 4) ciljevima.

Pristup zasnovan na modelu koristi modele poslovnih procesa kao ulazne artefakte za generisanje konceptualnog modela. Način obrade modela zavisi od vrste modela i od toga kako predstavljaju poslovne procese. U slučaju dijagrama aktivnosti, različiti akteri i objekti koji se koriste u poslovnom procesu mogu predstavljati entitete, dok aktivnosti

predstavljaju osnov za generisanje veza između entiteta u konceptualnom modelu. Mapiranjem ovih elemenata dijagrama aktivnosti može se kreirati odgovarajući konceptualni model. Na primjer, ukoliko dijagram aktivnosti prikazuje proces izdavanja sertifikata polaznicima kursa, aktivnost "Pošalji sertifikat polazniku" može se mapirati u atribut "Datum slanja sertifikata", koji označava datum slanja sertifikata polazniku.

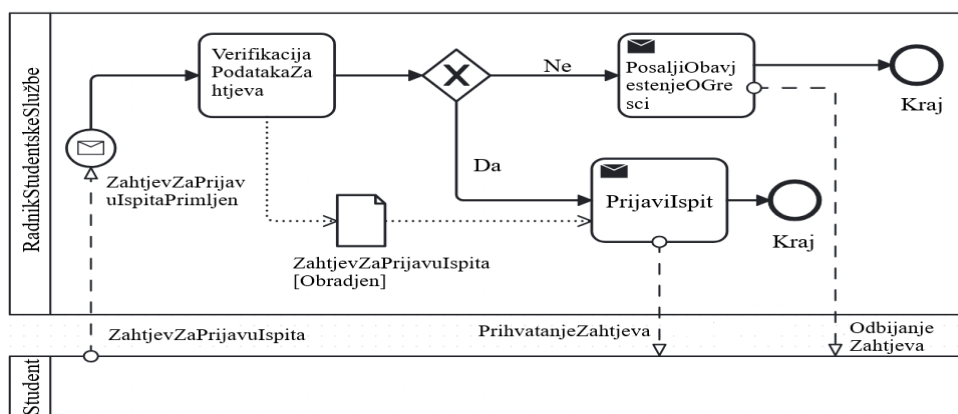
Ograničenja ovog pristupa se ogledaju u nedostatku semantičkog kapaciteta modela poslovnih procesa, jer na osnovu ovog modela nije moguće kreirati kompletnu strukturu modela sa svim vezama. Iz ovih razloga, potrebno je kombinovati ovaj pristup sa drugim izvorima podataka kako bi se dobio kompletan konceptualni model [33]. AMADEOS alat se posebno izdvaja kao alat koji uz korišćenje različitih notacija omogućava automatizovano generisanje konceptualnog modela na osnovu kolekcije modela poslovnih procesa.

Na sl. 2.10 prikazana je početna stranica AMADEOS alata.



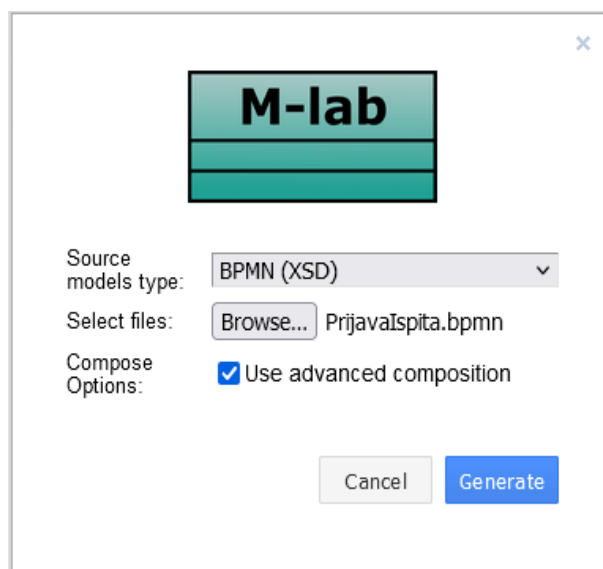
Slika 2.10: Početna stranica AMADEOS alata

Poslovni proces koji je korišćen za prikazivanje rada AMADEOS alata predstavljen je na sl. 2.11.



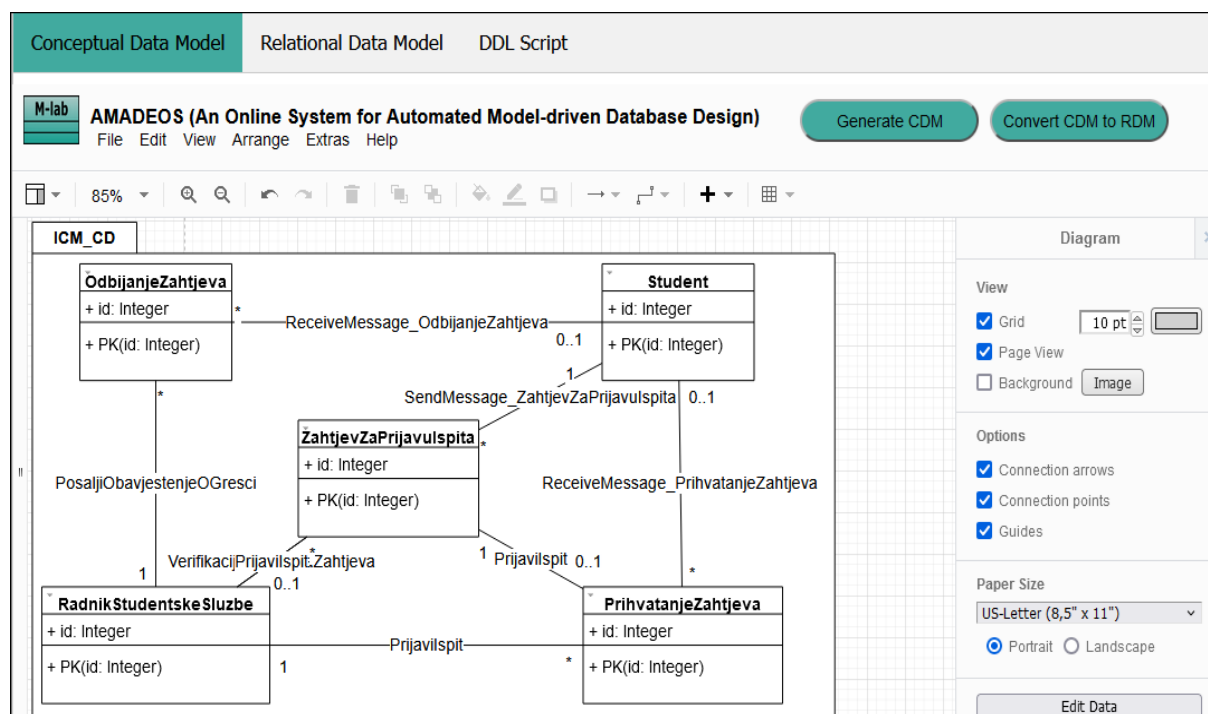
Slika 2.11: Model poslovnog procesa

Odabirom opcije *Generate CDM* otvara se modalni prozor sa sl. 2.12. Odabirom tipa modela i datoteke u kojoj je definisan model poslovnog procesa, moguće je generisati odgovarajući konceptualni model.



Slika 2.12: Modalni prozor za odabir parametara za generisanje konceptualnog modela

Generisani konceptualni model prikazan je na sl. 2.13.



Slika 2.13: Konceptualni model generisan AMADEOS alatom

2.2.3. Forma

Na osnovu formi strukturisanih na određen način moguće je automatizovati proces generisanja konceptualnog modela baze podataka. Entiteti, njihovi atributi i veze između entiteta se mogu definisati upotrebom dobro strukturisanih formi, tako što se analiziraju polja forme koji se mapiraju u entitete i attribute. Najznačajniji predstavnici ovog pristupa su EDDS [7] i IIS*Case [34].

2.2.4. Govor

Jedan od pristupa za generisanje konceptualnog modela je putem snimljene specifikacije zahtjeva koja se obrađuje posebnim alatima. Korišćenjem ranije snimljenog govora u audio datoteke ili bilježenjem audio podataka putem mikrofona u realnom vremenu, te prosljeđivanjem ovih podataka alatima za prepoznavanje govora, može se izvršiti ekstrakcija najbitnijih informacija.

Ove informacije se obrađuju sa ciljem generisanja konceptualnog modela koji odgovara audio podacima. Ovakav vid generisanja konceptualnog modela je najprirodniji, jer podrazumijeva govor kao ulazni artefakt. Proces se može implementirati korišćenjem alata za prepoznavanje govora, što podrazumijeva transkripciju koja generiše tekstualnu specifikaciju. Analiza tekstualne specifikacije bi prošla kroz iste faze kao u orijentaciji ka tekstualnoj specifikaciji. Jedini alat koji omogućava generisanje konceptualnog modela na osnovu govora je alat Speed.

Speed je prvi alat koji omogućava djelimičnu automatizaciju projektovanja baze podataka na osnovu govora. Alat omogućava automatizovano generisanje konceptualnog modela na osnovu govora.

Prvi korak projektovanja podrazumijeva unos audio podataka kao ulaznog artefakta kroz prilaganje audio datoteke. Audio podaci se analiziraju i kao rezultat se dobija tekst koji predstavlja tekstualnu specifikaciju.

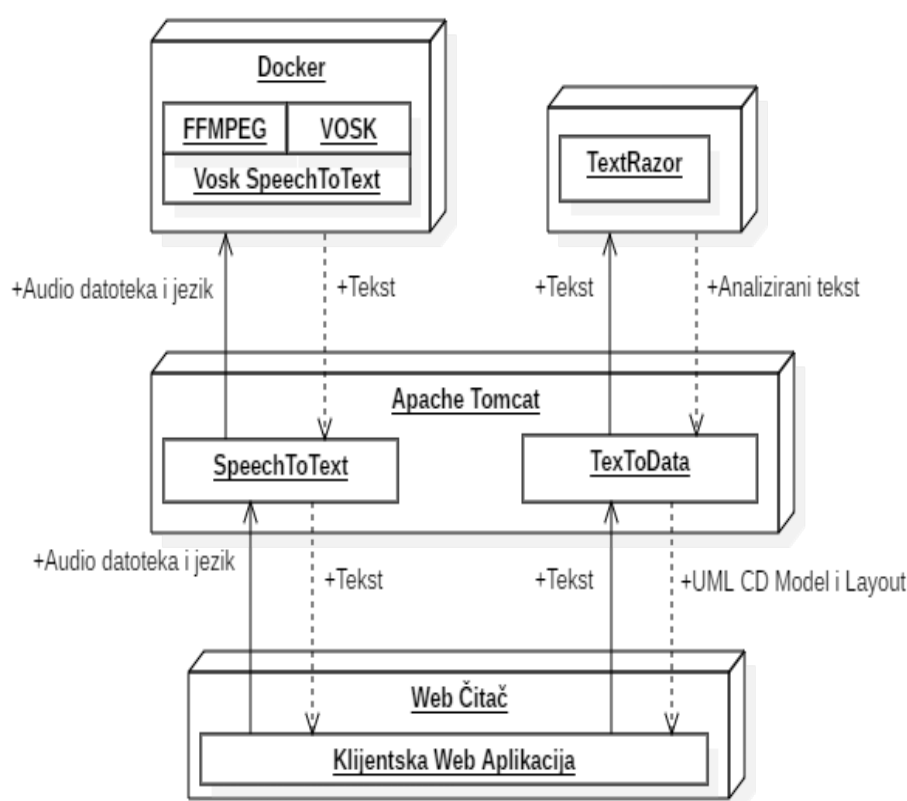
U narednim koracima Speed koristi logiku implementiranog *TexToData* alata, koji omogućava automatizovano generisanje konceptualnog modela na osnovu dobijene tekstualne specifikacije. Alat nudi mnoge mogućnosti kao što je dodavanje klasa, generalizacije, asocijacije, agregacije, kompozicije, kao i eksportovanje rezultata u XMI (eng. *XML Metadata Interchange*) formatu.

Alat omogućava navedenu funkcionalnost za veći broj jezika kao što su engleski, njemački i grčki, uz prethodno prilaganje audio datoteke. Nakon prilaganja audio datoteke, izvršava se procesiranje datoteke i generisanje odgovarajućeg teksta koji se prikazuje u tekstualni HTML (eng. *HyperText Markup Language*) element na klijentskoj strani. Odabirom opcije za generisanje dijagrama klasa, pokreće se proces generisanja modela, nakon čega se prikazuje odgovarajući dijagram klasa na osnovu teksta iz tekstualnog HTML elementa.

Proces generisanja konceptualnog modela je izvršen upotrebom *TexToData* alata koji omogućava automatizovano kreiranje konceptualnog modela. Nakon kreiranja konceptualnog modela, omogućeno je korišćenje velikog broja opcija za izmjenu vizualizacije modela, ažuriranje modela po potrebi kao i dodatne akcije kao što su čuvanje, preuzimanje i ubacivanje postojećih modela.

Na sl. 2.14 prikazana je arhitektura Speed-a. Klijentska strana aplikacije služi za slanje zahtjeva, dok serverska ima servisnu arhitekturu koja prima zahtjeve i šalje odgovore. Alat predstavlja kolekciju servisa, a proces sinteze konceptualnog modela je implementiran kao orkestracija. Orkestracija upravlja čitavim procesom od dobijanja teksta do vraćanja automatski generisanog konceptualnog modela.

Kako je na osnovu govora moguće generisati odgovarajuću tekstualnu specifikaciju, naredni koraci generisanja konceptualnog modela su isti kao i u pristupu zasnovanom na tekstualnoj specifikaciji, što predstavlja logiku implementacije Speed alata. Speed kroz integraciju alata za prepoznavanje govora i TextToData alata omogućava kompletan proces generisanja konceptualnog modela na osnovu govora.

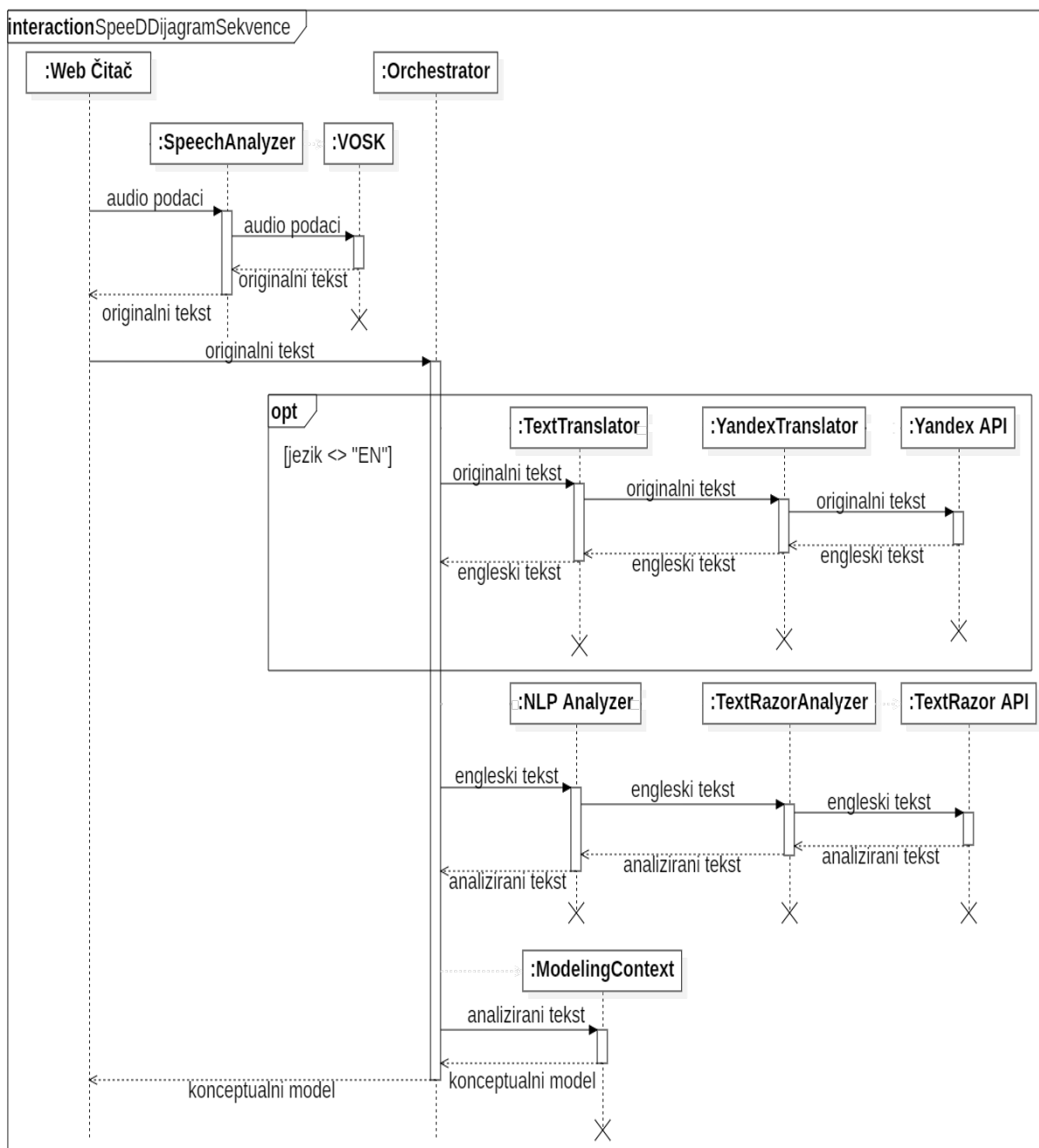


Slika 2.14: Arhitektura Speed-a [35]

Na sl. 2.15 prikazan je dijagram sekvence rada Speed alata. Na početku, *web* čitač šalje audio datoteku koja se proslijeđuje komponenti za analizu govora, koja uz korišćenje Vosk alata generiše odgovarajući tekst.

Tekst se dalje vraća klijentskoj strani i prikazuje u *web* čitač korisnika. Naredni korak je prevod dobijenog teksta na engleski jezik, jer alat za generisanje konceptualnog modela radi samo na engleskom jeziku. *Yandex* API (eng. *Application Programming Interface*) vraća odgovarajući tekst na engleskom jeziku, koji se analizira upotrebom *NLP Analyzer* komponente.

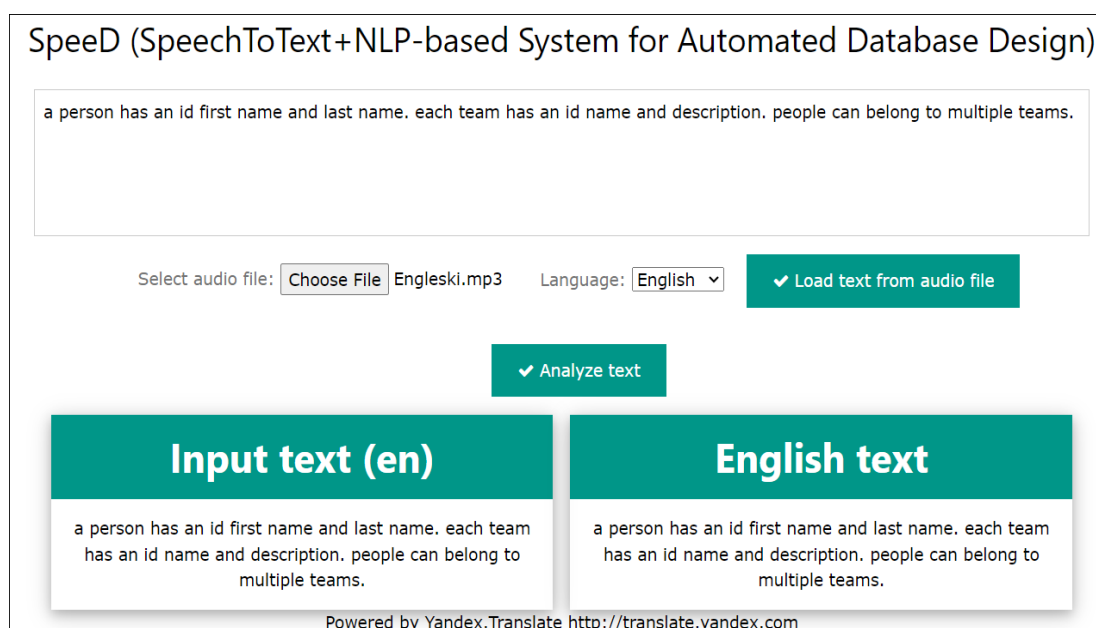
Komponenta koristi *TextRazorAnalyzer* i *TextRazorAPI* koji kao rezultat vraćaju analizirani tekst. Na kraju, *ModelingContext* komponenta kreira konceptualni model na osnovu analize iz prethodnog koraka.



Slika 2.15: Dijagram sekvence Speed alata [33]

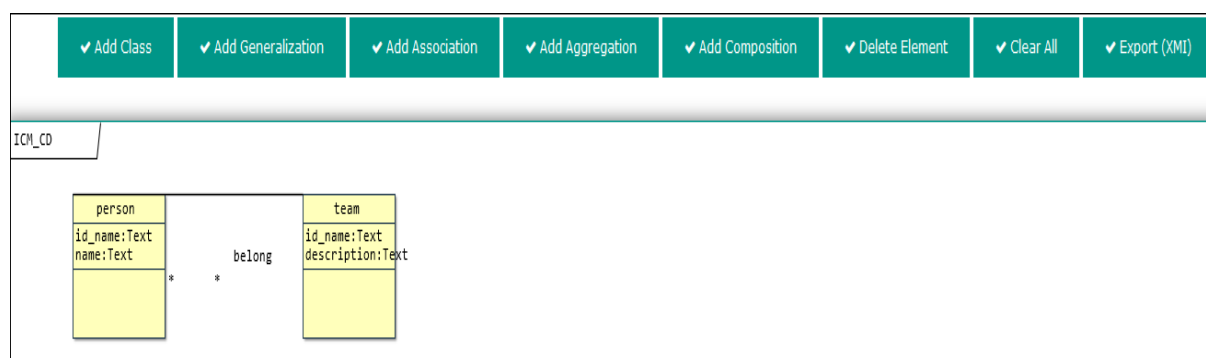
Postojeći alat se može unaprijediti dodatnim jezicima, omogućavanjem prepoznavanja govora u realnom vremenu, kao i uvođenjem alata za provjeru gramatike sa ciljem dobijanja tačnijeg konceptualnog modela. Sve ove izmjene su implementirane u novoj verziji Speed alata.

Na sl. 2.16 prikazan je dio početne stranice Speed alata nakon prepoznavanja govora na osnovu priložene audio datoteke u kojoj je izgovorena specifikacija zahtjeva na engleskom jeziku. Sistem za analizu teksta je prepoznao da je u pitanju engleski jezik, tako da prepoznati tekst odgovara prevedenom tekstu.



Slika 2.16: Početna stranica SpeedD alata

Na sl. 2.17 prikazan je drugi dio početne stranice u kojem se nalazi generisani konceptualni model i opcije za manipulaciju nad modelom.



Slika 2.17: Konceptualni model generisan u SpeedD alatu

2.3. Unapređenje razvoja baze podataka kroz automatizaciju

Tradicionalno, proces projektovanja baze podataka se izvršavao ručno. Razvojni timovi su se oslanjali na stručnost i iskustvo kako bi generisali šemu baze podataka. Razvojni timovi bi kroz različite alate imali mogućnost definisanja šeme baze podataka i dijagrama za modelovanje baze podataka. Pojava prvih alata je omogućila djelimičnu automatizaciju i mogućnost jednostavnijeg projektovanja baze podataka.

Različiti pristupi su korišćeni za implementaciju alata za automatizovani razvoj baze podataka. Današnji pristupi za projektovanje baza podataka podrazumijevaju korišćenje

inteligentnih alata zasnovanih na mašinskom učenju i vještačkoj inteligenciji, čime se postiže znatno veći uspjeh u automatizaciji projektovanja baze podataka. Neki od primjera ovakvih alata su Oracle Autonomous Database [36] i Microsoft Azure SQL Database [37]. Ovi alati mogu analizirati postojeće podatke i šablone za efikasno kreiranje šeme baze podataka.

Baze podataka su često povezane sa programskim jezicima, omogućavajući njihovu manipulaciju kroz programski kod. Iz ovog razloga pojavili su se razni alati koji omogućavaju generisanje šeme baze podataka na osnovu modela podataka u programskom kodu. Ovi alati često podržavaju generisanje modela baze podataka u okviru programskog koda, prethodnim definisanjem šeme baze podataka. Na ovaj način automatizacijom je smanjeno vrijeme potrebno za pisanje programskog koda, kao i smanjenje nastanka greške. Neki alati omogućavaju generisanje šeme baze podataka na osnovu specifikacija visokog nivoa. Ovi koraci predstavljaju početak procesa automatizacije projektovanja baze podataka.

Integracija sa DevOps alatima olakšava timovima da sarađuju prilikom promjena na bazi podataka. Obrazac kontinuirane integracije i implementacije (eng. *Continuous Integration And Continuous Delivery – CI/CD*) korišćen u DevOps okruženju, često koristi automatizaciju upravljanja bazom podataka. Na ovaj način omogućeno je rješavanje savremenih izazova koje donosi tehnologija. Razvojem tehnologije povećava se kompleksnost savremenih sistema, te je od ključne važnosti korišćenje savremenih alata koji kroz inovativne pristupe pružaju veliku tačnost, preciznost, efikasnost i sigurnost u projektovanju baza podataka. Praćenje trendova smanjuje vrijeme potrebno za projektovanje baza podataka i unapređuje konkurentnost kompanija.

Za razliku od manuelnog kreiranja baze podataka, automatizacija nudi brojne prednosti. Neke od osnovnih prednosti automatizovanog kreiranja baze podataka su:

1. skraćanje vremena – Automatizacija projektovanja baze podataka drastično smanjuje vrijeme potrebno za projektovanje. Manuelni rad podrazumijeva detaljnu analizu specifikacije, dok se automatizacijom može odmah doći do rezultata.
2. optimizacija – Automatizacijom postizemo optimizovani rad sa bazom podataka, smanjujući potrebne resurse za čuvanje podataka, što može biti od velikog značaja u velikim i kompleksnim bazama podataka.
3. smanjenje mogućnosti nastanka greške – Manuelno kreiranje baza podataka je podložno greškama, koje se mogu izbjeći korišćenjem alata za automatizaciju ovog procesa.
4. fleksibilnost – Kako su baze podataka podložne promjeni, automatizacijom se može na jednostavan način izvršiti adaptacija postojeće baze novim zahtjevima, bez potrebe za ručnim izmjenama.
5. skalabilnost – Automatizovani sistemi su veoma efikasni u prilagođavanju različitim zahtjevima te se mogu primijeniti na veliki broj baza podataka.
6. sigurnost – Moguće je obezbijediti sigurnost baza podataka, što smanjuje rizik od malicioznog djelovanja na bazu podataka.

Glava 3

Prepoznavanje govora

Prepoznavanje govora predstavlja proces generisanja odgovarajućeg teksta na osnovu audio podataka od strane softverskih alata. Ne postoji veliki broj softverskih alata koji omogućavaju prepoznavanje govora zbog svoje kompleksnosti, pogotovo kada je riječ o podršci za višejezičnost. Efikasnost i preciznost prepoznavanja govora su poboljšani korišćenjem savremenih pristupa, uključujući vještačku inteligenciju.

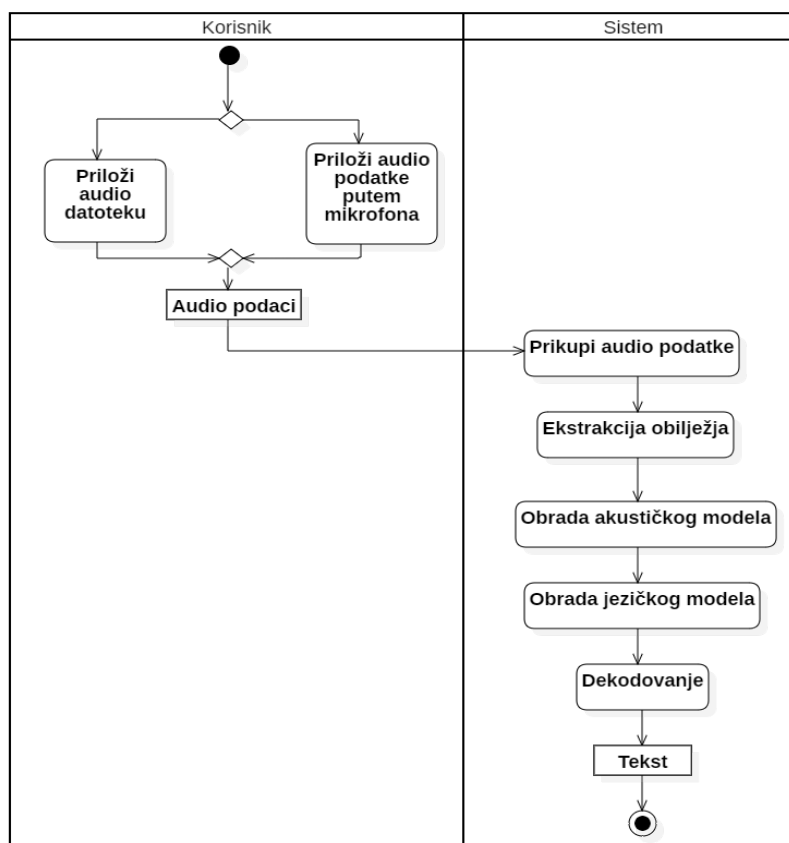
3.1. Istorijski razvoj

Još od 1950-ih godina, istraživači pokušavaju napraviti alat koji može razumjeti ljudski govor. Ova istraživanja su rezultovala nastankom prvog alata za prepoznavanje govora, zvanom Audrey, koji je razvijen u Belovim laboratorijama 1952. godine [13].

U početku, ovi alati su koristili jednostavnije metode za prepoznavanje, što uključuje korišćenje HMM, koji predstavlja dominantnu metodu za prepoznavanje govora koja se koristi i danas. Razvoj mašinskog učenja 1980-ih godina doveo je do pomaka u prepoznavanju govora. Mašinsko učenje je kroz modele neuronske mreže za prepoznavanje govora razvijalo različite alate za prepoznavanje govora, uključujući upotrebu RNN i metode dubokog učenja. Iako su se pojavili novi i bolji načini prepoznavanja govora, ovi alati su većinom imali mogućnost prepoznavanja govora jednog govornika. U 1990-im godinama, počele su se razvijati druge metode za prepoznavanje govora koje uključuju prepoznavanje govora različitih govornika, kroz korišćenje metoda kao što je model Gausovih mješavina (eng. *Gaussian Mixture Model* – GMM). Ovo je omogućilo da se podaci za treniranje modela ne fokusiraju na pojedinačnog govornika, nego da mogu obuhvatiti različite govornike. U 2000-im godina je nastala potpuno nova era prepoznavanja govora, koja uključuje metode dubokog učenja, najviše DNN i CNN. Ove tehnike uveliko poboljšavaju način prepoznavanja i treniranja modela, omogućavajući efikasno prepoznavanje uz visoke performanse, skalabilnost i preciznost [14].

3.2. Proces prepoznavanja govora

Prepoznavanje govora je kompleksan proces koji se sastoji od nekoliko ključnih faza. Kada se ove faze povežu, one formiraju cjelinu koja osigurava preciznost u prepoznavanju govora. U nastavku su prikazani osnovni koraci prepoznavanja govora, od prikupljanja audio podataka pa sve do generisanog teksta. Na sl. 3.1 prikazan je proces prepoznavanja govora.



Slika 3.1: Proces prepoznavanja govora

Faze prepoznavanja govora su:

1. prikupljanje audio podataka – Prvi korak u prepoznavanju govora je prikupljanje audio podataka. Ovo uključuje korišćenje postojećih audio datoteka ili prikupljanje audio podataka putem mikrofona u realnom vremenu. U ovoj fazi je važno da zvuk bude čist, odnosno da nema šuma i buke kako bi se što više smanjio štetan uticaj pozadinskog zvuka i obezbijedilo adekvatno prepoznavanje.
2. ekstrakcija obilježja – Iz prikupljenih audio podataka izdvajaju se bitni podaci za analizu u narednim koracima. Audio podaci se obično razdvajaju na manje cjeline kako bi se zasebno analizirali. Postoje razni načini za ekstrakciju obilježja, a neki od najznačajnijih su: LPC, MFCC i DTW. Dobijeni podaci se prosljeđuju akustičkom modelu.

3. akustičko modelovanje – Na osnovu podataka dobijenih iz ekstrakcije obilježja govora, akustički model koristi različite algoritme kako bi izračunao vjerovatnoće prepoznatih riječi. U ovoj fazi mogu se koristiti različite metode od HMM do metoda mašinskog učenja. Akustički model treba biti treniran na velikoj količini podataka kako bi mogao prepoznati lingvističke jedinice, odnosno foneme. Akustički model mapira akustička svojstva u foneme ili riječi. Krajnji rezultat akustičkog modela su vjerovatnoće lingvističkih jedinica s obzirom na izvedena obilježja u prethodnom koraku.
4. jezičko modelovanje – Jezički model koristi rezultat akustičkog modelovanja kako bi, upotrebom različitih statističkih metoda, prepoznao riječi. Jezički model pomaže predviđanju najvjerovatnije lingvističke jedinice na osnovu semantičkih informacija da bi se postigla veća tačnost. Neke od metoda koje se mogu koristiti u ovoj fazi su RNN, LSTM mreže i modeli transformatora.
5. dekodovanje – Izlaz jezičkog modela se analizira i odlučuje o krajnjem rezultatu procesa. Proces dekodovanja izdvaja riječi sa najvećom vjerovatnoćom. Na osnovu ovih vjerovatnoća se izdvajaju različite sekvence riječi i bira se ona sa najvećom vjerovatnoćom.
6. završetak procesa – Proces se završava generisanjem teksta koji predstavlja prepoznat govor na osnovu ulaznih audio podataka iz početne faze. U ovoj fazi se mogu koristiti različite tehnike za provjeru gramatike radi ispravljanja grešaka, poput grešaka u znakovima interpunkcije ili pogrešno napisane riječi.

3.3. Primjena alata za prepoznavanje govora

Alati za prepoznavanje govora, mogu biti korišćeni u raznim domenima, kao što su:

1. zdravstvo – Prepoznavanje govora se može koristiti u zdravstvu na različite načine, a jedan od njih je transkripcija medicinskih dokumenata. Zdravstveni radnici mogu koristiti posebne uređaje za diktiranje bilješki o pacijentima, uklanjajući potrebu za pisanjem. Ovo ubrzava proces dokumentovanja i povećava efikasnost zdravstvenog sektora.
2. finansije – Za mnoge korisnike bankarskih usluga korišćenje terminala, na kojima se mogu dobiti informacije o računu, kao i izvršavanje akcija na računu, može biti komplikovano. Uvođenjem alata za prepoznavanje govora, korisnici ovih servisa mogu putem govora davati komande, te se na ovaj način olakšava korišćenje bankarskih usluga.
3. automobilska industrija – Automobilska industrija sve češće koristi alate za prepoznavanje govora kako bi olakšala upotrebu automobila. Upotrebom govora za davanje komandi kao što su podešavanje temperature, navigacija, slanje poruka ili inicijacija poziva, vozači mogu ostati fokusirani na vožnju.
4. obrazovanje – Prepoznavanje govora u edukaciji je posebno izraženo prilikom učenja stranih jezika. Ovi alati generišu odgovarajući tekst i mogu koristiti dodatne alate za analizu rezultata te ih uporediti sa očekivanjem. Na ovaj način korisnici odmah mogu dobiti informacije o tačnosti izgovorenih riječi. Dodatno,

licima sa poteškoćama u učenju ovi sistemi mogu pomoći čuvanjem bilješki putem govora.

5. zabava – Često se u igrama koriste virtuelni asistenti za komunikaciju unutar igre. Ova integracija omogućava glasovnu kontrolu, pogotovo korišćenu u virtuelnoj realnosti.

Korišćenje ovih alata osnažuje korisnike da svoje svakodnevne aktivnosti obavljaju na prirodni način putem govora.

3.4. Alati i algoritmi za prepoznavanje govora

Postoje razni alati za prepoznavanje govora. Neki od njih uključuju *online* platforme za prepoznavanje, dok drugi predstavljaju biblioteke i alate prilagođene za korišćenje u programskim jezicima. Neki od najpoznatijih *online* platformi i alata za prepoznavanje govora su:

1. Google Speech-to-Text – Google-ov API za prepoznavanje govora. API pruža veliku preciznost i podršku za 125 jezika. Alat koristi najnovije tehnologije zasnovane na mašinskom učenju, kao što su CNN i RNN. Uz doplatu, moguće je integrisati ovaj sistem u različite aplikacije i servise. Za razliku od tradicionalnih tehnika za prepoznavanje govora koji koriste trenirane modele za specifičan jezik, Speech-to-Text koristi jedinstven model za sve jezike što pomaže boljem prepoznavanju govora. Speech-to-Text je treniran na milionima sati audio podataka i milijardama tekstualnih rečenica. Speech-to-Text omogućava odabir između različitih modela koji se koriste za različite namjene kao što glasovna kontrola, telefonski pozivi i video transkripcija [38].
2. Amazon Transcribe – Amazon-ov alat za automatsko prepoznavanje govora koji se jednostavno integriše u aplikacije za potrebe generisanja teksta na osnovu govora. Dio je AWS-a (eng. *Amazon Web Services*) koji omogućava prepoznavanje govora u realnom vremenu koristeći moderne tehnologije dubokog učenja. Alat omogućava podršku za veliki broj jezika i koristi model treniran na milionima sati audio podataka. Amazon Transcribe omogućava precizne transkripcije uzimajući u obzir bučna okruženja, različite naglaske i akustičke uslove [39].
3. Microsoft Azure AI Speech – Microsoft-ov *online* alat koji omogućava generisanje teksta na osnovu audio podataka i konverziju govornog jezika u drugi jezik u realnom vremenu. Alat koristi napredne AI tehnologije i pruža dodatne funkcionalnosti kao što su prevod govora, analiza govora, verifikovanje i prepoznavanje govornika [40].
4. Mozilla DeepSpeech – *Open-source* biblioteka koja za prepoznavanje govora koristi tehnike dubokog učenja. Biblioteka se može koristiti u raznim platformama, programskim jezicima, te omogućava jednostavnu integraciju sa drugim servisima. Biblioteka omogućava treniranje modela na sopstvenim podacima za domenski specifične namjene. DeepSpeech na zvaničnoj *web* stranici pruža uputstva za korišćenje i konfiguraciju projekata za programske jezike koje podržava.

5. CMU Sphinx – Predstavlja *open-source* Java biblioteku koja je razvijena na Carnegie Mellon Univerzitetu, koja koristi akustički i jezički model za prepoznavanje, te pruža podršku za *online* i *offline* prepoznavanje. Alat omogućava korisnicima kreiranje sopstvenih modela. CMU Sphinx koristi različite modele za prepoznavanje govora a to su: lista ključnih riječi, gramatika, statistički i fonetski jezički modeli. CMU Sphinx pruža API za konvertovanje govora u tekst, kao i prepoznavanje govornika. Konfiguracija projekta je pojednostavljena, te su na zvaničnoj *web* stranici prikazani primjeri korišćenja biblioteke [41].
6. Kaldi – Predstavlja *open-source* alat za prepoznavanje govora. Kaldi se koristi u raznim alatima kao što je Vosk. Alat koristi razne akustičke modele i omogućava prepoznavanje upotrebom GMM-a i tehnika dubokog učenja, kao što je DNN. Kaldi omogućava dekodovanje i treniranje podataka. Česta ažuriranja alata omogućavaju preciznije prepoznavanje uz upotrebu savremenijih tehnologija.

Neki od navedenih alata su *open-source*, što znači da ih je moguće prilagođavati sopstvenim potrebama. Alati se razlikuju po preciznosti, performansama, pristupačnosti u pogledu cijene koštanja, te je u zavisnosti od namjene potrebno izabrati alat koji najviše odgovara potrebama.

Kroz istoriju su se pojavljivali noviji i inovativniji načini prepoznavanja govora, kroz upotrebu savremenih algoritama. Počevši od HMM-a, pa sve do tehnika dubokog učenja, ovi alati poboljšavaju tačnost i performanse prepoznavanja govora.

Najznačajnije tehnike za prepoznavanje govora su:

1. HMM – Razvijen 1970-ih godina, koristi se za modelovanje akustičkih signala, te se od 1990. godine počeo koristiti za prepoznavanje riječi i prepoznavanje većeg broja govornika u komunikaciji. Koristi se i danas, a njegov nedostatak se ogleda u sofisticiranim i kompleksnim računanjima, kao i duže trajanje treniranja modela za razliku od drugih tehnika.
2. GMM – Predstavlja statistički model koji se obično koristi u kombinaciji sa HMM-om, koji reprezentuje vjerovatnoće u distribuciji podataka. GMM se često koristi kod ekstrakcije obilježja za modelovanje distribucije akustičkih karakteristika.
3. ANN – Način na koji funkcioniše ANN je sličan načinu funkcionisanja biološkog nervnog sistema, koristeći vezu čvorova, analogno neuronima. Svaku neuronsku mrežu karakterišu [42]:
 - karakter čvorova – određuje na koji način čvor obrađuje signale.
 - mrežna topologija – određuje na koji način su čvorovi povezani.
 - pravila učenja – određuje na koji način se težine određuju i prilagođavaju.
4. CNN – CNN je tip DNN-a koji je efikasan za tip podataka u obliku mreže, kao što su slike. Inicijalno su se koristili za obradu slika, dok su kasnije korišćeni i u svrhu prepoznavanja govora. CNN se koristi u fazi ekstrakcije obilježja upotrebom slojeva konvolucije. Na ovaj način uče kako da u budućnosti izdvoje bitne podatke.
5. RNN – predstavlja tip DNN-a koji čuva ranije dobijene podatke u memoriji kako bi prepoznao zavisnosti koje su ključne u prepoznavanju govora.

3.5. Vosk

Vosk je jedan od najpoznatijih alata za prepoznavanje govora. Prema istraživanju [24], za razliku od drugih poznatih alata za prepoznavanje govora kao što su DeepSpeech i Sphinx, Vosk se pokazao kao najbolji alat. Velika preciznost, mogućnost prepoznavanja govora velikog broja jezika i performanse su najveće prednosti ovog alata. Vosk predstavlja biblioteku za prepoznavanje koja se na jednostavan način može integrisati u okviru različitih programskih jezika kao što su Java, C i Python.

Treniranje govora je kompleksan proces, jer zahtijeva opsežne podatke kako bi se model trenirao. Vosk podržava male i velike modele. Veliki modeli zauzimaju više prostora na disku, ali su precizniji jer su trenirani na većim količinama podataka i koriste savremene načine treniranja kroz upotrebu mašinskog učenja i vještačke inteligencije. Mali modeli obično koriste tradicionalne metode, kao što je HMM. Iako su se tradicionalne metode pokazale kao dobre u prepoznavanju govora, savremene metode pružaju veću tačnost. Vosk podržava prepoznavanje govora za više od 20 jezika, a neki od njih su engleski, njemački, grčki, francuski, njemački itd. Vosk omogućava dodavanje novih jezika, uz minimalne izmjene programskog koda. Pružanje prepoznavanja govora za nove jezike podrazumijeva dodavanje postojećih ili kreiranje novih jezičkih modela. Zbog načina na koji Vosk funkcioniše, pored *offline* prepoznavanja, moguće je postići funkcionalnost prepoznavanja govora u realnom vremenu, tj. govora koji nije prethodno snimljen, nego se analizira neposredno nakon registrovanja mikrofonom. Jedan od zadataka ovog istraživanja je korišćenje Vosk alata u svrhu kreiranja alata koji može prepoznati govor u realnom vremenu i na osnovu prepoznatog govora generisati inicijalni konceptualni model baze podataka.

Osnova Vosk alata je Kaldi model koji je kompajliran iz tri modela, a to su rječnik, akustički i jezički model. Pored gotovih modela, Vosk omogućava kreiranje sopstvenih modela kako bi se povećala tačnost prepoznavanja ili kako bi se model prilagodio određenim domenima i akcentima. Vosk je skalabilna platforma, te se jednostavno može integrisati sa drugim platformama i različitim uređajima, od *desktop* računara do mobilnih uređaja. Veliki broj korisnika kroz forume, grupe za diskusije i GitHub repozitorijum, pruža veliku podršku zajednici. Vosk nudi dokumentaciju koja objašnjava korišćenje alata u raznim programskim jezicima, olakšavajući njegovu primjenu. Ova dokumentacija objašnjava načine za instalaciju, konfiguraciju, korišćenje i najbolje prakse za ovaj alat.

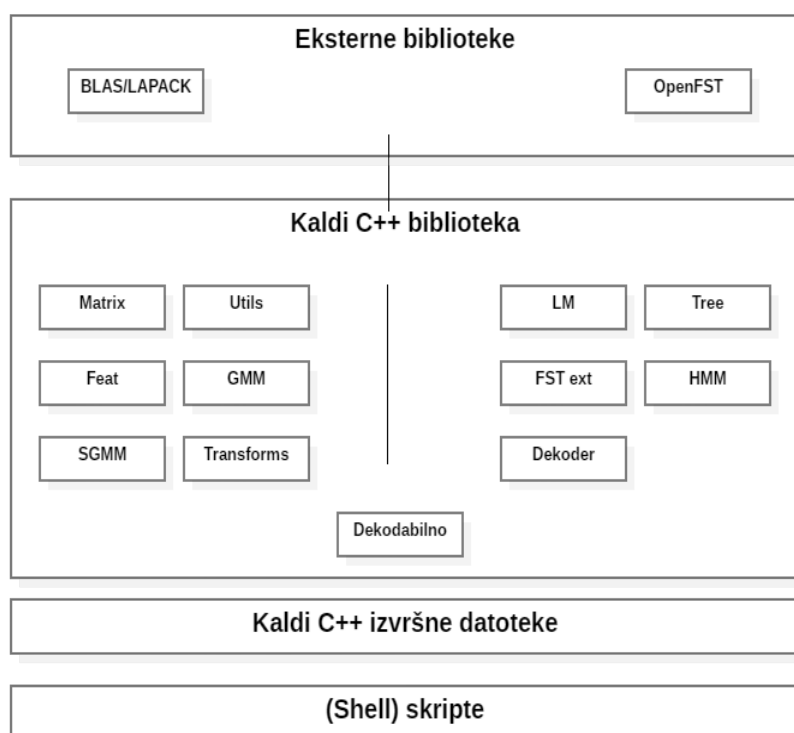
3.6. Kaldi

Kaldi je *open-source* skup alata za prepoznavanje govora napisan u C++ programskom jeziku, koji se može koristiti sa bibliotekama i alatima za postizanje funkcionalnosti prepoznavanja govora. Kaldi ima modularnu arhitekturu što olakšava istraživačima proširivanje postojećih funkcionalnosti. Kaldi pruža jednostavno razumljiv kod, koji uz jednostavnu konfiguraciju programerima olakšava izmjene i rad sa alatom. Kaldi pruža akustičko modelovanje, jezičko modelovanje, ekstrakciju obilježja i druge

funkcionalnosti za prepoznavanje govora. Kaldi u okviru akustičkog modelovanja podržava GMM i SGMM (eng. *Subspace Gaussian Mixture Model*) [20].

Veliki broj korisnika Kaldi-ja pruža podršku putem liste elektronske pošte. Kaldi karakteriše brzina i efikasnost u prepoznavanju govora. Na zvaničnoj *web* stranici alata postavljena je detaljna dokumentacija koja uključuje primjere i uputstva koja pomažu korisnicima da koriste alat.

Na sl. 3.2 prikazane su osnovne komponente Kaldi alata. Prikazani su algoritmi koji se koriste u raznim fazama prepoznavanja govora i korišćene eksterne biblioteke.



Slika 3.2: Komponente Kaldi alata [20]

Do sada je objavljeno šest verzija Kaldi alata, dok je trenutna verzija 5.5.

3.7. Izazovi alata za prepoznavanje govora

Alati za prepoznavanje govora su doživjeli značajne pomake, ali kako bi se povećala tačnost u prepoznavanju govora, neophodno je adresirati brojne izazove u prepoznavanju.

Osnovni izazovi alata za prepoznavanje govora su:

1. prepoznavanje različitih govornika – Ljudi imaju različite karakteristike glasa, kao što su brzina govora i intonacija, te je teško kreirati takav alat koji bi mogao poslužiti za prepoznavanje govora različitih govornika u situacijama kada više osoba istovremeno govori.

2. treniranje modela – Prepoznavanje govora podrazumijeva postojanje modela za svaki od jezika, što znači da je potrebno prikupiti mnogo audio podataka za treniranje modela za svaki od jezika koje bi alat podržavao.
3. varijacije u brzini, akcentima i dijalektima – Govor različitih brzina, akcenata i dijalekata je takođe problematičan za prepoznavanje. Različite brzine govora utiču na karakteristike zvuka, posebno u slučajevima kada se određene riječi izgovaraju brže ili sporije nego što bi trebalo. Modeli su obično trenirani na određenim akcentima i dijalektima, tako da je prepoznavanje otežano ukoliko je govor drugog akcenta ili dijalekta.
4. šum i pozadinska buka – Kako se modeli obično treniraju na čistim audio podacima, odnosno podacima bez šuma i pozadinske buke, prepoznavanje govora pod ovim uslovima je otežano i daje slabiju tačnost u odnosu na čist zvuk.
5. reverberacija – Reverberacija, odnosno zvuk odjeka, može izmijeniti karakteristike zvuka te otežati njegovo prepoznavanje. Postoje brojna istraživanja o negativnim uticajima reverberacije na prepoznavanje govora, a neki od njih su opisani u istraživanjima [43] i [44].
6. nedostatak konteksta – Nedostatak konteksta je problematičan za prepoznavanje, jer ovi alati najčešće izračunavaju vjerovatnoće prepoznatih riječi na osnovu prethodnih riječi, što može rezultovati pogrešnim prepoznavanjem.
7. kompleksna struktura – Kako su ovi sistemi često kompleksne strukture, kreiranje alata sa dobrim performansama je otežano. Takođe, prepoznavanje govora je iz istih razloga problematično u realnom vremenu gdje je brzina procesiranja od velikog značaja.

Glava 4

Konceptualno modelovanje na osnovu govora

Projektovanje baze podataka na osnovu govora podrazumijeva proces generisanja šeme baze podataka na osnovu audio podataka. Audio podaci se analiziraju različitim alatima, nakon čega se dolazi do osnovnih elemenata potrebnih za generisanje konceptualnog modela, odnosno odgovarajuće šeme baze podataka.

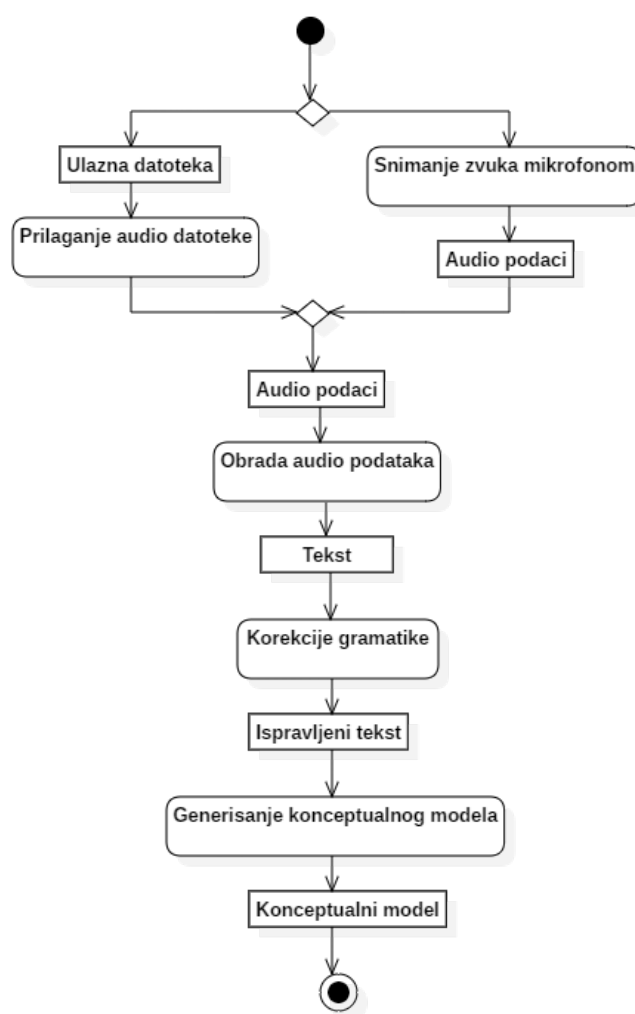
4.1. Razvoj konceptualnog modela baze podataka na osnovu govora

Automatizovano projektovanje baze podataka se pokazalo kao odličan način za projektovanje baze podataka bez potrebe za opsežnim znanjem kako bi se generisala šema baze podataka. Varijanta automatizovanog projektovanja koja se oslanja na govor predstavlja najprirodniji način generisanja konceptualnog modela, jer je često govor prvi korak u prikupljanju korisničkih zahtjeva, na osnovu kojih se izdvajaju najbitnije informacije potrebne za projektovanje. Iako je ovakav način najprirodniji, on ima brojne nedostatke u odnosu na ostale varijante ili na manuelno projektovanje. Baze podataka se najčešće projektuju na maternjem jeziku članova projektnog tima. Kako je automatizovano projektovanje baze podataka na osnovu govora zasnovano na upotrebi kompleksnih alata za prepoznavanje govora, ono zahtijeva postojanje alata koji može prepoznati govor za mnoge jezike, dok ovi alati često podržavaju samo jedan jezik, najčešće engleski.

Korišćenje alata koji podržava samo jedan ili manji broj jezika može značiti neupotrebljivost od strane korisnika koji ne poznaju te jezike, što predstavlja veliko ograničenje. Trenutno postoji veoma mali broj alata koji podržavaju prepoznavanje govora za veliki broj jezika.

Proces automatizovanog generisanja konceptualnog modela na osnovu govora reprezentovan je UML dijagramom aktivnosti na sl. 4.1. Na početku je potrebno prikupiti audio podatke kroz prilaganje audio datoteke ili putem mikrofona. Podaci se obrađuju i kao rezultat se dobija tekst. Alati za prepoznavanje govora mogu pogriješiti u prepoznavanju govora. Takođe, prilaganje audio datoteke lošeg kvaliteta ili korišćenje

lošeg mikrofona, smanjuju kvalitet zvuka što dovodi do pogrešnog prepoznavanja govora. Upotrebom alata za provjeru gramatike mogu se ispraviti greške u dobijenom tekstu. Ovaj korak je opcioni, ali može pomoći generisanju tačnijeg konceptualnog modela. Dobijeni tekst se analizira upotrebom NLP (eng. *Natural Language Processing*) tehnika koji izdvajaju najbitnije elemente teksta. Ovi elementi se obrađuju, te se svakoj jezičkoj jedinici pridružuje njena gramatička oznaka, koja se koristi prilikom generisanja konceptualnog modela. Jezičke jedinice koje su izdvojene na osnovu njihovih gramatičkih oznaka se analiziraju i utvrđuju se elementi konceptualnog modela, tj. entitetski tipovi, njihove međusobne veze i atributi. Nakon izdvajanja bitnih informacija iz teksta, generiše se odgovarajući konceptualni model.



Slika 4.1: Generisanje konceptualnog modela na osnovu govora

Čak i kada se koriste alati koji podržavaju veliki broj jezika, dobijeni tekst bi se trebao analizirati kako bi se izdvojili osnovni elementi teksta sa ciljem generisanja konceptualnog modela. Alati za analizu teksta takođe najčešće podržavaju samo engleski jezik, što predstavlja dodatnu prepreku. Jedan od načina prevazilaženja ove prepreke je prevodom dobijenog teksta na jedan od jezika koji alat za generisanje konceptualnog

modela može obraditi, a zatim naknadnim prevođenjem entiteta i atributa na prvobitni jezik dobija se konceptualni model na odgovarajućem jeziku.

Problem može nastati ukoliko prevod nije tačan ili ukoliko se izgubi smisao prilikom prevođenja. Iz ovih razloga najbolje je koristiti alat koji podržava višejezičnost, što najčešće nije slučaj. Prvi alat za generisanje konceptualnog modela na osnovu teksta je *TexToData*, koji upravo koristi ovakav princip, odnosno prevođenje teksta na engleski, te korišćenje alata za generisanje konceptualnog modela na osnovu teksta na engleskom jeziku. Nakon dobijanja konceptualnog modela, riječi sa engleskog jezika se prevode na prvobitni jezik, nakon čega se korisniku prikazuje odgovarajući konceptualni model.

U radu je upravo izvršena implementacija višejezičnosti kao jednog od najvažnijih poboljšanja trenutnog *Speed* alata. Iako je višejezičnost implementirana na nivou prepoznavanja govora, dobijeni tekst se svakako treba prevesti na engleski jezik zbog upotrebe poslovne logike *TexToData* alata, koji podržava direktno generisanje konceptualnog modela samo na engleskom jeziku.

Speed je trenutno jedini alat koji nudi generisanje konceptualnog modela na osnovu govora. Razvijanjem ovih alata u budućnosti mogu se značajno smanjiti vrijeme i troškovi potrebni za projektovanje baze podataka. Sve dok se ne razviju napredni alati sa malom vjerovatnoćom nastanka greške, potrebno je da projektni timovi uključuju ljude koji imaju veliko znanje i iskustvo u projektovanju baza podataka.

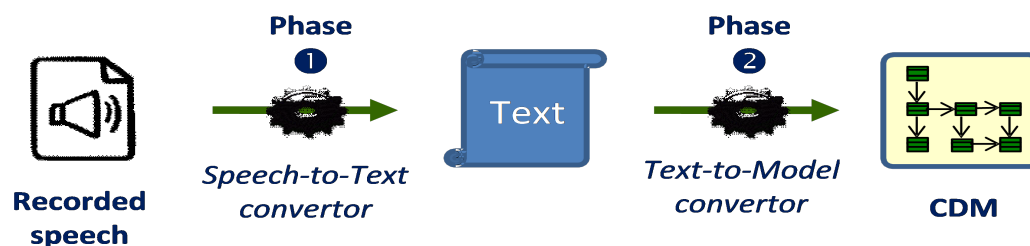
Ovi alati mogu poslužiti za automatsko generisanje inicijalnog modela baze podataka, prije ručne provjere dobijenog konceptualnog modela, nakon čega se u različitim fazama projektovanja ručno mogu izvršiti provjere i izmjene dobijenih rezultata. Kombinacija automatizovanog pristupa zajedno sa ručnom ispravkom garantuje brzinu, efikasnost i visok nivo integriteta projektovane baze podataka.

Prema [15], postoji veliki potencijal za korišćenje ovakvih alata u budućnosti koji omogućavaju generisanje konceptualnog modela na osnovu korisničkih priča koje su ranije snimljene, čime se pomaže agilnom razvoju softvera.

Prema [15], generisanje konceptualnog modela se izvršava u dva koraka:

1. snimljeni govor se konvertuje u odgovarajući tekst.
2. tekst se konvertuje u odgovarajući konceptualni model korišćenjem NLP alata koji koriste eksterne servise.

Proces sinteze konceptualnog modela zasnovanog na govoru prikazan je na sl. 4.2.



Slika 4.2: Proces sinteze konceptualnog modela zasnovan na govoru [35]

4.2. Obrada audio podataka

Prvi korak u generisanju konceptualnog modela na osnovu govora je obrada dobijenih audio podataka. Audio podaci se mogu dobiti prilaganjem audio datoteke ili govorom putem mikrofona u realnom vremenu. Postoji mnogo alata za obradu audio podataka, dok se ovaj rad primarno bavi Vosk alatom.

4.3. Provjera gramatike

Alati za prepoznavanje govora su podložni greškama. Pogrešno prepoznate riječi ili nedostatak znakova interpunkcije utiču na tačnost generisanog teksta. Posebno u slučaju automatskog projektovanja baza podataka, tačnost prepoznatog govora je od velikog značaja. Alati za provjeru gramatike mogu pomoći u ispravljanju teksta dobijenog od alata za prepoznavanje govora, te na ovaj način pomoći ispravnoj identifikaciji entiteta, njihovih atributa i međusobnih povezanosti. Ovi dodaci alatima za provjeru govora povećavaju tačnost prepoznatog teksta, što može značajno da utiče na kvalitet generisanog konceptualnog modela.

Kako bi se povećala tačnost prepoznatog govora, moguće je iskoristiti razne alate za provjeru gramatike i ispravnosti napisanih riječi. Dodatno, alati za prepoznavanje gramatike mogu ispraviti znakove interpunkcije u tekstu, što je od posebnog značaja jer alati za prepoznavanje govora često nemaju mogućnost prepoznavanja znakova interpunkcije.

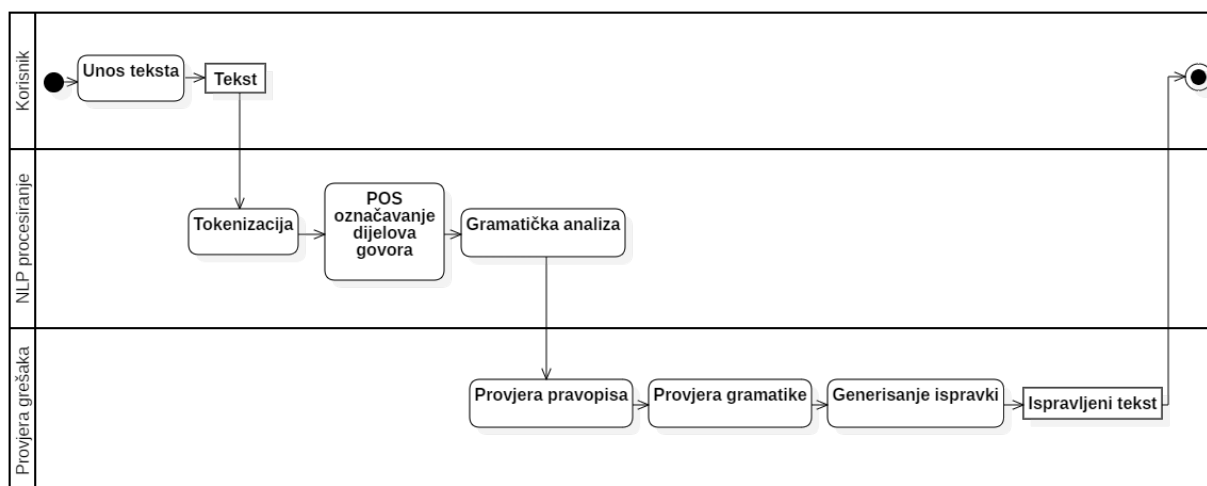
Male nepravilnosti u tekstualnoj specifikaciji mogu značajno uticati na konačno generisanu šemu baze podataka, te narušiti integritet i upotrebljivost baze podataka. Zbog ovih razloga, važno je pregledati i korigovati tekst nakon njegovog generisanja. Alati za provjeru gramatike i pravopisa mogu pomoći u ovom procesu, pružajući sugestije za ispravku teksta. Čak i u slučaju korišćenja ovih softverskih alata moguća je pojava greške. Na kraju procesa generisanja tekstualne specifikacije i upotrebe alata za ispravku grešaka potrebno je ručno pregledati tekst, kako bi se utvrdila ispravnost generisanih rezultata.

Postoje razni alati za provjeru gramatike, a neki od njih su: ProWritingAid, Jazzy i LanguageTool. Upotrebom ovih alata postiže se veća tačnost prepoznatog teksta, jer oni mogu na osnovu konteksta predložiti eventualne izmjene u riječima ukoliko neke riječi ne odgovaraju kontekstu.

Upotrebom alata za provjeru gramatike moguće je u realnom vremenu dobiti informacije o potencijalnim greškama i sugestije za njihove ispravke, ne samo za pojedinačne riječi nego i strukturu rečenice. Alati često nude više od jednog načina za ispravku teksta, te je manuelnom provjerom potrebno izabrati ispravan prijedlog. Ovi alati omogućavaju konzistentnost u napisanom tekstu, pogotovo kada je riječ o velikom tekstu da bi se održao isti stil pisanja, posebno u slučaju kad više autora doprinose istom tekstu.

4.3.1. Proces provjere gramatike

Osnovni cilj procesa provjere gramatike je identifikacija grešaka u tekstu i potencijalnih ispravki od strane alata za provjeru gramatike. Proces provjere gramatike zavisi od alata koji se koriste, dok je uopšteni proces prikazan na sl. 4.3.



Slika 4.3: Proces provjere gramatike

Proces provjere gramatike:

1. unos teksta – Proces provjere gramatike započinje unosom teksta u sistem za provjeru gramatike. Uneseni tekst predstavlja ulazni artefakt za dalju analizu.
2. tokenizacija – Tokenizacija, kao dio NLP-a, predstavlja proces dijeljenja teksta na manje segmente koji se zovu tokeni. Tokeni mogu predstavljati slova, riječi, fraze, čitave rečenice, znakove interpunkcije itd. Tokeni se kreiraju sa ciljem jednostavnije analize teksta bez gubitka konteksta.
3. POS označavanje dijelova govora – POS označavanje dijelova govora je proces gramatičkog označavanja riječi. Na primjer, riječ može biti označena kao imenica, glagol, pridjev itd. POS označavanjem dijelova govora sistem može razumjeti gramatičku ulogu svake riječi u tekstu.
4. gramatička analiza – Gramatičkom analizom se utvrđuje da li je rečenica strukturisana na ispravan način.
5. provjera pravopisa – U ovoj fazi se vrši provjera da li su riječi ispravno napisane. Provjerava se da li su slova ispravno napisana, da li postoje nepotrebni razmaci, kao i slične greške.
6. provjera gramatike – Provjerava da li se tekst pridržava gramatičkih pravila i standarda.
7. generisanje ispravki – Nakon završene analize teksta, sistem generiše prijedloge za ispravke. Na osnovu ovih prijedloga korisnik može odlučiti koje ispravke želi da primijeni na tekst.

4.3.2. Alati za provjeru gramatike i pravopisa

Postoje razni alati za provjeru gramatike i pravopisa, pri čemu mnogi pružaju obje funkcionalnosti. Međutim, s obzirom na to da Vosk alat za prepoznavanje govora uvijek uspješno prepoznaje riječi, odnosno rezultat prepoznavanja uvijek predstavlja ispravno napisane riječi, provjera pravopisa nije toliko važna za razliku od provjere gramatike. Provjera pravopisa je važna ukoliko se dobijeni tekst ručno mijenja, gdje postoji šansa za nastanak greške u pisanju. Alati i biblioteke za provjeru gramatike i pravopisa koji su korišćeni u okviru praktičnog dijela rada su:

1. Jazzy – Jazzy je *open-source* biblioteka za provjeru pravopisa. Omogućava provjeru tačnosti napisanih riječi, te ukoliko riječ nije pronađena u rječniku, Jazzy će analizirati i predložiti potencijalne izmjene. Napisan je u Java programskom jeziku u kojem je implementiran algoritam za prepoznavanje pogrešno napisanih riječi. Jednostavan je i lako ga je integrisati sa drugim aplikacijama [27].
2. ProWritingAid – ProWritingAid je alat za asistenciju u pisanju koji nudi brojne funkcionalnosti kao što su provjera pravopisa, provjera gramatike, analiza stila pisanja, te poboljšanje jasnoće teksta. Stil pisanja se odnosi na formalnost, izraze i ton, dok poboljšanje jasnoće osigurava jasnoću napisanog teksta. Provjera gramatike podrazumijeva prepoznavanje znakova interpunkcije i pogrešno napisanih riječi. Upotrebom ovog alata moguće je poboljšati stil pisanja, tako što alat na osnovu napisanog teksta nudi poboljšanja čitljivosti teksta upotrebljavajući alternativne riječi i fraze. Alat u ovom slučaju ističe mjesta na kojima je moguće poboljšati tekst i nudi korisniku izmjene postojećeg teksta. ProWritingAid pruža API, koji omogućava jednostavnu integraciju u različitim aplikacijama putem integracije sa ProWritingAid bibliotekom. Omogućena je provjera plagijacije na osnovu objavljenih radova skenirajući njihov sadržaj [26].
3. LanguageTool – LanguageTool je *open-source* alat za korekciju pravopisa i provjeru gramatike. LanguageTool je moćan alat koji omogućava provjeru grešaka u znakovima interpunkcije, strukture rečenica i korišćenih riječi. Takođe, alat kroz sugestije omogućava promjene stila rečenica radi poboljšanja jasnoće. Alat omogućava podršku za više od 25 jezika. Lako ga je integrisati sa različitim vrstama aplikacija. Posjeduje API kome se mogu slati zahtjevi za provjeru gramatike, kao i korišćenje ostalih funkcionalnosti. Kako se radi o alatu otvorenog koda, postoji široka zajednica korisnika koja pruža podršku alatu. U pitanju je moćan alat koji nudi preciznost i veliku tačnost, sa čestim ažuriranjima i poboljšanjima. Dodatno, alat nudi mogućnost parafraziranja teksta. Ovo može biti posebno korisno u dužim tekstovima u kojima je potrebno omogućiti konzistentnost i preciznost u pisanju, kako bi se koristile najprikladnije riječi. Uz doplatu, moguće je koristiti naprednije algoritme, kao što su metode vještačke inteligencije za provjeru teksta [28].
4. Sapling – Sapling je alat za provjeru gramatike i pravopisa koji uključuje i API. API omogućava provjeru gramatike za veći broj jezika, a neki od njih su kineski, italijanski i njemački. Sapling koristi metode vještačke inteligencije za provjeru

gramatike. API omogućava probnu verziju, te je po isteku potrebno izvršiti uplatu za korišćenje servisa. Dodatno, Sapling se može koristiti kao *self-hosted* rješenje, što omogućava potpunu kontrolu [45].

5. TextGears – TextGears je alat za provjeru gramatike i pravopisa koji takođe nudi API. Pored provjere gramatike, API pruža dodatne opcije kao što je analiza čitljivosti, detekcija jezika i analiza teksta. TextGears omogućava korišćenje probne verzije, te nakon je nakon isteka potrebno izvršiti uplatu za korišćenje API-ja [46].

4.4. Analiza teksta upotrebom NLP tehnika

Naredni korak je analiza teksta upotrebom NLP tehnika. Analiza teksta podrazumijeva razdvajanje teksta u manje jedinice koje se dalje analiziraju. Svakoj jedinici se dodjeljuju gramatičke oznake i određuju se zavisnosti između jedinica. Analizom jezičkih jedinica se mogu odrediti elementi konceptualnog modela. Na primjer, subjekat može predstavljati entitet u konceptualnom modelu. Međutim, alati koji implementiraju NLP tehnike bi trebali razumjeti kontekst kako bi se sa sigurnošću prepoznali ispravni elementi konceptualnog modela. Zbog ovog je važno korišćenje savremenih tehnika dubokog učenja u implementaciji NLP alata. Jedan od alata za izdvajanje karakteristika teksta je TextRazor [47] alat koji omogućava identifikaciju entiteta, koncepta i tema o kojima se raspravlja u tekstu, prepoznavanje veza između entiteta, prepoznavanje jezika itd.

4.5. Generisanje konceptualnog modela na osnovu teksta

Posljednji korak je generisanje konceptualnog modela na osnovu jezičkih jedinica. Dobijene jezičke jedinice iz prethodnog koraka se analiziraju kako bi se identifikovale osnovne jedinice konceptualnog modela kao što su entiteti, atributi i veze koje se kasnije vizuelno reprezentuju. Postoje razni alati za vizuelno predstavljanje konceptualnog modela i omogućavanje izmjena dobijenih modela. Automatizovano generisanje konceptualnog modela na osnovu govora je podložno greškama, zato je važno da alati za vizuelno predstavljanje modela omogućavaju dodatne opcije izmjena modela, kao što su izmjene nad entitetskim tipovima, njihovim međusobnim vezama i atributima. Često se konceptualni modeli predstavljaju u okviru raznih dokumenata, što znači da je važno da ovi alati imaju mogućnost eksportovanja dijagrama u grafičkim formatima. Alati u okviru istraživačke laboratorije M-lab koriste UML notaciju za reprezentaciju konceptualnog modela.

Glava 5

Implementacija pristupa

U ovoj glavi je detaljno opisana implementacija alata unaprijedene verzije Speed alata.

Unapređenje Speed alata je izvršeno reinženjeringom postojećih i dodavanjem novih komponenata, čime se poboljšava proces direktnog inženjeringa baze podataka na osnovu govora. Unapređenje je implementirano kroz tri funkcionalnosti:

1. prepoznavanje govora u *online* i *offline* režimima – funkcionalnost koja omogućava prepoznavanje ranije snimljenog govora ili govora koji se registruje mikrofonom u realnom vremenu.
2. višejezičnost – prepoznavanje govora je omogućeno za razne jezike.
3. provjera gramatike – Komponenta koja omogućava ispravku grešaka u tekstu, što rezultuje dobijanjem tačnijeg konceptualnog modela, odnosno šeme baze podataka.

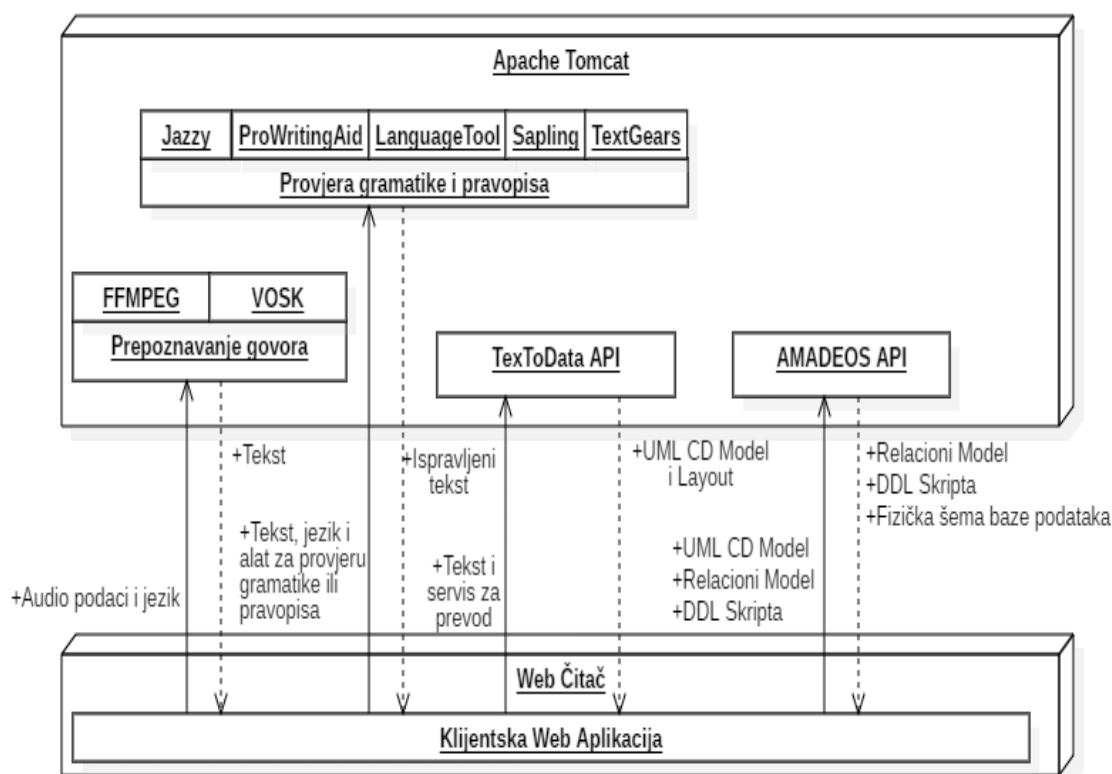
Speed omogućava direktni inženjering baze podataka, od analize govora pa sve do generisanja konkretne šeme baze podataka. Proces projektovanja baze podataka na osnovu govora započinje analizom teksta koji se dobija konvertovanjem audio podataka u tekst. Kako bi generisana šema baze podataka bila tačnija, uveden je mehanizam provjere dobijenog teksta upotrebom alata za provjeru gramatike i pravopisa. Dalji proces projektovanja se oslanja na TexToData alat koji omogućava dobijanje konceptualnog modela na osnovu teksta i AMADEOS alat za generisanje odgovarajućeg relacionog modela i DDL skripte. Nakon generisanja DDL skripte, alat omogućava generisanje fizičke baze podataka.

Komponente TexToData i AMADEOS alata koje omogućavaju da se u potpunosti ostvari automatizacija projektovanja baze podataka su:

1. generisanje konceptualnog modela;
2. generisanje relacionog modela;
3. generisanje DDL skripte.

Ovim komponentama se pristupa putem REST servisa. Pored ovih komponenata omogućena je izmjena dobijenih modela, kao i snimanje i dodavanje postojećih modela. Nakon generisanja DDL skripte moguće je generisati konkretnu šemu baze podataka za izabrani DBMS.

Arhitektura implementiranog sistema predstavljena je na sl. 5.1.



Slika 5.1: Arhitektura unaprijeđenog Speed sistema

FFMPEG je alat koji vrši konverziju između formata audio datoteka, kao i izmjenu drugih karakteristika kao što su broj uzoraka po sekundi, broj kanala i jačina zvuka. Speed koristi FFMPEG kako bi mijenjao karakteristike audio datoteka koje odgovaraju Vosk alatu za prepoznavanje govora.

Korisnici mogu izabrati pet alata za provjeru gramatike ili pravopisa, a to su Jazzy, ProWritingAid, LanguageTool, Sapling i TextGears.

Za prepoznavanje govora je potrebno proslijediti audio podatke i jezik, nakon čega se dobija odgovarajući tekst. Naredni korak je slanje teksta, jezika i naziva odabranog alata za provjeru gramatike ili pravopisa, nakon čega se dobija rezultat odabranog alata. Naredni koraci se izvršavaju putem API-ja kako bi se generisao konceptualni model, relacioni model, DDL skripta i fizička šema baze podataka.

5.1. Prepoznavanje govora

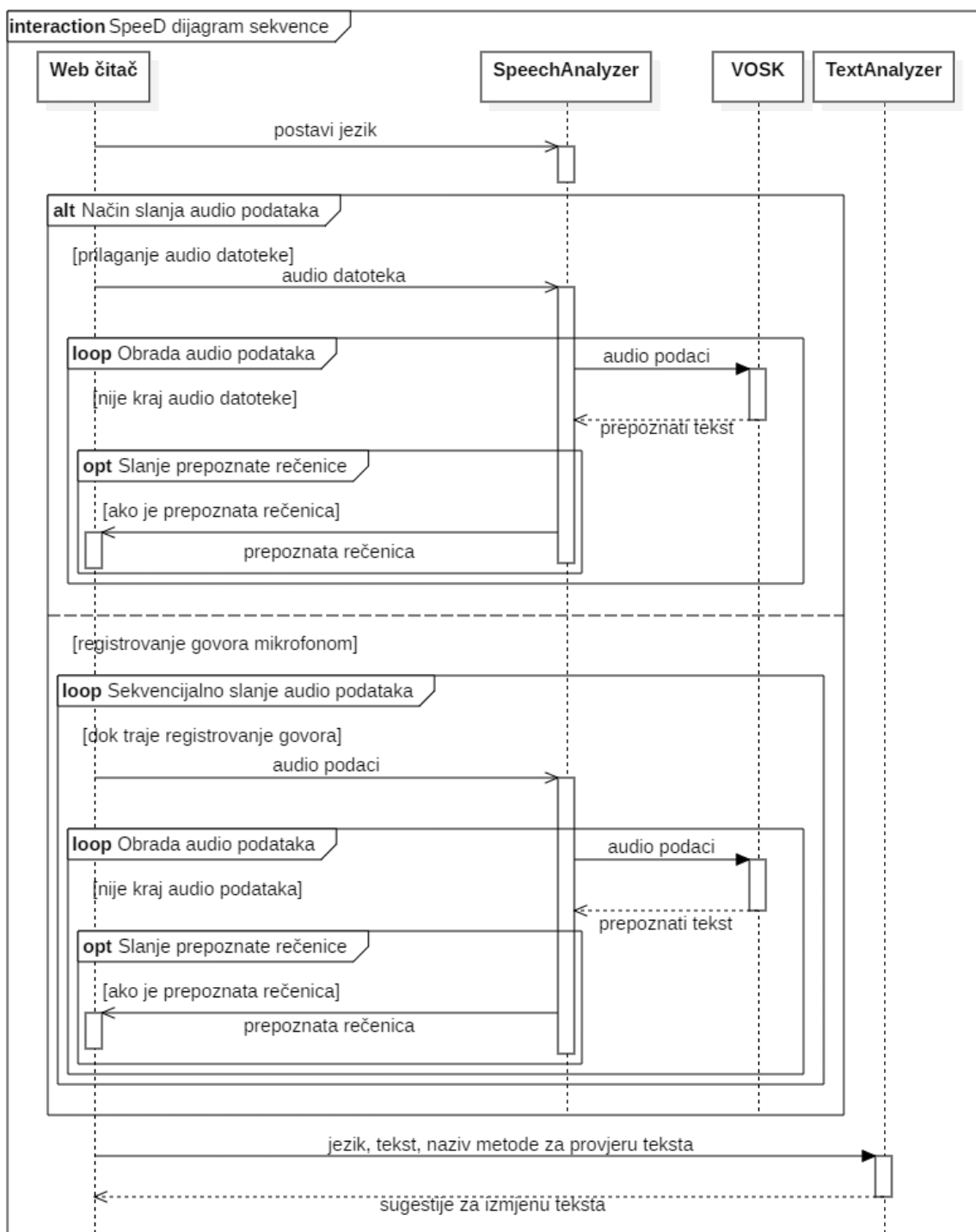
U implementiranom pristupu je moguće doći do audio podataka na dva načina:

1. prilaganjem audio datoteke;
2. registrovanjem govora u realnom vremenu.

U oba slučaja je važno odabrati jezik koji je korišćen u audio podacima kako bi se prepoznavanje izvršilo na ispravan način.

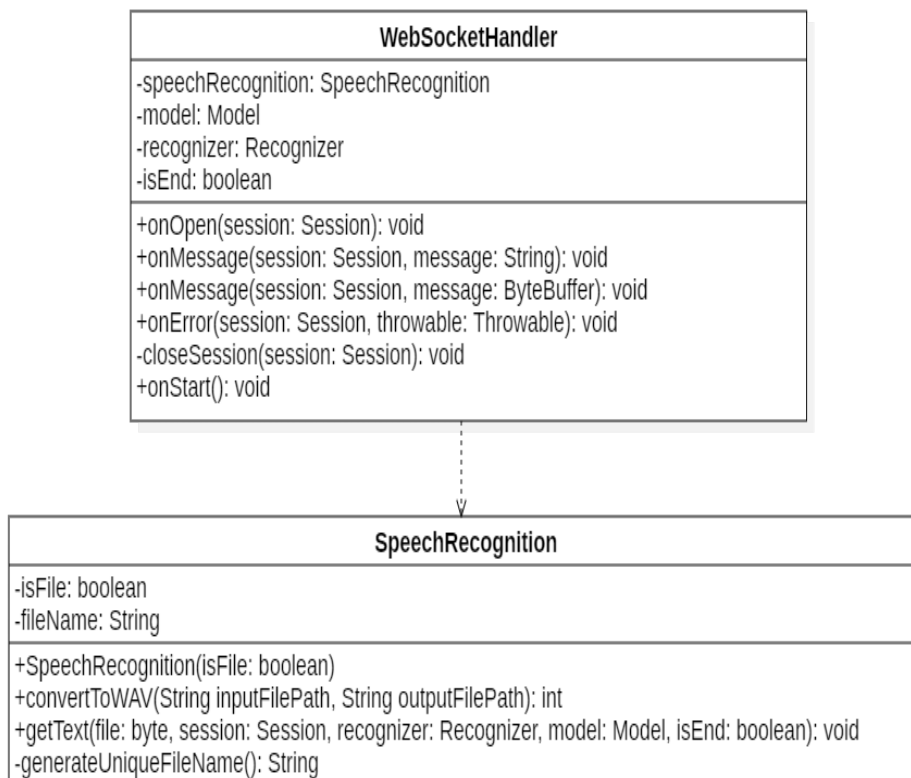
Funkcionalnost prepoznavanja govora je implementirana za razne jezike upotrebom Vosk alata u Java programskom jeziku. Podrška za nove jezike može se dodati

dodavanjem novih modela u posebnu datoteku i ažuriranjem konfiguracione datoteke. Na sl. 5.2 prikazan je proces prepoznavanja govora i analize teksta u unaprijedom Speed alatu.



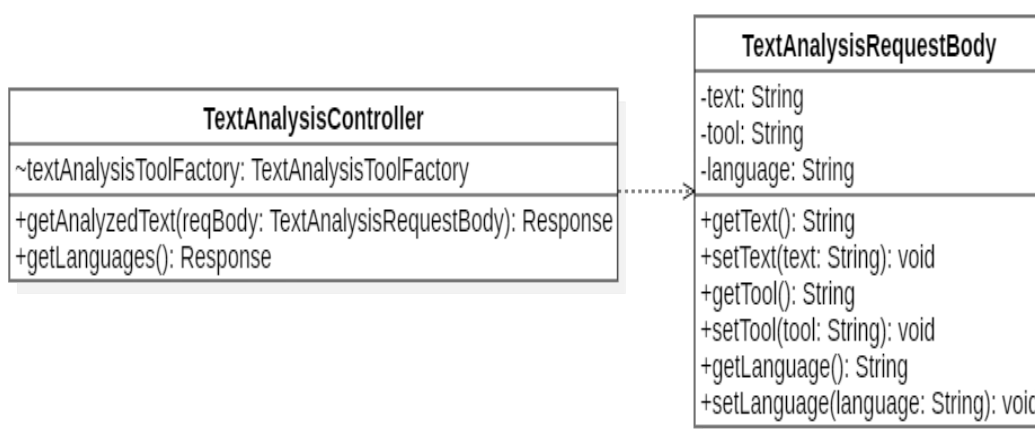
Slika 5.2: Proces analize teksta u unaprijedom Speed alatu

Na sl. 5.3 prikazan je dijagram klasa unaprijeđenog Speed alata sa klasama zaduženim za prepoznavanje govora. Prepoznavanje govora je realizovano upotrebom *web socket*-a kako bi se omogućilo dobijanje rezultata u realnom vremenu.



Slika 5.3: Prikaz klasa korišćenih za prepoznavanje govora

Speed koristi jednostavan kontroler za definisanje krajnje tačke za provjeru gramatike i pravopisa, što je prikazano na sl. 5.4.



Slika 5.4: Prikaz klasa korišćenih za definisanje krajnje tačke za provjeru gramatike i pravopisa

5.1.1. Prilaganje audio datoteke

Prilaganje audio datoteke podrazumijeva korišćenje unaprijed pripremljene audio datoteke u kojoj je izgovorena specifikacija zahtjeva. Prilaganjem audio datoteke kroz klijentsku aplikaciju vrši se slanje datoteke na servis koji vraća odgovarajući tekst. Pozitivna strana ovog pristupa je da se uz pomoć unaprijed snimljenih datoteka, na primjer korisničkih priča, može izvršiti projektovanje baze podataka. Loša strana je u tome što je potrebno posjedovati audio datoteku, što često nije slučaj. Iz ovog razloga je uveden dodatni mehanizam projektovanja baze podataka na osnovu govora, a to je registrovanjem govora u realnom vremenu putem mikrofona. Dodatan nedostatak ove metode je u veličini priložene audio datoteke. Ukoliko datoteka zauzima mnogo prostora, može doći do sporog slanja datoteke a samim tim i usporenog prepoznavanja govora.

5.1.2. Registrovanje govora u realnom vremenu

Registrovanjem govora u realnom vremenu i periodičnim slanjem podataka na serversku stranu, omogućeno je prepoznavanje govora. Za razliku od prilaganja audio datoteke, u ovom slučaju je potrebno periodično slati podatke dobijene putem mikrofona. U implementaciji alata sa klijentske strane se svake sekunde¹ šalju podaci sa mikrofona. Ovaj način dobijanja tekstualne specifikacije se može koristiti kao zamjena za prilaganje audio datoteke ili kao dodatak.

Vosk alat je namijenjen za *offline* prepoznavanje, međutim moguće ga je prilagoditi za prepoznavanje govora u realnom vremenu. U oba slučaja prepoznavanje govora i dobijanje rezultata je realizovano upotrebom *web socket*-a. Na sl. 5.5 prikazan je primjer komunikacije putem *web socket*-a, odnosno odgovori koji se dobijaju nakon prilaganja audio datoteke.

Data	Length	Time
↑ file	4	10:25:49.319
↑ Binary Message	1.1 MB	10:25:49.323
↓ Library members are students or faculty employees.	51	10:25:50.410
↓ Library members borrow library units student steady one of the programs.	73	10:25:50.831
↓ Students have index number and limit of library units borrowed.	64	10:25:51.173
↓ Library member has id name address telephone and number of library units borrowed.	83	10:25:51.603
↓ Faculty employee has room and phone.	37	10:25:51.793
↓ Library unit has you need tag and available information.	57	10:25:52.027
↓ Library units or magazines or books.	37	10:25:52.200
↓ Library unit has name year and author.	39	10:25:52.389
↓ Magazines have number magazines and books of iss in.	53	10:25:52.643
↓ Publisher his name and residents publisher publishes library units.	68	10:25:52.925

Slika 5.5: Primjer rezultata komunikacije putem *web socket*-a

Komunikacija putem *web socket*-a je implementirana kroz klijentsku stranu u JavaScript kodu i serversku u okviru Java programskog jezika. Serverska aplikacija

¹ Audio podaci se šalju svake sekunde kako bi prepoznavanje teksta bilo što bliže realnom vremenu, omogućavajući korisniku da brže dobije rezultat, a empirijski je utvrđeno da ovaj interval dovoljno dobro balansira brzinu i opterećenje servera.

prima odabrani jezik i audio podatke, nakon čega šalje klijentu odgovore koji predstavljaju prepoznate rečenice.

Na osnovu proslijeđenog jezika upotrebljava se odgovarajući jezički model. U slučaju odabira opcije za registrovanje govora u realnom vremenu, audio podaci se periodično šalju sa klijentske na serversku stranu, nakon čega se na serverskoj strani analiziraju na identičan način kao u slučaju sa prilaganjem audio datoteke. U ovom slučaju, audio podaci se šalju svake sekunde.

Nakon što se prepozna rečenica, server je vraća klijentu. U svakom trenutku je moguće prekinuti registrovanje govora nakon čega se šalje odgovarajuća poruka o završetku komunikacije na serversku stranu.

5.2. Provjera gramatike

Druga implementirana komponenta je komponenta za provjeru gramatike. Alati za provjeru gramatike mogu pogriješiti u prepoznavanju. Kako bi se implementirala baza podataka koja najbolje odgovara potrebama korisnika, važno je ispraviti pogrešno prepoznat tekst. Alati za provjeru teksta, pored provjere tačnosti napisanih riječi, često imaju mogućnost provjere nedostataka u znakovima interpunkcije i razumijevanja konteksta na osnovu kojeg se mogu izdvojiti sugestije za poboljšanje teksta.

U implementiranom alatu je omogućeno korišćenje pet softverskih komponentata za provjeru gramatike. Ove komponente se razlikuju u tačnosti i načinu na koji implementiraju funkcionalnost provjere gramatike. Neki od njih su jednostavniji, odnosno imaju mogućnost provjere tačnosti riječi na osnovu rječnika, dok drugi rade sa kompleksnim alatima umjetne inteligencije i imaju mogućnost prepoznavanja konteksta. Dodatno, neki od njih omogućavaju funkcionalnost samo na engleskom jeziku, dok drugi podržavaju veći broj jezika.

Na sl. 5.6 je prikazana krajnja tačka za analizu teksta. Krajnja tačka prima JSON objekat sa jezikom i nazivom odabrane softverske komponente. Naziv se koristi kako bi se instancirao objekat odgovarajuće klase za analizu teksta. Rezultat metode za analizu teksta se vraća na klijentsku stranu.

```
@POST
@Path("/analyzetext")
@Consumes(MediaType.APPLICATION_JSON)
@Produces(MediaType.APPLICATION_JSON)
public Response getAnalyzedText(TextAnalysisRequestBody reqBody)
{
    TextAnalysisTool textAnalysisTool;
    try {
        textAnalysisTool = TextAnalysisToolFactory.getTool(reqBody.getTool(), reqBody
        .getLanguage());
    } catch (IllegalArgumentException e) {
        return Response.status(Response.Status.BAD_REQUEST)
            .entity("Invalid tool specified: " + e.getMessage()).build();
    }
    String result = textAnalysisTool.analyzeText(reqBody.getText());
    return Response.ok(result).build();
}
```

Slika 5.6: Krajnja tačka za analizu teksta

Analiza teksta je bitna kako bi se dobila što tačnija šema baze podataka. Za prepoznavanje gramatičkih grešaka potrebno je prethodno odabrati jezik na kojem je napisan tekst. Alati često podržavaju samo jedan jezik, najčešće engleski. Analiza teksta je implementirana upotrebom pet različitih komponenata za provjeru gramatike i pravopisa, a to su:

1. Jazzy – Speed omogućava prepoznavanje pravopisnih grešaka upotrebom Jazzy biblioteke. Na ovaj način nije potrebno korišćenje eksternih servisa za dobijanje sugestija o unapređenju teksta. Na sl. 5.7 prikazana je klasa u kojoj se koristi Jazzy biblioteka za analizu teksta. Klasa implementira *SpellCheckListener* interfejs, što zahtijeva implementaciju određenih metoda. *JazzySpellChecker* klasa koristi *SuggestionResult* klasu za definisanje odgovora koji se prosljeđuje klijentu.

```
package org.unibl.etf.speed.textanalysis;

import java.io.File;

public class JazzySpellChecker implements SpellCheckListener, TextAnalysisTool {
    private SpellChecker spellChecker;
    private List<SuggestionResult> suggestionResults;
    private SpellDictionaryHashMap dictionaryHashMap;

    public JazzySpellChecker() {
        suggestionResults = new ArrayList<>();
        File dict = new File(Config.DICTIONARY);
        try {
            dictionaryHashMap = new SpellDictionaryHashMap(dict);
        } catch (IOException e) {
            e.printStackTrace();
        }
        initialize();
    }

    private void initialize() {
        spellChecker = new SpellChecker(dictionaryHashMap);
        spellChecker.addSpellCheckListener(this);
    }

    @Override
    public void spellingError(SpellCheckEvent event) {...}

    public List<String> getSuggestions(String misspelledWord) {...}

    public JSONArray getMisspelledWordsJSON(String text) {...}

    public String analyzeText(String text) {...}
}
```

Slika 5.7: Prikaz JazzySpellChecker klase

2. LanguageTool – Upotrebom LanguageTool biblioteke ostvarena je funkcionalnost provjere gramatike. Na sl. 5.8 prikazan je dio koda koji koristi LanguageTool biblioteku za analizu govora. Upotrebom kratkog koda, na osnovu kojeg se dolazi do jezika, kreira se objekat koji se koristi za prepoznavanje grešaka teksta. Nakon poziva odgovarajuće metode, dobija se lista objekata koja pruža informacije o predloženim izmjenama teksta, uključujući početne i krajnje pozicije u tekstu za svaku izmjenu. Ovi podaci se obrađuju kako bi se poslali na klijentsku stranu

u odgovarajućem formatu. Podaci se nakon slanja klijentskoj strani ugrađuju u tekstualni element.

```
JLanguageTool languageTool = new JLanguageTool(Languages.getLanguageForShortCode(
    languageShortCode));
List<RuleMatch> matches = languageTool.check(text);
```

Slika 5.8: Prikaz LanguageTool koda za analizu teksta

3. ProWritingAid – Predstavlja *online* alat za provjeru gramatike. U parametrima je potrebno pored kratkog koda jezika unijeti i API ključ. ProWritingAid pruža biblioteku za jednostavniju komunikaciju i dobijanje analiziranog teksta. Na sl. 5.9 prikazan je dio koda koji poziva ProWritingAid krajnju tačku koja vraća predložene izmjene. Prolaskom kroz izmjene u nastavku se kreira lista JSON objekata sa predloženim izmjenama koje se vraćaju klijentu.

```
TextApi apiInstance = new TextApi();
AsyncResponseTextAnalysisResponse result = apiInstance.post(request);
List<DocTag> tags = result.getResult().getTags();
```

Slika 5.9: Prikaz ProWritingAid koda za analizu teksta

4. Sapling – Pored raznih drugih krajnjih tačaka koje ovaj alat podržava, za provjeru gramatike je korišćena krajnja tačka <https://api.sapling.ai/api/v1/edits>, odabirom HTTP POST zahtjeva i slanjem tijela zahtjeva u JSON (eng. *JavaScript Object Notation*) formatu u kojem se nalazi ključ, tekst, jezik i identifikator sesije. Na sl. 5.10 prikazan je primjer tijela zahtjeva u JSON formatu.

```
{
  "key": "key_example",
  "text": "A student is a person within the system each person has a name each course
  has a title, type, description, professor and assistant. Students enroll in courses."
  ,
  "lang": "en",
  "session_id": "test_session"
}
```

Slika 5.10: Sapling tijelo HTTP zahtjeva

Nakon slanja zahtjeva, krajnja tačka vraća odgovor koji je prikazan na sl. 5.11. Odgovor se parsira u okviru Java programskog jezika iz kojeg se izdvajaju najvažnije informacije kao što su sugestije i pozicije sugestija u tekstu. Za svaku prepoznatu grešku teksta, Sapling pruža samo jednu sugestiju za ispravku. Ove informacije služe kako bi se kreirao odgovor koji klijentska strana može koristiti za prikaz sugestija korisniku. Sapling može prepoznati razne tipove grešaka. U navedenom primjeru su prikazana dva različita tipa grešaka, a to je pogrešan ili nedostatak člana i greška u znaku interpunkcije.

```

{
  "edits": [
    {
      "description": "Incorrect/missing article.",
      "end": 1,
      "error_type": "U:DET:ART",
      "general_error_type": "Grammar",
      "id": "8e91dcc2-2abd-5b4a-bc7d-c1f8eadba719",
      "replacement": "Each",
      "sentence": "A student is a person within the system each person has a name
each course has a title , type, description , professor and assistant.",
      "sentence_start": 0,
      "start": 0
    },
    {
      "description": "Punctuation error.",
      "end": 39,
      "error_type": "M:PUNCT",
      "general_error_type": "Punctuation",
      "id": "0ee29eb2-b3c3-5146-9762-626d69554802",
      "replacement": "system,",
      "sentence": "A student is a person within the system each person has a name
each course has a title , type, description , professor and assistant.",
      "sentence_start": 0,
      "start": 33
    },
    {
      "description": "Punctuation error.",
      "end": 62,
      "error_type": "M:PUNCT",
      "general_error_type": "Punctuation",
      "id": "6474d3d9-1ce5-5a7a-9c62-4bbafdfa19d3",
      "replacement": "name,",
      "sentence": "A student is a person within the system each person has a name
each course has a title , type, description , professor and assistant.",
      "sentence_start": 0,
      "start": 58
    }
  ]
}

```

Slika 5.11: Sapling odgovor HTTP zahtjeva

5. TextGears – Kako bi se dobile sugestije, potrebno je poslati HTTP POST zahtjev na krajnju tačku <https://api.textgears.com/analyze>. Tijelo HTTP zahtjeva se šalje u *form data* formatu. Kroz *form data* se šalju dva ključa, a to su *text* sa vrijednošću "A student is a person within the system each person has a name each course has a title type description professor and and assistant. Students enroll in courses." i *key* koji predstavlja API ključ. Na sl. 5.12 je prikazno tijelo HTTP zahtjeva u JSON formatu.

```

{
  "text": "A student is a person within the system each person has a name each course has
a title type description professor and and assistant. Students enroll in courses.",
  "tool": "TextGears",
  "language": "en"
}

```

Slika 5.12: TextGears tijelo HTTP zahtjeva

Odgovor se parsira u okviru programskog koda iz kojeg se izvlače informacije o sugestijama i mjestima na kojima je potrebno izmijeniti tekst. Komponenta je

implementirana kroz krajnju tačku kojom se šalje tekst, naziv alata za provjeru gramatike i jezik. Krajnja tačka se nalazi na adresi <http://m-lab.etf.unibl.org:8080/Speed/rest/analyzetext>.

Slanjem POST zahtjeva sa ova tri parametra dobija se odgovor koji je prikazan na sl. 5.13.

```
{
  "status": true,
  "response": {
    "grammar": {
      "result": true,
      "errors": [
        {
          "id": "e1761086621",
          "offset": 33,
          "length": 11,
          "description": {
            "en": "Did you mean system . Each ?"
          },
          "bad": "system each",
          "better": [
            "system. Each"
          ],
          "type": "grammar"
        }, ...
      ]
    },
    "stats": {
      "fleschKincaid": {
        "readingEase": 65.7,
        "grade": "8th or 9th grade",
        "interpretation": "Plain text. Easily understood by 13- to 15-year-old students"
      },
      "gunningFog": 9.9,
      "colemanLiau": 11.7,
      "SMOG": 10.3,
      "vocabularySize": {
        "active": null,
        "passive": null
      },
      "emotion": {
        "negative": 0.5081063473534495,
        "positive": 0.49189365264655044
      },
      "counters": {
        "length": 160,
        "clearLength": 133,
        "words": 28,
        "sentences": 2
      }
    }
  }
}
```

Slika 5.13: TextGear odgovor HTTP zahtjeva

Na sl. 5.14 prikazan je primjer odgovora krajnje tačke Speed-a na zahtjev za provjeru gramatike. Krajnja tačka vraća niz JSON objekata sa četiri podatka, a to su originalni tekst, sugestije kao niz podataka tipa *string*, početak i kraj zamjene u početnom tekstu. Brojevi koji predstavljaju početak i kraj predstavljaju broj karaktera od kojih počinje, odnosno završava tekst koji se može zamijeniti sa sugestijom iz niza. Ovim je na jedinstven način implementiran odgovor krajnje tačke za prepoznavanje grešaka teksta, nezavisno od korišćenog alata.

```
[
  {
    "original": "system each",
    "start": 33,
    "end": 44,
    "suggestions": [
      "system. Each"
    ]
  },
  {
    "original": "name each",
    "start": 136,
    "end": 145,
    "suggestions": [
      "name. Each"
    ]
  },
  {
    "original": "professor",
    "start": 260,
    "end": 269,
    "suggestions": [
      ": professor"
    ]
  }
]
```

Slika 5.14: Odgovor HTTP zahtjeva za provjeru teksta

5.3. Integracija sa TexToData i AMADEOS alatima

Speed alat je integrisan sa TexToData i AMADEOS alatima kako bi se omogućilo izvršavanje kompletnog procesa projektovanja baze podataka na osnovu govora. Nakon što se generiše odgovarajući tekst i isprave greške kroz implementirane komponente, dalji proces projektovanja baze podataka se oslanja na logiku ovih alata. Nakon što se generiše tekst, potrebno je proći kroz tri koraka kako bi se generisala konkretna šema baze podataka a to su:

1. generisanje konceptualnog modela – Nakon generisanja teksta, odabirom odgovarajuće opcije moguće je kreirati konceptualni model. Krajnja tačka koja se koristi za dobijanje konceptualnog modela je <http://m-lab.etf.unibl.org:8080/Textodata/rest/modeler>, dok je primjer tijela HTTP POST zahtjeva prikazan na sl. 5.15.

```
{
  "text": "A student is a person within the system. Each person has a name. Each course has a title type description : professor and and assistant. Students enroll in courses.",
  "translationService": "yandex",
  "fromLang": null
}
```

Slika 5.15: HTTP tijelo zahtjeva za generisanje konceptualnog modela

Kao odgovor, krajnja tačka servisa vraća odgovarajuće podatke koji se koriste za vizuelizaciju entiteta, atributa i njihovih veza. Primjer odgovora je prikazan na sl. 5.16. Odgovor vraća prosljeđeni tekst, zajedno sa jezikom originalnog

teksta i informacijama u *xml* formatu potrebnim za vizuelno prikazivanje konceptualnog modela. Sistem omogućava izmjenu dobijenog modela kao što je izmjena povezanosti klasa kroz upotrebu agregacije i kompozicije. Dodatno, Speed omogućava dodavanje novih i uklanjanje postojećih entiteta, kao i izmjenu atributa, operacija i relacija kao što su asocijacija, agregacija, kompozicija i generalizacija.

```
{
  "englishText": "A student is a person within the system. Each person has a name. Each
  course has a title type description : professor and and assistant. Students enroll in
  courses.",
  "layout": "<?xml version=\"1.0\" encoding=\"UTF-8\" standalone=\"no\"?><notation:
  Diagram xmlns:notation=\"http://www.eclipse.org/gmf/runtime/1.0.2/notation\", ...",
  "mxGraphUml": "<mxGraphModel dx=\"1038\" dy=\"612\" grid=\"1\" gridSize=\"10\" guides
  =\"1\" tooltips=\"1\" connect=\"1\" arrows=\"1\" fold=\"1\" page=\"1\", ...",
  "originalText": "A student is a person within the system. Each person has a name. Each
  course has a title type description : professor and and assistant. Students enroll in
  courses.",
  "originalTextLang": "en",
  "uml": "<?xml version=\"1.0\" encoding=\"UTF-8\" standalone=\"no\"?> <uml:Model xmlns:
  uml=\"http://www.eclipse.org/uml2/5.0.0/UML\" name=\"ICM\", ..."
}
```

Slika 5.16: Odgovor na HTTP zahtjev za generisanje konceptualnog modela

2. generisanje relacionog modela – Naredni korak je generisanje odgovarajućeg relacionog modela na osnovu konceptualnog modela. Za ovu funkcionalnost iskorištena je krajnja tačka <http://m-lab.etf.unibl.org:8080/amadeos/services/generate/composed/cmd2rdm>. Primjer odgovora HTTP zahtjeva je prikazan na sl. 5.17. Odgovor se vraća u JSON formatu sa podacima koji se prikazuju na klijentskoj strani kroz vizuelnu reprezentaciju. Nakon generisanog relacionog modela, moguće je mijenjati model po potrebama.

```
{
  "uml": "<?xml version=\"1.0\" encoding=\"UTF-8\"?> <uml:Model xmi:version=\"20131001\"
  xmlns:xmi=\"http://www.omg.org/spec/XMI/20131001\" xmlns:uml=\"http://www.eclipse.org
  /uml2/5.0.0/UML\", ...",
  "notation": "<?xml version=\"1.0\" encoding=\"UTF-8\" standalone=\"no\"?> <notation:
  Diagram xmlns:notation=\"http://www.eclipse.org/gmf/runtime/1.0.2/notation\" xmlns:
  style=\"http://www.eclipse.org/papyrus/infra/viewpoints/policy/style\", ...",
  "status": 0
}
```

Slika 5.17: Odgovor na HTTP zahtjev za generisanje relacionog modela

3. kreiranje DDL skripte – Nakon generisanog relacionog modela, moguće je generisati odgovarajuću DDL skriptu kroz krajnju tačku <http://m-lab.etf.unibl.org:8080/amadeos/services/generate/composed/rdm2ddl>. Primjer odgovora HTTP zahtjeva za generisanje DDL skripte nalazi se na sl. 5.18. Dobijena skripta se prikazuje u klijentsku aplikaciju. Nakon prikazivanja skripte, moguće je izmijeniti upite po potrebama.

```
{
  "script": "CREATE SCHEMA Model; CREATE TABLE Model.student ( id int NOT NULL, PRIMARY
  KEY (id) ); CREATE TABLE Model.course ( id int NOT NULL, professor varchar(15) NOT
  NULL, assistant varchar(15) NOT NULL, title_description varchar(15) NOT NULL, PRIMARY
  KEY (id) ); CREATE TABLE Model.course_enroll_student ( course_id int NOT NULL,
  student_id int NOT NULL, PRIMARY KEY (course_id, student_id) ); CREATE TABLE Model.
  person ( id int NOT NULL, name varchar(15) NOT NULL, PRIMARY KEY (id) ); ALTER TABLE
  Model.student ADD CONSTRAINT student_FK_person FOREIGN KEY (id) REFERENCES Model.
  person (id) ON DELETE RESTRICT ON UPDATE CASCADE; ALTER TABLE Model.
  course_enroll_student ADD CONSTRAINT course_enroll_student_FK_course FOREIGN KEY (
  course_id) REFERENCES Model.course (id) ON DELETE RESTRICT ON UPDATE CASCADE; ALTER
  TABLE Model.course_enroll_student ADD CONSTRAINT course_enroll_student_FK_student
  FOREIGN KEY (student_id) REFERENCES Model.student (id) ON DELETE RESTRICT ON UPDATE
  CASCADE; CREATE INDEX INDEX_student ON Model.student (id ASC); CREATE INDEX
  INDEX_course ON Model.course (id ASC); CREATE INDEX INDEX_course_enroll_student ON
  Model.course_enroll_student (course_id ASC, student_id ASC); CREATE INDEX
  INDEX_person ON Model.person (id ASC); ",
  "status": 0
}
```

Slika 5.18: Odgovor na HTTP zahtjev za generisanje DDL skripte

Nakon dobijanja DDL skripte, moguće je izvršiti generisanje fizičke baze podataka upotrebom krajnje tačke <http://m-lab.etf.unibl.org:8080/amadeos/services/generate/composed/ddl2database>.

5.4. Klijentska aplikacija

Klijentska *web* aplikacija unaprijeđenog Speed sistema sadrži četiri stranice, a to su:

1. *Speech and Text Analyzer* – analiza govora i teksta,
2. *Conceptual Data Model* – manipulacija konceptualnim modelom,
3. *Relational Data Model* – manipulacija relacionim modelom,
4. *DDL Script* – generisanje DDL skripte.

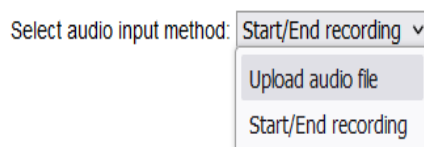
Na prvoj stranici korisnici kroz opadajuće menije mogu odabrati jezik, izvor audio podataka, metodu za analizu teksta i servis za prevod.

Padajući meni za odabir jezika prikazan je na sl. 5.19.



Slika 5.19: Padajući meni za odabir jezika

Klijentska aplikacija omogućava odabir između prilaganja audio datoteke i registrovanje govora u realnom vremenu kroz padajući meni prikazan na sl. 5.20. Odabirom opcije za registrovanje govora putem mikrofona korisniku se prikazuju dugmadi za početak i kraj snimanja govora. Odabirom opcije za početak snimanja govora periodično će se slati registrovani govor sa mikrofona na serversku stranu, sve dok se ne odabere opcija za završetak snimanja govora. U slučaju odabira opcije za prilaganje audio datoteke, nakon prilaganja automatski će se poslati datoteka na server.



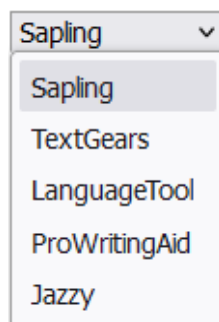
Slika 5.20: Metoda audio ulaza

Nakon dobijanja teksta, moguće je izabrati različite servise za prevod teksta na engleski jezik. Ovi servisi se koriste kako bi se izvršilo generisanje konceptualnog modela, jer analizirani tekst za utvrđivanje entiteta, njihovih međusobnih veza i atributa radi samo na engleskom jeziku. Padajući meni za izbor alata za prevođenje je prikazan na sl. 5.21.



Slika 5.21: Padajući meni za odabir servisa za prevod

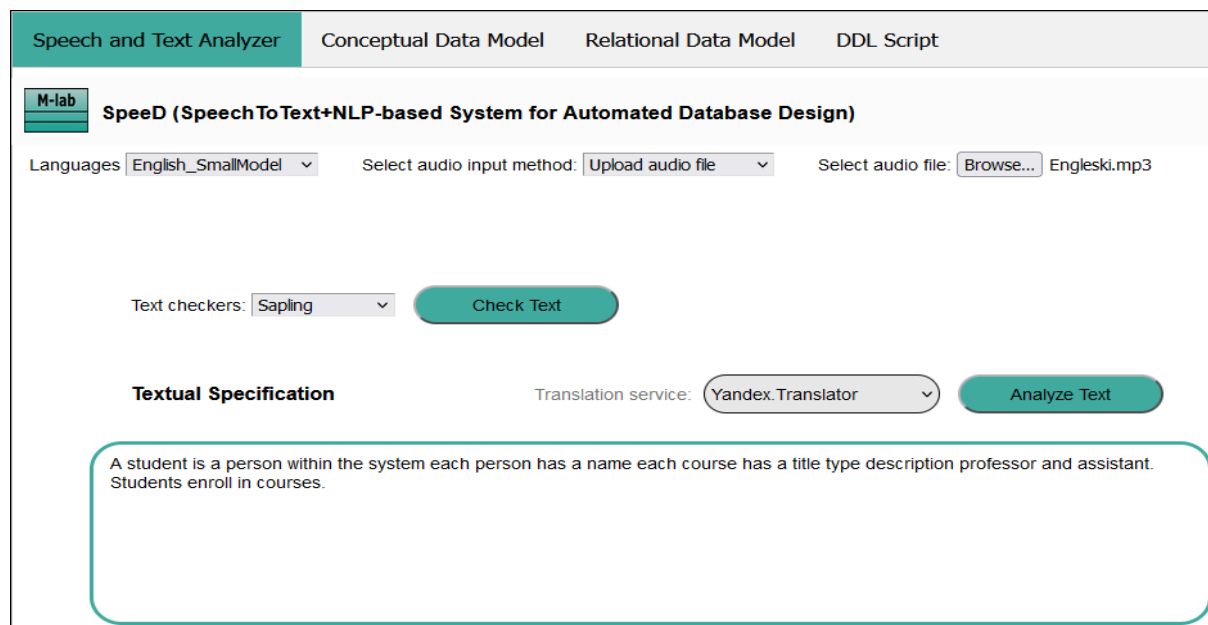
Klijentska aplikacija omogućava odabir između različitih komponenata za provjeru gramatike i pravopisa kroz padajući meni prikazan na sl. 5.22.



Slika 5.22: Padajući meni za odabir komponente za provjeru gramatike ili pravopisa

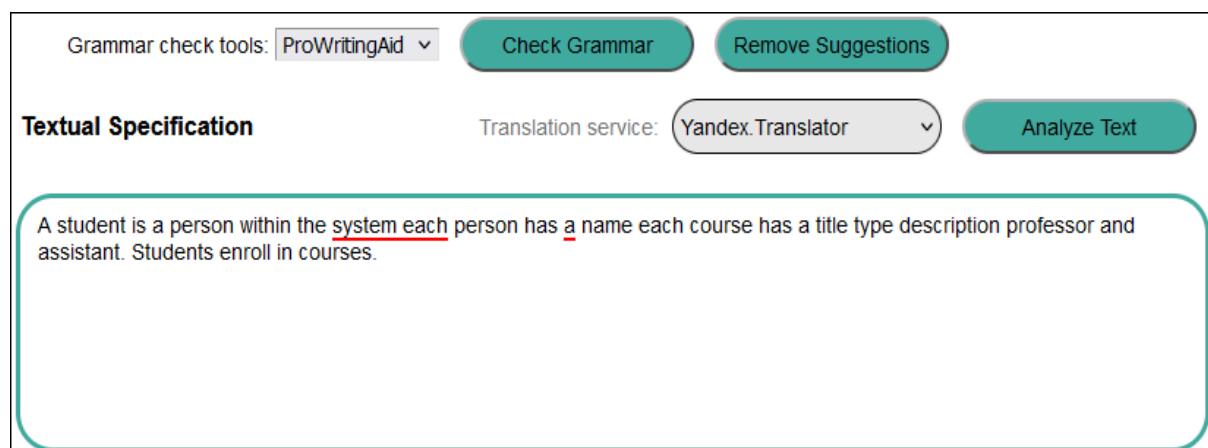
5.5. Ilustrativni primjer projektovanja baze podataka na osnovu govora

Na sl. 5.23 nalazi se primjer prepoznatog teksta nakon prilaganja datoteke "Engleski.mp3". U tekstualnom HTML elementu se nalazi odgovarajući tekst prepoznat na osnovu prirodnog govora iz audio datoteke.



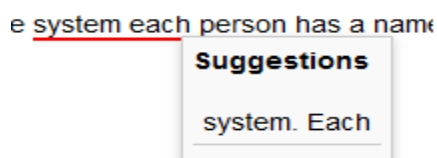
Slika 5.23: Početna stranica Speed alata nakon prepoznavanja govora

Na sl. 5.24 prikazan je izgled klijentske aplikacije nakon poziva ProWritingAid alata. Nakon što server vrati odgovor, klijentska strana podvlači tekst na osnovu pozicija iz odgovora, te na odabir prikazuje potencijalne izmjene. ProWritingAid je prepoznao dvije greške, te je podvučen dio teksta u kojem je alat prepoznao greške. Naredni koraci podrazumijevaju pregled prepoznatih grešaka i njihovo ispravljanje ukoliko je potrebno.



Slika 5.24: Rezultat alata za provjeru gramatike na prepoznati tekst

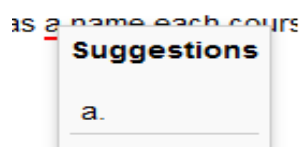
Na sl. 5.25 prikazane su sugestije. Sugestije su izlistane jedna ispod druge, a maksimalan broj sugestija je 50. Konkretno, tekst "system each" se može zamijeniti sa "system. Each". Alat je prepoznao da je u pitanju kraj rečenice, te da naredna rečenica počinje velikim početnim slovom.



Slika 5.25: Prikaz prve otkrivene greške i mogućnosti za izmjenu podvučenog teksta

Ispravka ove greške omogućava tačnije generisanje konceptualnog modela, što rezultuje tačnijom šemom baze podataka. U ovom primjeru alat prikazuje samo jednu sugestiju, dok najčešće postoji više sugestija za određeni tekst. Odabirom neke od sugestija tekst će se zamijeniti, a crvena linija ispod teksta će nestati.

Na sl. 5.26 prikazane su sugestije. U ovom primjeru alat je prepoznao da je u pitanju kraj rečenice, iako nije. Iz ovog razloga nije izvršena izmjena teksta. Odabirom opcije *Remove Suggestions* briše se podvučena linija, odnosno više nije moguće pogledati i zamijeniti tekst sugestijama.



Slika 5.26: Prikaz druge otkrivene greške i mogućnosti za izmjenu podvučenog teksta

Nakon prihvatanja sugestija dobija se tekst sa sl. 5.27. Nakon ispravke teksta, moguće je ponovo provjeriti da li postoje greške u tekstu koristeći druge dostupne metode. Način rada sa ostalim metodama za analizu teksta je isti kao u ovom primjeru.

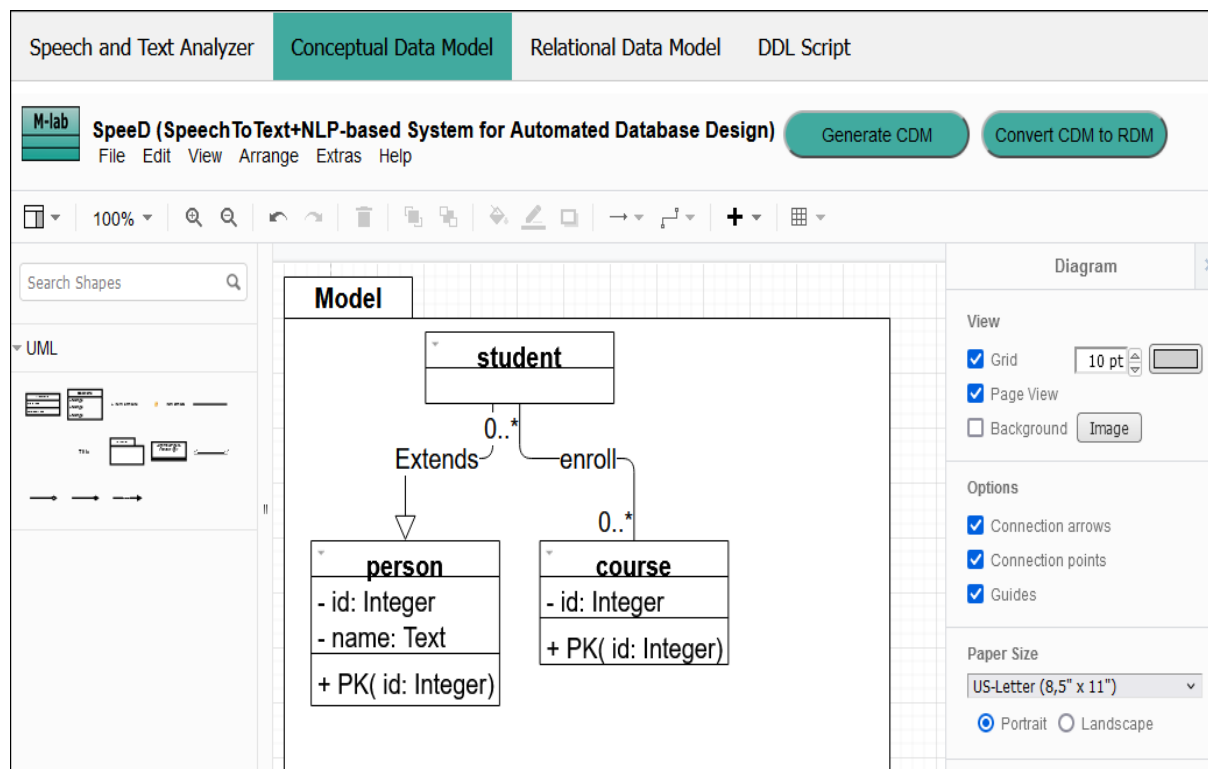
A student is a person within the system. Each person has a name each course has a title type description professor and assistant. Students enroll in courses.

Slika 5.27: Dobijeni tekst nakon ispravke upotrebom alata za provjeru gramatike

Nakon odabira opcije za generisanje konceptualnog modela, dobijamo model sa sl. 5.28. Konceptualni model prikazuje entitet *student* kojeg generalizuje entitet *person*. Za

entitet *person* prepoznavanje atributa nije izvršeno na ispravan način. Studenti se mogu prijaviti na nula ili više kurseva. Entitet *course* ima atribut *id*.

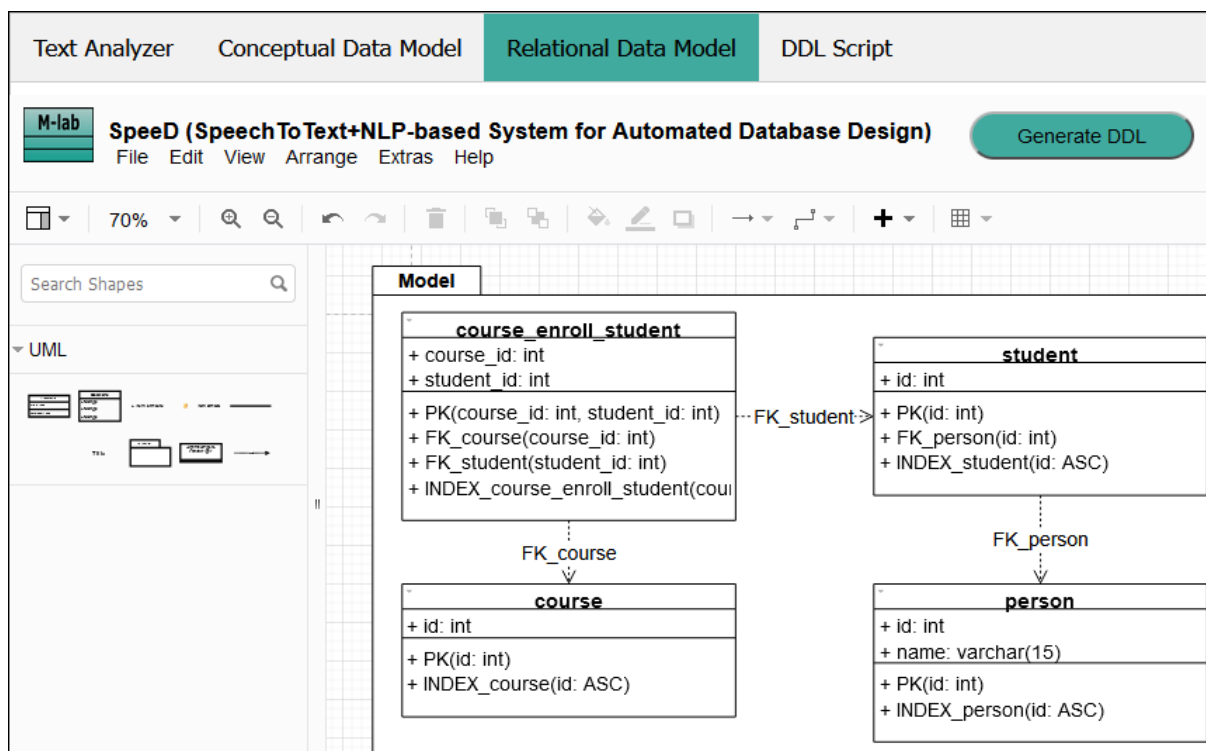
Naredni korak je generisanje relacionog modela. Modalni prozor za odabir DBMS-a i ostalih parametara je prikazan na sl. 5.29.



Slika 5.28: Konceptualni model generisan na osnovu prepoznatog teksta

Slika 5.29: Modalni prozor za odabir DBMS-a i konfiguraciju

Nakon sinteze konceptualnog modela, naredna faza projektovanja je generisanje relacionog modela. Na sl. 5.30 prikazan je dobijeni relacioni model. Alat je uspješno mapirao elemente konceptualnog modela u odgovarajuće elemente relacionog modela.



Slika 5.30: Relacioni model generisan na osnovu konceptualnog modela

Nakon generisanja relacionog modela, moguće je generisati skriptu baze podataka za odabrani DBMS, što je prikazano na sl. 5.31. Alat je mapirao klase, atribute i veze u odgovarajuće upite za odabrani DBMS.

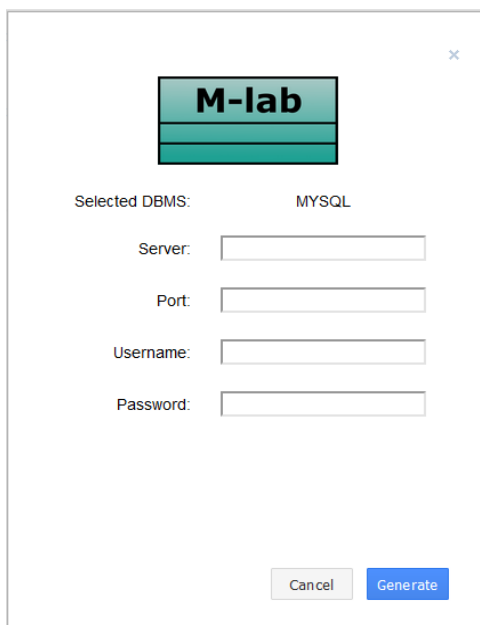
```

1 CREATE SCHEMA Model;
2 CREATE TABLE Model.course ( id int NOT NULL, PRIMARY KEY (id) );
3 CREATE TABLE Model.course_enroll_student ( course_id int NOT NULL, student_id int NOT NULL, PRIMARY
4 KEY (course_id, student_id) );
5 CREATE TABLE Model.person ( id int NOT NULL, name varchar(15) NOT NULL, PRIMARY KEY (id) );
6 CREATE TABLE Model.student ( id int NOT NULL, PRIMARY KEY (id) );
7 ALTER TABLE Model.course_enroll_student ADD CONSTRAINT course_enroll_student_FK_course FOREIGN KEY
8 (course_id) REFERENCES Model.course (id) ON DELETE RESTRICT ON UPDATE CASCADE;
9 ALTER TABLE Model.course_enroll_student ADD CONSTRAINT course_enroll_student_FK_student FOREIGN KEY
10 (student_id) REFERENCES Model.student (id) ON DELETE RESTRICT ON UPDATE CASCADE;
11 ALTER TABLE Model.student ADD CONSTRAINT student_FK_person FOREIGN KEY (id) REFERENCES Model.person
12 (id) ON DELETE RESTRICT ON UPDATE CASCADE;
13 CREATE INDEX INDEX_course ON Model.course (id ASC);
14 CREATE INDEX INDEX_course_enroll_student ON Model.course_enroll_student (course_id ASC, student_id
15 ASC);
16 CREATE INDEX INDEX_person ON Model.person (id ASC);
17 CREATE INDEX INDEX_student ON Model.student (id ASC);

```

Slika 5.31: DDL skripta generisana na osnovu relacionog modela

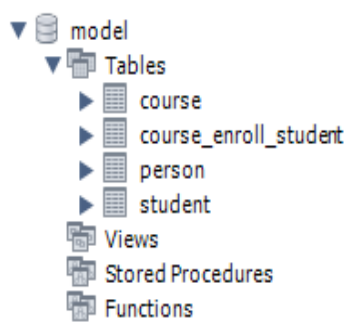
Naredni korak je generisanje šeme baze podataka, što podrazumijeva unos parametara konekcije nakon čega se izvršava dobijena skripta. Na sl. 5.32 prikazan je prozor za unos parametara konekcije za ranije odabrani DMBS.



The image shows a modal dialog box titled "M-lab" with a close button in the top right corner. Below the title bar, it displays "Selected DBMS: MYSQL". There are four input fields labeled "Server:", "Port:", "Username:", and "Password:". At the bottom of the dialog, there are two buttons: "Cancel" and "Generate".

Slika 5.32: Modalni prozor za unos konekcionih parametara za odabrani DBMS

Nakon unosa parametara i odabira opcije za generisanje fizičke šeme baze podataka, izvršiće se dobijena skripta za odabrani DBMS. Na sl. 5.33 prikazana je fizička šema baze podataka.



Slika 5.33: Fizička šema baze podataka

Glava 6

Eksperimentalni rezultati

U ovoj glavi su opisani rezultati eksperimenta automatizovanog generisanja konceptualnog modela baze podataka na osnovu govora. Eksperimentalni rezultati predstavljaju dobijenu preciznost, odziv i efektivnost prepoznavanja entiteta, atributa i asocijacija i generalizacija u dobijenom modelu.

Ako TP predstavlja broj tačno prepoznatih koncepta generisanog konceptualnog modela, a FP broj koncepta koji su pogrešno prepoznati, onda se preciznost može definisati kao:

$$P = \frac{TP}{TP + FP} \quad (6.1)$$

Računanjem preciznosti se dolazi do informacija o tome koliki je udio prepoznatih konceptata koji su ispravno identifikovani.

Ako FN predstavlja broj koncepta koji nedostaju, odziv se može definisati kao:

$$R = \frac{TP}{TP + FN} \quad (6.2)$$

Računanjem odziva se dolazi do informacija o tome koliko je kompletan automatski generisani model u odnosu na ciljni.

Efektivnost se računa kao harmonijska sredina preciznosti i odziva i može se definisati kao:

$$F = 2 \times \frac{P \times R}{P + R} \quad (6.3)$$

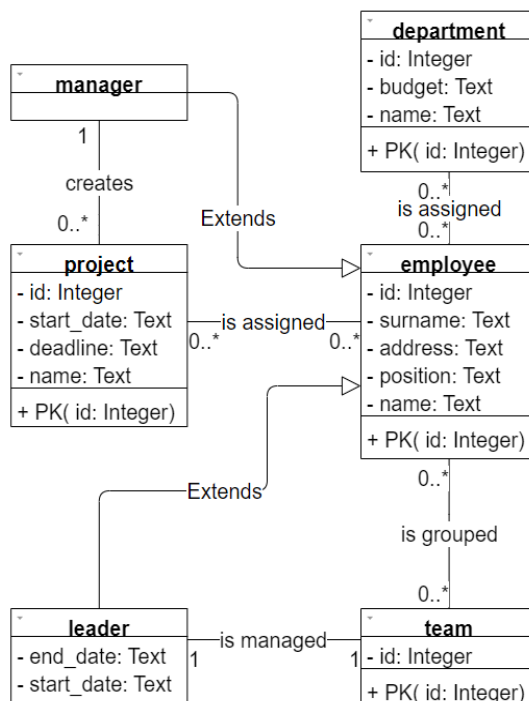
U eksperimentu izvršena je analiza preciznosti, odziva i efektivnosti prilikom korišćenja različitih Vosk modela i opcija za analizu govora. Prikazana su dva primjera, odnosno jedan vezan za organizacionu strukturu a drugi za mrežnu infrastrukturu. Prikazani su i analizirani rezultati za oba primjera na engleskom i njemačkom jeziku. Audio datoteke na njemačkom jeziku koriste sintetički generisan glas umjesto stvarnog govora. Prikazani su očekivani konceptualni modeli za oba primjera, odnosno četiri konceptualna modela, po dva za oba jezika (engleski i njemački). Prikazan je rezultat jedne audio datoteke sa kombinacijama malih i velikih modela, bez ili sa korišćenjem opcija za analizu govora.

6.1. Očekivani rezultati konceptualnog modelovanja na osnovu govora

U nastavku su prikazani očekivani rezultati četiri primjera konceptualnog modelovanja na osnovu govora.

6.1.1. Očekivani rezultati konceptualnog modelovanja na primjeru organizacione strukture zasnovanog na govoru na engleskom jeziku

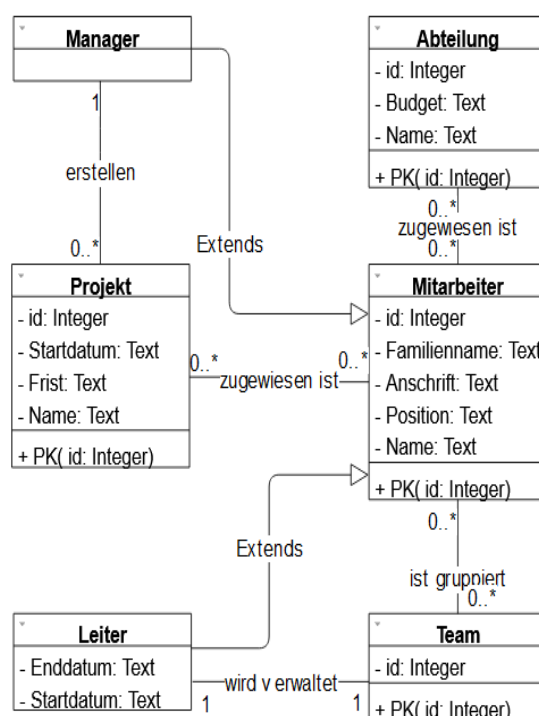
Na osnovu specifikacije zahtjeva organizacione strukture "Each employee has a name, surname, address, and position. Employees are assigned to different departments. Each department has a name and budget. A manager is an employee within an organization. Employees are assigned to multiple projects. Each project has a name, start date, and deadline. A manager creates these projects. Employees are grouped into teams. A leader is an employee. A leader has a start date and an end date. Each team is managed by a leader." manuelno je projektovan referentni konceptualni model, koji je prikazan na sl. 6.1.



Slika 6.1: Referentni konceptualni model organizacione strukture na engleskom jeziku

6.1.2. Očekivani rezultati konceptualnog modelovanja na primjeru organizacione strukture zasnovanog na govoru na njemačkom jeziku

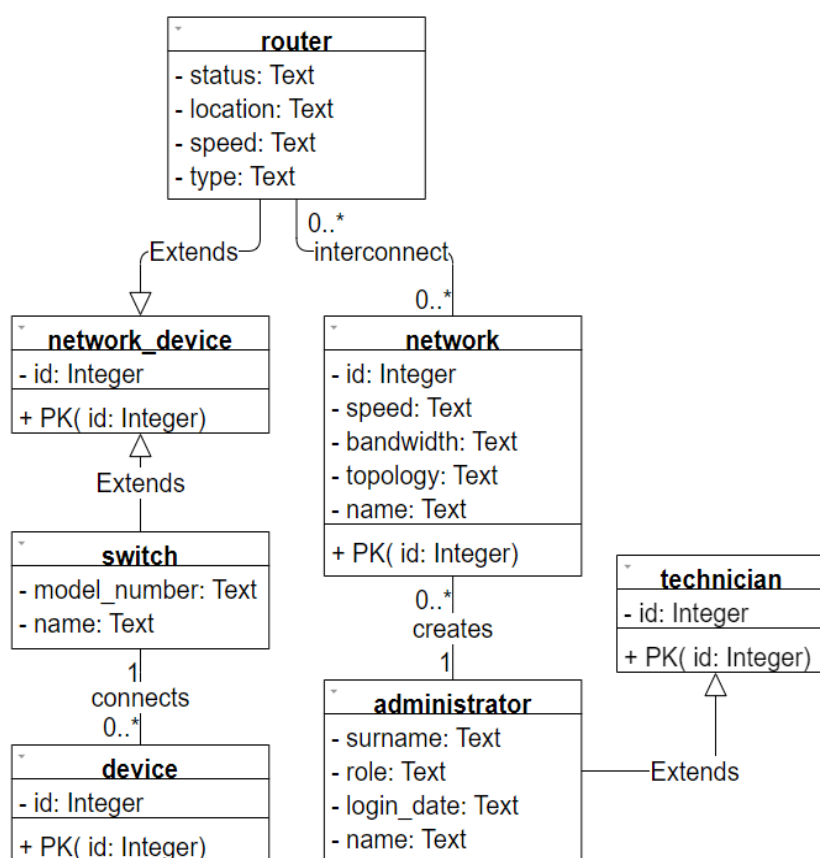
Na osnovu specifikacije zahtjeva za prikaz organizacione strukture "Jeder Mitarbeiter hat einen Namen, Nachnamen, Adresse und Position. Mitarbeiter werden verschiedenen Abteilungen zugewiesen. Jede Abteilung hat einen Namen und ein Budget. Ein Manager ist ein Mitarbeiter innerhalb einer Organisation. Mitarbeiter werden mehreren Projekten zugewiesen. Jedes Projekt hat einen Namen, ein Startdatum und eine Frist. Ein Manager erstellt diese Projekte. Mitarbeiter werden in Teams gruppiert. Ein Leiter ist ein Mitarbeiter. Ein Leiter hat ein Startdatum und ein Enddatum. Jedes Team wird von einem Leiter verwaltet." manuelno je projektovan referentni konceptualni model, koji je prikazan na sl. 6.2. Tekst na njemačkom jeziku predstavlja prevedeni tekst iz prošlog primjera. Dobijeni konceptualni model je identičan kao u prethodnom primjeru, osim što su nazivi entiteta, njihovih veza i atributa na njemačkom jeziku.



Slika 6.2: Referentni konceptualni model organizacione strukture na njemačkom jeziku

6.1.3. Očekivani rezultati konceptualnog modelovanja na primjeru mrežne infrastrukture zasnovanog na govoru na engleskom jeziku

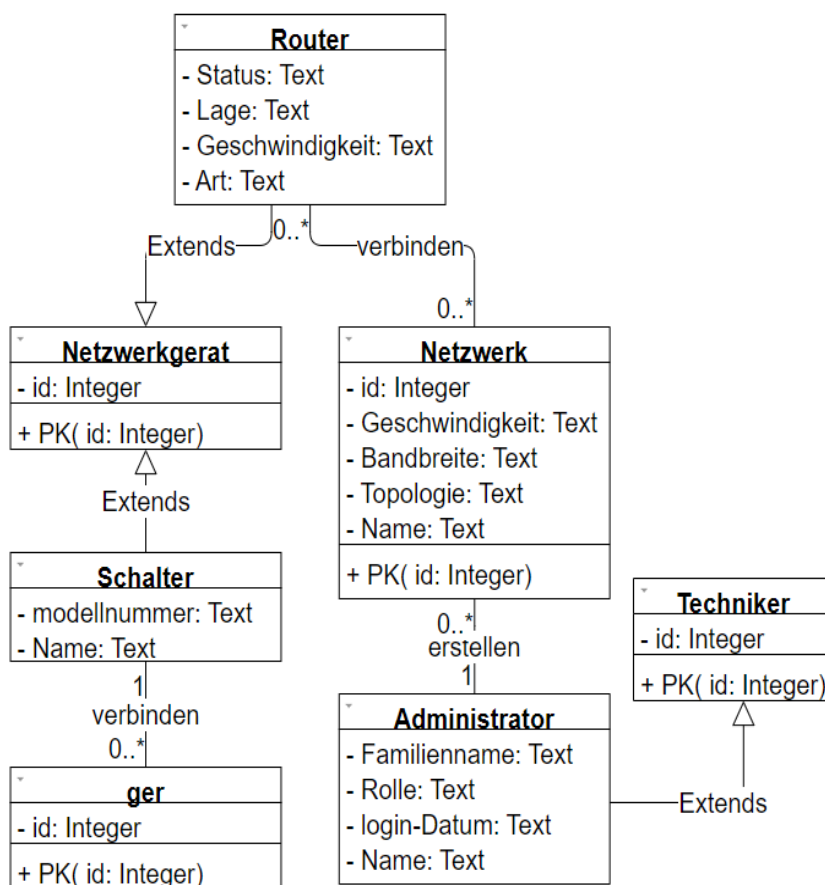
Na osnovu specifikacije zahtjeva za prikaz mrežne infrastrukture "Each router has a type, status, location, and speed. Routers interconnect different networks. Each network has a name, speed, bandwidth, and topology. A switch connects devices. A switch has a name and a model number. An administrator is a technician. An administrator has a name, surname, role, and login date. An administrator creates networks. A router is a network device. A switch is also a network device." manuelno je projektovan referentni konceptualni model, koji je prikazan na sl. 6.3.



Slika 6.3: Referentni konceptualni model mrežne infrastrukture na engleskom jeziku

6.1.4. Očekivani rezultati konceptualnog modelovanja na primjeru mrežne infrastrukture zasnovanog na govoru na njemačkom jeziku

Na osnovu specifikacije zahtjeva za prikaz mrežne infrastrukture "Jeder Router hat einen Typ, einen Status, einen Standort und eine Geschwindigkeit. Router verbinden verschiedene Netzwerke. Jedes Netzwerk hat Name, Geschwindigkeit, Bandbreite und Topologie. Ein Switch verbindet Geräte. Ein Switch hat einen Namen und eine Modellnummer. Ein Administrator ist ein Techniker. Ein Administrator hat Name, Nachname, Rolle und Anmeldedatum. Ein Administrator erstellt Netzwerke. Ein Router ist ein Netzwerkgerät. Ein Switch ist ebenfalls ein Netzwerkgerät." manuelno je projektovan referentni konceptualni model, koji je prikazan na sl. 6.4. Dobijeni konceptualni model je identičan kao u prethodnom primjeru, osim što su nazivi entiteta, njihovih veza i atributa na njemačkom jeziku.

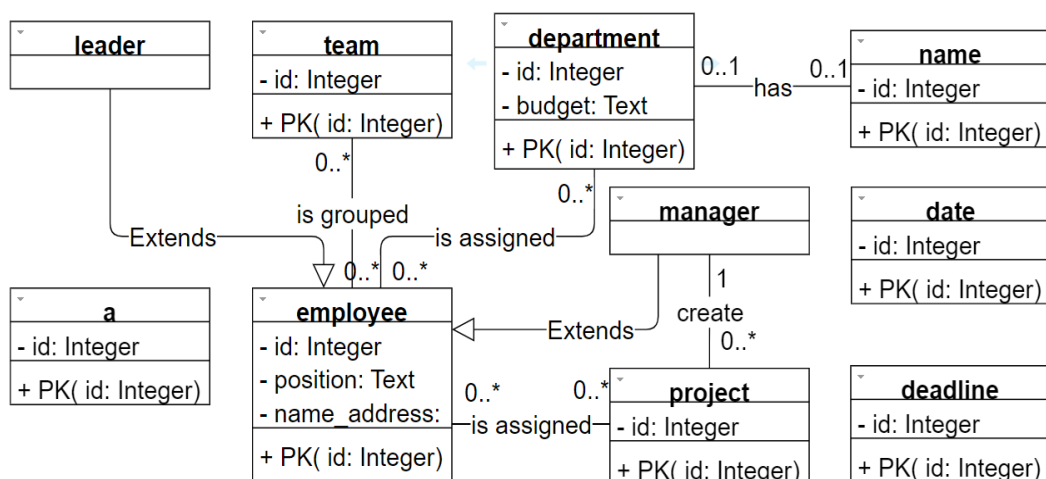


Slika 6.4: Referentni konceptualni model mrežne infrastrukture na njemačkom jeziku

6.2. Eksperimentalni rezultati konceptualnog modelovanja na osnovu govora

U nastavku su prikazani primjeri izvođenja eksperimenata za jednu ulaznu datoteku sa svim kombinacijama opcija. Rezultati primjera konceptualnog modelovanja na osnovu govora su:

1. **Upotreba audio datoteke na engleskom jeziku za prikaz organizacione strukture pomoću malog Vosk modela.** Dobijeni tekst audio datoteke je *"Each employee has a name sir name address and position. Employees are assigned to different departments. Each department has a name and budget. A manager is an employee within an organization. Employees are assigned to multiple projects. Each project his a name started to date and the deadline. A manager create these projects. Employees are grouped into teams. A leader is an employee. A leader a his a started to date and and and date. Each team is managed by be either."*. Na sl. 6.5 prikazan je automatski generisan konceptualni model na osnovu prepoznatog teksta, dok je u tabeli 6.1 prikazana evaluacija dobijenog konceptualnog modela.

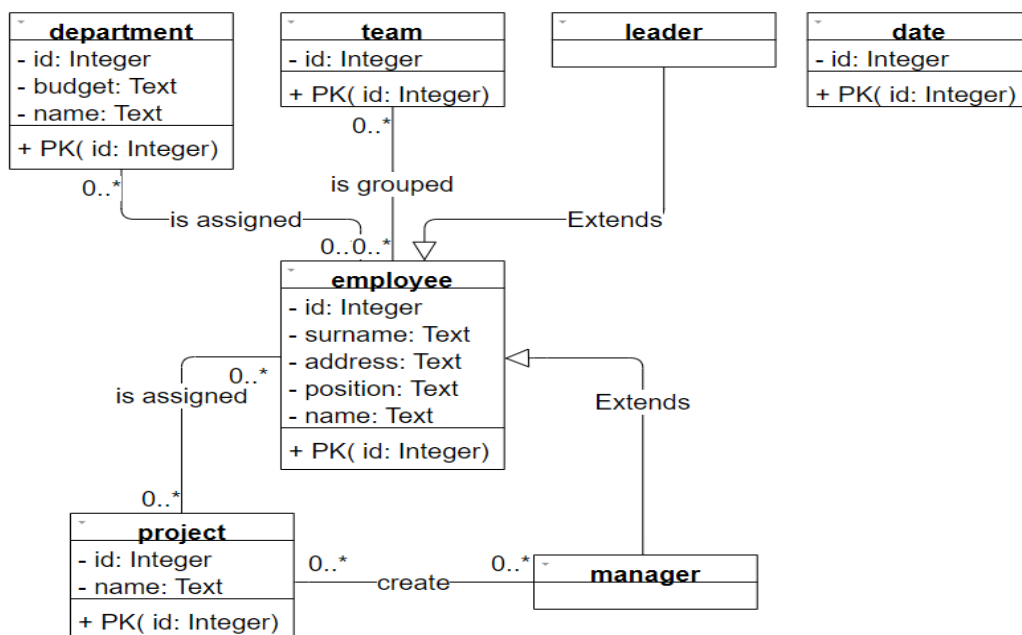


Slika 6.5: Konceptualni model dobijen na osnovu teksta generisanog upotrebom malog Vosk modela

Tabela 6.1: Evaluacija automatski generisanog konceptualnog modela

Test	TP	FP	FN	R	P	F
Entiteti	6	4	0	1	0.6	0.75
Atributi	6	5	9	0.4	0.55	0.46
Asocijacije	4	1	1	0.8	0.8	0.8
Generalizacije	2	0	0	1	1	1

2. **Upotreba audio datoteke na engleskom jeziku za prikaz organizacione strukture pomoću malog Vosk modela i alata za provjeru gramatike.** Dobijeni tekst audio datoteke je "Each employee has a name, surname, address and position. Employees are assigned to different departments. Each department has a name and budget. A manager is an employee within an organization. Employees are assigned to multiple projects. Each project has a name, start to date and deadline. Managers create these projects. Employees are grouped into teams. A leader is an employee. A leader has started to date and date. Each team is managed by either.". Na sl. 6.6 prikazan je automatski generisan konceptualni model na osnovu dobijenog teksta, dok je u tabeli 6.2 prikazana evaluacija dobijenog konceptualnog modela.

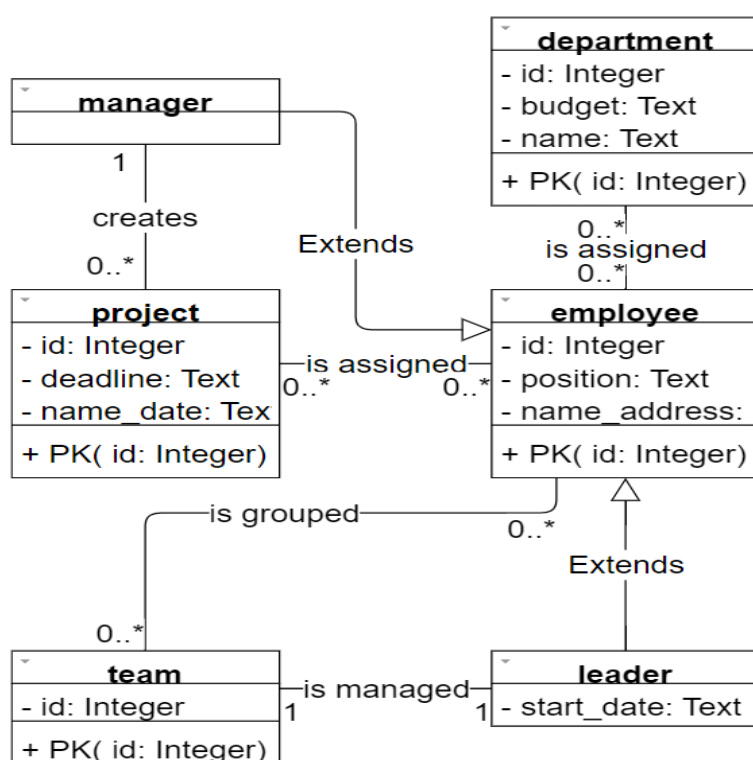


Slika 6.6: Konceptualni model dobijen na osnovu analiziranog teksta generisanog upotrebom malog Vosk modela

Tabela 6.2: Evaluacija automatski generisanog konceptualnog modela

Test	TP	FP	FN	R	P	F
Entiteti	6	1	0	1	0.86	0.92
Atributi	11	1	4	0.73	0.92	0.81
Asocijacije	3	1	1	0.75	0.75	0.75
Generalizacije	2	0	0	1	1	1

3. **Upotreba audio datoteke na engleskom jeziku za prikaz organizacione strukture pomoću velikog Vosk modela.** Dobijeni tekst audio datoteke je ”Each employee has a name surname address and position. Employees are assigned to different departments. Each department has a name and budget. A manager is an employee within an organization. Employees are assigned to multiple projects. Each project has a name start date and a deadline. A manager creates these projects. Employees are grouped into teams. A leader is an employee. A leader has a start date and and ends date. Each team is managed by a leader.”. Na sl. 6.7 prikazan je automatski generisan konceptualni model na osnovu dobijenog teksta, dok je u tabeli 6.3 prikazana evaluacija dobijenog konceptualnog modela.

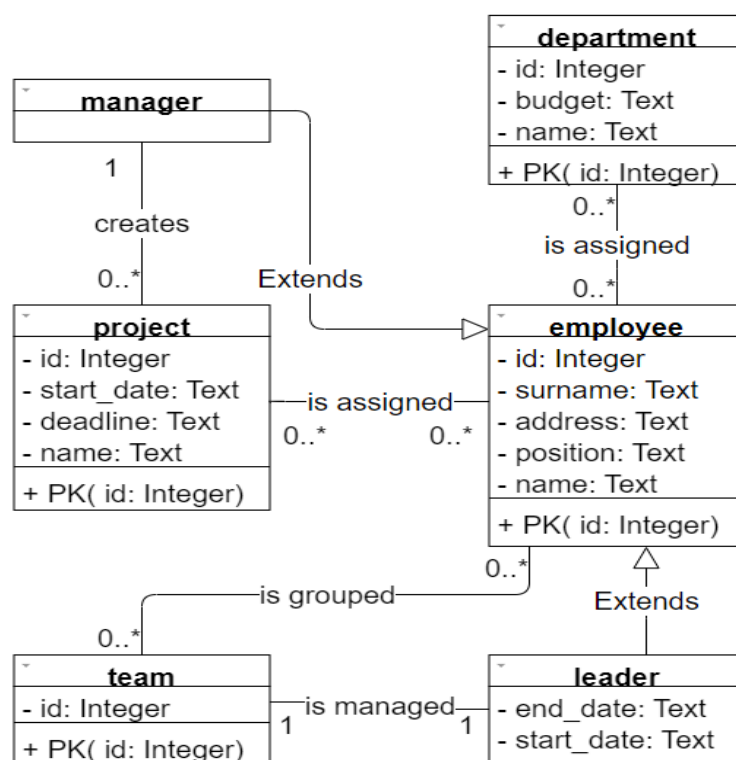


Slika 6.7: Konceptualni model dobijen na osnovu teksta generisanog upotrebom malog Vosk modela

Tabela 6.3: Evaluacija automatski generisanog konceptualnog modela

Test	TP	FP	FN	R	P	F
Entiteti	6	0	0	1	1	1
Atributi	9	2	6	0.6	0.82	0.69
Asocijacije	5	0	0	1	1	1
Generalizacije	2	0	0	1	1	1

4. **Upotreba audio datoteke na engleskom jeziku za prikaz organizacione strukture pomoću velikog Vosk modela i alata za provjeru gramatike.** Dobijeni tekst audio datoteke je "Each employee has a name, surname, address and position. Employees are assigned to different departments. Each department has a name and budget. A manager is an employee within an organization. Employees are assigned to multiple projects. Each project has a name, start date and a deadline. A manager creates these projects. Employees are grouped into teams. A leader is an employee. A leader has a start date and an end date. Each team is managed by a leader." . Na sl. 6.8 prikazan je automatski generisan konceptualni model na osnovu dobijenog teksta, dok je u tabeli 6.4 prikazana evaluacija dobijenog konceptualnog modela.



Slika 6.8: Konceptualni model dobijen na osnovu analiziranog teksta generisanog upotrebom malog Vosk modela

Tabela 6.4: Evaluacija automatski generisanog konceptualnog modela

Test	TP	FP	FN	R	P	F
Entiteti	6	0	0	1	1	1
Atributi	15	0	0	1	1	1
Asocijacije	5	0	0	1	1	1
Generalizacije	2	0	0	1	1	1

6.2.1. Eksperimentalni rezultati konceptualnog modelovanja zasnovanog na govoru na engleskom jeziku na primjeru organizacione strukture

Rezultati analize prepoznavanja entiteta, atributa, asocijacija i generalizacija za primjere organizacione strukture na engleskom jeziku prikazani su u tabelama 6.5, 6.6, 6.7 i 6.8 redom.

Tabela 6.5: Evaluacija entiteta u automatski generisanim konceptualnim modelima

Primjer	Ulaz	Model	Analiza teksta	TP	FP	FN	R	P	F
P1	Audio datoteka	Mali	Da	6	1	0	1	0.86	0.92
			Ne	6	4	0	1	0.6	0.75
		Veliki	Da	6	0	0	1	1	1
			Ne	6	0	0	1	1	1
P2	Audio datoteka	Mali	Da	4	2	1	0.8	0.67	0.73
			Ne	4	2	1	0.8	0.67	0.73
		Veliki	Da	6	0	0	1	1	1
			Ne	5	1	0	1	0.83	0.91
P3	Mikrofon	Mali	Da	6	3	0	1	0.67	0.8
			Ne	4	2	2	0.67	0.67	0.67
		Veliki	Da	6	5	0	1	0.55	0.71
			Ne	2	1	4	0.33	0.67	0.44
P4	Mikrofon	Mali	Da	6	0	0	1	1	1
			Ne	6	1	0	1	0.86	0.92
		Veliki	Da	6	0	0	1	1	1
			Ne	5	0	1	0.83	1	0.91

Tabela 6.6: Evaluacija atributa u automatski generisanim konceptualnim modelima

Primjer	Ulaz	Model	Analiza teksta	TP	FP	FN	R	P	F
P1	Audio datoteka	Mali	Da	11	1	4	0.73	0.92	0.81
			Ne	6	5	9	0.4	0.55	0.46
		Veliki	Da	15	0	0	1	1	1
			Ne	9	2	6	0.6	0.82	0.69
P2	Audio datoteka	Mali	Da	12	3	3	0.8	0.8	0.8
			Ne	7	7	8	0.47	0.5	0.48
		Veliki	Da	14	1	1	0.93	0.93	0.93
			Ne	9	2	6	0.6	0.82	0.69
P3	Mikrofon	Mali	Da	8	1	7	0.53	0.89	0.66
			Ne	2	8	13	0.13	0.2	0.16
		Veliki	Da	9	5	6	0.6	0.64	0.62
			Ne	1	3	14	0.07	0.25	0.11
P4	Mikrofon	Mali	Da	10	0	5	0.67	1	0.8
			Ne	6	2	9	0.4	0.75	0.52
		Veliki	Da	15	0	0	1	1	1
			Ne	8	1	7	0.53	0.89	0.66

Tabela 6.7: Evaluacija asocijacija u automatski generisanim konceptualnim modelima

Primjer	Ulaz	Model	Analiza teksta	TP	FP	FN	R	P	F
P1	Audio datoteka	Mali	Da	3	1	1	0.75	0.75	0.75
			Ne	4	1	1	0.8	0.8	0.8
		Veliki	Da	5	0	0	1	1	1
			Ne	5	0	0	1	1	1
P2	Audio datoteka	Mali	Da	1	1	4	0.2	0.5	0.29
			Ne	1	1	4	0.2	0.5	0.29
		Veliki	Da	3	0	2	0.6	1	0.75
			Ne	1	0	4	0.2	1	0.33
P3	Mikrofon	Mali	Da	5	0	0	1	1	1
			Ne	1	0	4	0.2	1	0.33
		Veliki	Da	4	2	1	0.8	0.67	0.73
			Ne	0	0	5	0	0	0
P4	Mikrofon	Mali	Da	5	0	0	1	1	1
			Ne	5	0	0	1	1	1
		Veliki	Da	5	0	0	1	1	1
			Ne	2	0	3	0.4	1	0.57

Tabela 6.8: Evaluacija generalizacija u automatski generisanim konceptualnim modelima

Primjer	Ulaz	Model	Analiza teksta	TP	FP	FN	R	P	F
P1	Audio datoteka	Mali	Da	2	0	0	1	1	1
			Ne	2	0	0	1	1	1
		Veliki	Da	2	0	0	1	1	1
			Ne	2	0	0	1	1	1
P2	Audio datoteka	Mali	Da	1	0	1	0.5	1	0.67
			Ne	1	0	1	0.5	1	0.67
		Veliki	Da	1	0	1	0.5	1	0.67
			Ne	2	0	0	1	1	1
P3	Mikrofon	Mali	Da	2	0	0	1	1	1
			Ne	0	1	2	0	0	0
		Veliki	Da	2	0	0	1	1	1
			Ne	0	0	5	0	0	0
P4	Mikrofon	Mali	Da	2	0	0	1	1	1
			Ne	1	2	1	0.5	0.33	0.4
		Veliki	Da	2	0	0	1	1	1
			Ne	2	0	0	1	1	1

6.2.2. Eksperimentalni rezultati konceptualnog modelovanja zasnovanog na govoru na njemačkom jeziku na primjeru organizacione strukture

Rezultati analize prepoznavanja entiteta, atributa, asocijacija i generalizacija za primjere organizacione strukture na njemačkom jeziku prikazani su u tabeli 6.9.

Tabela 6.9: Evaluacija automatski generisanih konceptualnih modela

Primjer	Model	Analiza teksta	TP	FP	FN	R	P	F
P1 - entiteti	Mali	Da	6	1	0	1	0.86	0.92
		Ne	6	1	0	1	0.86	0.92
	Veliki	Da	5	0	1	0.83	1	0.91
		Ne	5	0	1	0.83	1	0.91
P2 - entiteti	Mali	Da	6	0	0	1	1	1
		Ne	6	0	0	1	1	1
	Veliki	Da	6	0	0	1	1	1
		Ne	6	0	0	1	1	1
P1 - atributi	Mali	Da	7	7	8	0.47	0.5	0.48
		Ne	7	7	8	0.47	0.5	0.48
	Veliki	Da	5	2	10	0.33	0.71	0.45
		Ne	5	2	10	0.33	0.71	0.45
P2 - atributi	Mali	Da	10	3	5	0.67	0.77	0.72
		Ne	10	3	5	0.67	0.77	0.72
	Veliki	Da	7	2	8	0.47	0.78	0.59
		Ne	7	2	8	0.47	0.78	0.59
P1 - asocijacije	Mali	Da	4	0	1	0.8	1	0.89
		Ne	4	0	1	0.8	1	0.89
	Veliki	Da	3	0	2	0.6	1	0.75
		Ne	3	0	2	0.6	1	0.75
P2 - asocijacije	Mali	Da	5	0	0	1	1	1
		Ne	5	0	0	1	1	1
	Veliki	Da	4	0	1	0.8	1	0.89
		Ne	4	0	1	0.8	1	0.89
P1 - generalizacije	Mali	Da	1	0	1	0.5	1	0.67
		Ne	1	0	1	0.5	1	0.67
	Veliki	Da	2	0	0	1	1	1
		Ne	2	0	0	1	1	1
P2 - generalizacije	Mali	Da	2	0	0	1	1	1
		Ne	2	0	0	1	1	1
	Veliki	Da	2	0	0	1	1	1
		Ne	2	0	0	1	1	1

6.2.3. Eksperimentalni rezultati konceptualnog modelovanja zasnovanog na govoru na engleskom jeziku na primjeru mrežne infrastrukture

Rezultati analize prepoznavanja entiteta, atributa, asocijacija i generalizacija za primjere mrežne infrastrukture na engleskom jeziku prikazani su u tabelama 6.10, 6.11, 6.12 i 6.13, redom.

Tabela 6.10: Evaluacija entiteta u automatski generisanim konceptualnim modelima

Primjer	Ulaz	Model	Analiza teksta	TP	FP	FN	R	P	F
P1	Audio datoteka	Mali	Da	7	4	0	1	0.64	0.78
			Ne	6	9	0	1	0.4	0.57
		Veliki	Da	6	0	1	0.86	1	0.92
			Ne	6	0	1	0.86	1	0.92
P2	Audio datoteka	Mali	Da	6	2	1	0.86	0.75	0.8
			Ne	5	2	2	0.71	0.71	0.71
		Veliki	Da	5	3	2	0.71	0.63	0.67
			Ne	2	3	5	0.29	0.4	0.34
P3	Mikrofon	Mali	Da	7	7	0	1	0.5	0.67
			Ne	4	7	3	0.57	0.36	0.44
		Veliki	Da	7	2	0	1	0.78	0.88
			Ne	3	1	4	0.43	0.75	0.55
P4	Mikrofon	Mali	Da	6	0	1	0.86	1	0.92
			Ne	7	5	0	1	0.58	0.73
		Veliki	Da	6	0	1	0.86	1	0.92
			Ne	7	0	0	1	1	1

Tabela 6.11: Evaluacija atributa u automatski generisanim konceptualnim modelima

Primjer	Ulaz	Model	Analiza teksta	TP	FP	FN	R	P	F
P1	Audio datoteka	Mali	Da	9	7	9	0.5	0.56	0.53
			Ne	6	15	8	0.43	0.29	0.35
		Veliki	Da	11	5	6	0.65	0.69	0.67
			Ne	8	7	9	0.47	0.53	0.5
P2	Audio datoteka	Mali	Da	3	7	15	0.17	0.3	0.22
			Ne	3	6	15	0.17	0.33	0.22
		Veliki	Da	7	8	11	0.39	0.47	0.43
			Ne	1	5	17	0.06	0.17	0.09
P3	Mikrofon	Mali	Da	4	8	14	0.22	0.33	0.26
			Ne	2	6	16	0.11	0.25	0.15
		Veliki	Da	5	5	13	0.28	0.5	0.36
			Ne	3	3	15	0.17	0.5	0.25
P4	Mikrofon	Mali	Da	10	3	8	0.56	0.77	0.65
			Ne	7	8	11	0.39	0.47	0.43
		Veliki	Da	10	2	8	0.56	0.83	0.67
			Ne	7	4	11	0.39	0.64	0.48

Tabela 6.12: Evaluacija asocijacija u automatski generisanim konceptualnim modelima

Primjer	Ulaz	Model	Analiza teksta	TP	FP	FN	R	P	F
P1	Audio datoteka	Mali	Da	3	2	0	1	0.6	0.75
			Ne	2	0	1	0.67	1	0.8
		Veliki	Da	3	1	0	1	0.75	0.86
			Ne	3	1	0	1	0.75	0.86
P2	Audio datoteka	Mali	Da	1	1	2	0.33	0.5	0.4
			Ne	0	1	3	0	0	0
		Veliki	Da	0	2	3	0	0	0
			Ne	0	0	3	0	0	0
P3	Mikrofon	Mali	Da	2	0	1	0.67	1	0.8
			Ne	1	0	2	0.33	1	0.5
		Veliki	Da	2	1	1	0.67	0.67	0.67
			Ne	0	2	3	0	0	0
P4	Mikrofon	Mali	Da	2	0	1	0.67	1	0.8
			Ne	3	0	0	1	1	1
		Veliki	Da	2	0	1	0.67	1	0.8
			Ne	3	0	0	1	1	1

Tabela 6.13: Evaluacija generalizacija u automatski generisanim konceptualnim modelima

Primjer	Ulaz	Model	Analiza teksta	TP	FP	FN	R	P	F
P1	Audio datoteka	Mali	Da	2	0	1	0.67	1	0.8
			Ne	1	1	2	0.33	0.5	0.4
		Veliki	Da	2	0	1	0.67	1	0.8
			Ne	1	0	2	0.33	1	0.5
P2	Audio datoteka	Mali	Da	0	0	3	0	0	0
			Ne	0	0	3	0	0	0
		Veliki	Da	0	2	3	0	0	0
			Ne	0	2	3	0	0	0
P3	Mikrofon	Mali	Da	2	2	1	0.67	0.5	0.57
			Ne	1	2	2	0.33	0.33	0.33
		Veliki	Da	3	0	0	1	1	1
			Ne	0	0	3	0	0	0
P4	Mikrofon	Mali	Da	3	0	0	1	1	1
			Ne	3	0	0	1	1	1
		Veliki	Da	3	0	0	1	1	1
			Ne	3	0	0	1	1	1

6.2.4. Eksperimentalni rezultati konceptualnog modelovanja zasnovanog na govoru na njemačkom jeziku na primjeru mrežne infrastrukture

Rezultati analize prepoznavanja entiteta, atributa, asocijacija i generalizacija za primjere mrežne infrastrukture na njemačkom jeziku prikazani su u tabeli 6.14.

Tabela 6.14: Evaluacija automatski generisanih konceptualnih modela

Primjer	Model	Analiza teksta	TP	FP	FN	R	P	F
P1 - entiteti	Mali	Da	7	4	0	1	0.64	0.78
		Ne	7	4	0	1	0.64	0.78
	Veliki	Da	7	1	0	1	0.88	0.94
		Ne	7	1	0	1	0.88	0.94
P2 - entiteti	Mali	Da	6	0	1	0.86	1	0.92
		Ne	6	0	1	0.86	1	0.92
	Veliki	Da	7	0	0	1	1	1
		Ne	7	0	0	1	1	1
P1 - atributi	Mali	Da	8	5	10	0.44	0.62	0.51
		Ne	8	5	10	0.44	0.62	0.51
	Veliki	Da	8	6	10	0.44	0.57	0.5
		Ne	8	6	10	0.44	0.57	0.5
P2 - atributi	Mali	Da	7	3	11	0.39	0.7	0.5
		Ne	7	3	11	0.39	0.7	0.5
	Veliki	Da	10	2	8	0.56	0.83	0.67
		Ne	10	2	8	0.56	0.83	0.67
P1 - asocijacije	Mali	Da	2	2	1	0.67	0.5	0.57
		Ne	2	2	1	0.67	0.5	0.57
	Veliki	Da	3	0	0	1	1	1
		Ne	3	0	0	1	1	1
P2 - asocijacije	Mali	Da	2	0	1	0.67	1	0.8
		Ne	2	0	1	0.67	1	0.8
	Veliki	Da	3	0	0	1	1	1
		Ne	3	0	0	1	1	1
P1 - generalizacije	Mali	Da	2	1	1	0.67	0.67	0.67
		Ne	2	1	1	0.67	0.67	0.67
	Veliki	Da	2	0	1	0.67	1	0.8
		Ne	2	0	1	0.67	1	0.8
P2 - generalizacije	Mali	Da	3	0	0	1	1	1
		Ne	3	0	0	1	1	1
	Veliki	Da	3	0	0	1	1	1
		Ne	3	0	0	1	1	1

6.3. Sistematizacija rezultata

U tabelama 6.15, 6.16, 6.17, 6.18 prikazani su agregirani eksperimentalni rezultati za engleski jezik.

Tabela 6.15: Engleski jezik / audio datoteka / mali model / bez korekcija gramatike

Kategorija	Organizaciona struktura			Mrežna infrastruktura			Prosjek		
	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]
Klase	87	70	77	82	51	61	85	61	69
Atributi	35	50	41	28	34	29	32	42	35
Asocijacije	55	83	61	50	75	58	53	79	60
Generalizacije	50	58	52	42	46	43	46	52	48
Ukupno za studiju	50	61	55	43	45	44	47	53	50

Tabela 6.16: Engleski jezik / audio datoteka / mali model / sa korekcijom gramatike

Kategorija	Organizaciona struktura			Mrežna infrastruktura			Prosjek		
	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]
Klase	95	80	86	93	72	79	94	76	83
Atributi	68	90	77	36	49	42	52	70	60
Asocijacije	74	81	76	67	78	69	71	80	73
Generalizacije	88	100	92	59	63	59	74	82	76
Ukupno za studiju	76	87	81	54	61	57	65	74	69

Tabela 6.17: Engleski jezik / audio datoteka / veliki model / bez korekcija gramatike

Kategorija	Organizaciona struktura			Mrežna infrastruktura			Prosjek		
	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]
Klase	79	88	82	65	79	70	72	84	76
Atributi	45	70	54	27	46	33	36	58	44
Asocijacije	40	75	48	50	44	47	45	60	48
Generalizacije	75	75	75	33	50	38	54	63	57
Ukupno za studiju	52	86	65	38	63	47	45	75	56

Tabela 6.18: Engleski jezik / audio datoteka / veliki model / sa korekcijom gramatike

Kategorija	Organizaciona struktura			Mrežna infrastruktura			Prosjek		
	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]
Klase	100	89	93	86	85	85	93	87	89
Atributi	88	89	89	47	62	53	68	76	71
Asocijacije	85	92	87	59	61	58	72	77	73
Generalizacije	88	100	92	67	75	70	78	88	81
Ukupno za studiju	90	89	89	59	70	64	75	80	77

U tabelama 6.19, 6.20, 6.21, 6.22 prikazani su agregirani eksperimentalni rezultati za njemački jezik.

Tabela 6.19: Njemački jezik / audio datoteka / mali model / bez korekcija gramatike

Kategorija	Organizaciona struktura			Mrežna infrastruktura			Prosjek		
	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]
Klase	100	93	96	93	82	85	97	88	91
Atributi	57	64	60	42	66	51	50	65	56
Asocijacije	90	100	95	67	75	69	79	88	82
Generalizacije	75	100	84	84	84	84	80	92	84
Ukupno za studiju	73	79	76	60	71	65	67	75	71

Tabela 6.20: Njemački jezik / audio datoteka / mali model / sa korekcijom gramatike

Kategorija	Organizaciona struktura			Mrežna infrastruktura			Prosjek		
	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]
Klase	100	93	96	93	82	85	97	88	91
Atributi	57	64	60	42	66	51	50	65	56
Asocijacije	90	100	95	67	75	69	79	88	82
Generalizacije	75	100	84	84	84	84	80	92	84
Ukupno za studiju	73	79	76	60	71	65	67	75	71

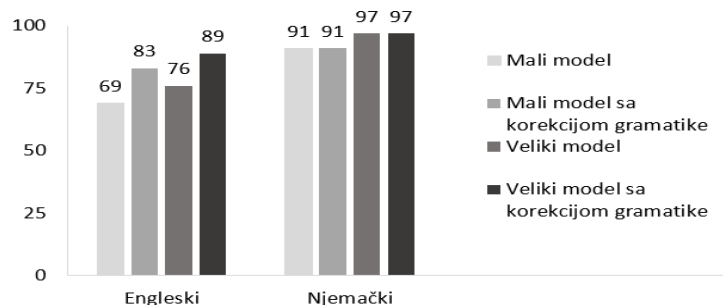
Tabela 6.21: Njemački jezik / audio datoteka / veliki model / bez korekcija gramatike

Kategorija	Organizaciona struktura			Mrežna infrastruktura			Prosjek		
	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]
Klase	92	100	96	100	94	97	96	97	97
Atributi	40	75	52	50	70	59	45	73	56
Asocijacije	70	100	82	100	100	100	85	100	91
Generalizacije	100	100	100	84	100	90	92	100	95
Ukupno za studiju	61	89	72	69	83	75	65	86	74

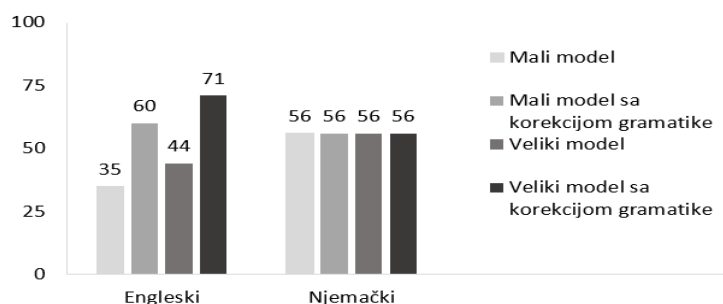
Tabela 6.22: Njemački jezik / audio datoteka / veliki model / sa korekcijom gramatike

Kategorija	Organizaciona struktura			Mrežna infrastruktura			Prosjek		
	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]	R[%]	P[%]	F[%]
Klase	92	100	96	100	94	97	96	97	97
Atributi	40	75	52	50	70	59	45	73	56
Asocijacije	70	100	82	100	100	100	85	100	91
Generalizacije	100	100	100	84	100	90	92	100	95
Ukupno za studiju	61	89	72	69	83	75	65	86	74

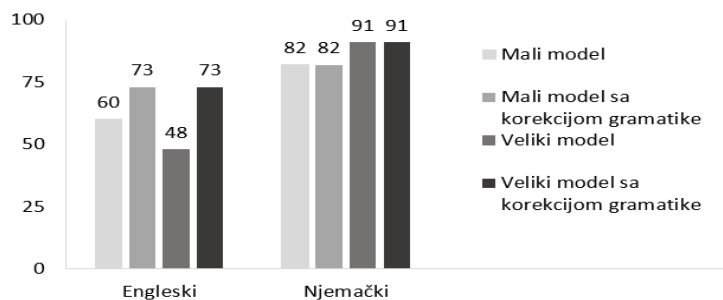
Na sl. 6.9, 6.10, 6.11 i 6.12 prikazane su prosječne efektivnosti sinteze klasa, atributa, asocijacija i generalizacije, respektivno.



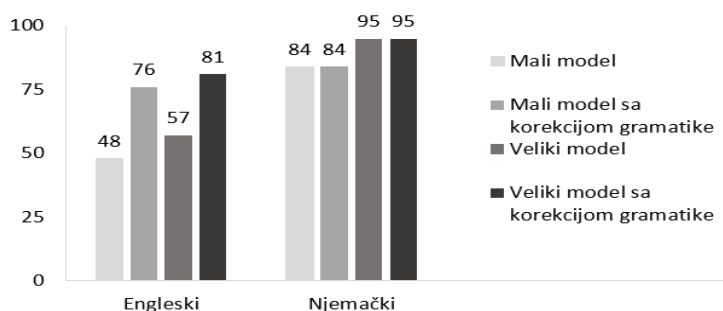
Slika 6.9: Prosječna efektivnost sinteze klasa



Slika 6.10: Prosječna efektivnost sinteze atributa



Slika 6.11: Prosječna efektivnost sinteze asocijacija



Slika 6.12: Prosječna efektivnost sinteze generalizacija

Glava 7

Zaključak

Projektovanje baze podataka na osnovu govora podrazumijeva prolazak kroz četiri faze projektovanja a to su prikupljanje i dokumentovanje zahtjeva, konceptualno modelovanje, relaciono modelovanje i generisanje fizičke šeme baze podataka. Automatizacija ovog procesa podrazumijeva prolazak kroz kompletan proces bez potrebe za ručnom intervencijom u bilo kojoj fazi. Upotreba govora u projektovanju baze podataka podrazumijeva dokumentovanje korisničkih zahtjeva u audio formatu i obradu audio podataka kako bi se izdvojili podaci bitni za generisanje fizičke šeme baze podataka. U radu je prikazano softversko rješenje koje automatizuje projektovanje baze podataka na osnovu govora, što znači da je izvršena automatizacija cjelokupnog procesa, počevši od dokumentovanja zahtjeva u audio formatu, pa sve do generisanja fizičke šeme baze podataka. Ukoliko je potrebno, ostavljena je mogućnost izmjene rezultata za svaku od faza projektovanja, što omogućava ispravku u slučaju grešaka.

Implementacija alata podrazumijeva proširenje postojećeg SpeeD alata za generisanje konceptualnog modela na osnovu govora. Proširenje je realizovano razvojem dodatnih komponenata za prepoznavanje govora, komponenta za provjeru gramatike i pravopisa, zajedno sa integracijom sa TexToData i AMADEOS alatima. Implementirane komponente služe za generisanje odgovarajućeg teksta na osnovu audio podataka i ispravku grešaka dobijenog teksta. Komponenta za prepoznavanje govora omogućava prepoznavanje na osnovu priložene audio datoteke ili govora putem mikrofona. U oba slučaja rezultati prepoznavanja se dobijaju na klijentskoj strani u realnom vremenu. Dobijeni tekst predstavlja tekstualnu specifikaciju. Nakon dobijanja teksta, komponenta za provjeru gramatike može poslužiti za ispravku potencijalnih grešaka, što ima za cilj generisanje što tačnije fizičke šeme baze podataka. Ova komponenta je implementirana korišćenjem nekoliko alata i biblioteka. Integracija sa TexToData alatom omogućava konceptualno modelovanje na osnovu tekstualne specifikacije, dok integracija sa AMADEOS alatom omogućava relaciono modelovanje na osnovu konceptualnog modela, generisanje DDL skripte i fizičke šeme baze podataka.

Implementirani pristup omogućava rad sa različitim jezicima, izvorima audio podataka (upotrebom mikrofona ili audio datoteke) i alatima za provjeru gramatike koji utiču na krajnje generisanu fizičku šemu baze podataka. SpeeD predstavlja prvi alat za automatizaciju projektovanja baze podataka na osnovu govora.

Literatura

- [1] P. P.-S. Chen, “English sentence structure and entity-relationship diagrams,” *Information Sciences*, vol. 29, no. 2-3, pp. 127–149, 1983.
- [2] I. Y. Song, Y. Zhu, H. Ceong, O. Thonggoom, “Methodologies for semi-automated conceptual data modeling from requirements,” in *Proc. of ER 2015*, pp. 18–31, 2015.
- [3] D. Brdjanin, S. Maric, “Model-driven techniques for data model synthesis,” *Electronics*, vol. 17, no. 2, pp. 130–136, 2013.
- [4] I. Y. Song, Y. Zhu, H. Ceong, O. Thonggoom, “Heuristics-based entity-relationship modelling through natural language processing,” in *Proc. of AICS 2004*, pp. 302–313, 2004.
- [5] H. Harmain, R. Gaizauskas, “Cm-builder: A natural language-based case tool for object-oriented analysis,” *Automated Software Engineering*, vol. 10, no. 2, pp. 157–181, 2003.
- [6] S. Puroo, “Apsara: A tool to automate system design via intelligent pattern retrieval and synthesis,” *SIGMIS Database*, vol. 29, no. 4, pp. 45–57, 1998.
- [7] J. Choobineh, A. W. Lo, “Cabsydd: Case-based system for database design,” *Journal of Management Information Systems*, vol. 21, no. 3, pp. 281–314, 2004.
- [8] V. Sugumaran, V. C. Storey, “Ontologies for conceptual modeling: Their creation, use, and management,” *Data & Knowledge Engineering*, vol. 42, no. 3, pp. 251–271, 2002.
- [9] O. Thonggoom, *Semi-automatic conceptual data modelling using entity and relationship instance repositories*. PhD thesis, Drexel University, 2011.
- [10] “TexToData.” <http://m-lab.etf.unibl.org:8080/Textodata/>.
- [11] “AMADEOS.” <http://m-lab.etf.unibl.org:8080/amadeos/>.
- [12] Z. Spasic, A. Vukotic, D. Brdjanin, D. Banjac, G. Banjac, “UML-based forward database engineering,” in *Proc. of INFOTEH 2023*, pp. 1–6, 2023.
- [13] J. Meng, J. Zhang, H. Zhao, “Overview of the speech recognition technology,” in *Proc. of the 4th Int. Conf. on Computational and Information Sciences*, pp. 199–202, 2012.
- [14] M. Malik, M. K. Malik, K. Mehmood, I. Makhdoom, “Automatic speech recognition: a survey,” *Multimedia Tools and Applications*, vol. 80, no. 6, pp. 9411–9457, 2021.
- [15] D. Brdjanin, G. Banjac, N. Babic, N. Golubovic, “Towards the speech-driven database design,” in *Proc. of TELFOR 2022*, pp. 1–4, IEEE, 2022.
- [16] B. Millidge, A. K. Seth, C. L. Buckley, “Predictive coding: A theoretical and experimental review,” *ArXiv*, vol. abs/2107.12979, 2022.
- [17] Z. K. Abdul, A. B. K. Al-Talabani, “Mel frequency cepstral coefficient and its applications: A review,” *IEEE Access*, vol. 10, pp. 122136–122158, 2022.
- [18] P. Senin, “Dynamic time warping algorithm review,” Tech. Rep. 855, Information and Computer Science Department, University of Hawaii at Manoa, Honolulu, USA, 2008.

- [19] A. Trivedi, N. Pant, P. Shah, S. Sonik, S. Agrawal, "Speech to text and text to speech recognition systems a review," *IOSR Journal of Computer Engineering*, vol. 20, no. 2, pp. 36–43, 2018.
- [20] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, K. Veselý, "The kaldi speech recognition toolkit," in *Proc. of IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*, pp. 1–4, 2011.
- [21] A. Hannun, C. Case, J. Casper, B. Catanzaro, G. Diamos, E. Elsen, R. Prenger, S. Satheesh, S. Sengupta, A. Coates, A. Y. Ng, "Deep speech: Scaling up end-to-end speech recognition," *ArXiv*, vol. abs/1412.5567, 2014.
- [22] "LinTo." <https://linto.ai>.
- [23] "CMU Sphinx." <https://cmusphinx.github.io>.
- [24] "VOSK." <https://alphacephei.com/vosk/>.
- [25] T. F. Pereira, C. E. Salgado, A. L. Lima, R. J. Machado, "A web-based voice interaction framework proposal for enhancing information systems user experience," in *Proc. Computer Science*, vol. 196, pp. 235–244, 2021.
- [26] "ProWritingAid." <https://prowritingaid.com>.
- [27] "Jazzy." <https://github.com/realml/jazzy>.
- [28] "LanguageTool." <https://languagetool.org>.
- [29] "Speed." <http://m-lab.etf.unibl.org:8080/Speed/>.
- [30] S. Ram, S. M. Curran, "An automated tool for relational database design," *Information Systems*, vol. 14, no. 3, pp. 247–259, 1989.
- [31] D. W. Embley, W. Y. Mok, "Mapping conceptual models to database schemas," in *Handbook of Conceptual Modeling*, pp. 123–163, 2011.
- [32] "DBomnina." <http://m-lab.etf.unibl.org:8080/dbomnina/>.
- [33] D. Brdjanin, M. Grumic, G. Banjac, M. Miscovic, I. Dujlovic, A. Kelec, N. Obradovic, D. Banjac, D. Volas, S. Maric, "Towards an online multilingual tool for automated conceptual database design," in *Proc. of IDC 2022*, pp. 144–153, 2022.
- [34] I. Lukovic, P. Mogin, J. Pavicevic, S. Ristic, "An approach to developing complex database schemas using form types," *Software: Practice & Experience*, vol. 37, no. 15, pp. 1621–1656, 2007.
- [35] D. Brdjanin, G. Banjac, D. Keserovic, N. Babic, N. Golubovic, "Combining speech processing and text processing in conceptual database design," *Proc. of TELFOR 2024*, vol. 16, no. 1, pp. 8–13, 2024.
- [36] "Oracle Autonomous Database." <https://www.oracle.com/autonomous-database/>.
- [37] "Microsoft Azure SQL Database." <https://azure.microsoft.com/en-us/products/azure-sql/database>.
- [38] "GoogleSpeechToText." <https://cloud.google.com/speech-to-text>.
- [39] "AmazonTranscribe." <https://aws.amazon.com/transcribe/>.
- [40] "AzureAISpeech." <https://azure.microsoft.com/en-us/products/ai-services/ai-speech>.
- [41] "CMUSphinxLib." <https://cmusphinx.github.io/wiki/>.
- [42] J. Zou, Y. Han, S. S. So, "Overview of artificial neural networks," *Methods Mol Biol*, vol. 458, pp. 15–23, 2008.

-
- [43] J. Barker, S. Watanabe, E. Vincent, J. Trmal, “The fifth ‘chime’ speech separation and recognition challenge: Dataset, task and baselines,” in *Proc. INTERSPEECH 2018*, pp. 1561–1565, 2018.
- [44] Z. S. Taghavi, A. Satvathy, H. Sameti, “A change of heart: Improving speech emotion recognition through speech-to-text modality conversion,” arXiv preprint arXiv:2307.11584 [cs.SD], 2023.
- [45] “Sapling.” <https://sapling.ai>.
- [46] “TextGears.” <https://textgears.com>.
- [47] “TextRazor.” <https://www.textrazor.com>.

Biografija

Dejan Keserović, dipl. inž. elektrotehnike, rođen je 27.11.1996. godine u Banjoj Luci. Završio osnovnu školu u Banjoj Luci 2011. godine. Ekonomsku školu, smjer Ekonomski tehničar, završio 2015. godine u Banjoj Luci. Osnovne studije, na studijskom programu Računarstvo i informatika, upisao je 2015. godine na Elektrotehničkom fakultetu Univerziteta u Banjoj Luci, a diplomirao u septembru 2022. godine na temu "HIDS sistemi", sa prosječnom ocjenom studiranja 7,37. Drugi ciklus studija upisao 2022. godine na Elektrotehničkom fakultetu u Banjoj Luci, studijski program Računarstvo i informatika. Prosječna ocjena položenih ispita je 9,60.

Profesionalni angažman: Od aprila 2022. godine zaposlen kao programer u kompaniji "Lanaco d.o.o."

Objavljeni radovi kandidata na kojima je (ko)autor:

1. D. Brdjanin, G. Banjac, D. Keserovic, N. Babic, N. Golubovic, "Combining speech processing and text processing in conceptual database design," Telfor Journal, vol. 16, no. 1, pp. 8–13, 2024.

УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ
ПОДАЦИ О АУТОРУ ОДБРАЊЕНОГ МАСТЕР/МАГИСТАРСКОГ РАДА

Име и презиме аутора мастер/магистарског рада: **Дејан Кесеровић**

Датум, мјесто и држава рођења аутора: **27.11.1996. године, Бања Лука, Босна и Херцеговина**

Назив завршеног факултета/Академије аутора и година дипломирања:

Електротехнички факултет Универзитета у Бањој Луци, 2022. године

Датум одбране завршног/дипломског рада аутора: **06.09.2022. године**

Наслов завршног/дипломског рада аутора: **HIDS системи**

Академско звање коју је аутор стекао одбраном завршног/дипломског рада:

дипломирани инжењер електротехнике - 240 ECTS – рачунарство и информатика

Академско звање које је аутор стекао одбраном мастер/магистарског рада:

мастер електротехнике – 300 ECTS – рачунарство и информатика

Назив факултета/Академије на коме је мастер/магистарски рад одбрањен:

Електротехнички факултет Универзитета у Бањој Луци

Наслов мастер/магистарског рада и датум одбране: **Аутоматизовано пројектовање**

базе података на основу говора, 11.02.2025. године

Научна област мастер/магистарског рада према CERIF шифрарнику: **T120**

Имена ментора и чланова комисије за одбрану мастер/магистарског рада:

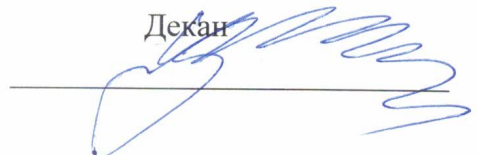
проф. др Зоран Ђурић, предсједник

проф. др Дражен Брђанин, ментор

проф. др Јован Галић, члан

У Бањој Луци, дана 28. 1. 2025.

Декан



ИЗЈАВА О АУТОРСТВУ

**Изјављујем да је
мастер/магистарски рад**

Наслов рада: Аутоматизовано пројектовање базе података на основу говора

Наслов рада на енглеском језику: Automated speech-based database design

- резултат сопственог истраживачког рада,
- да мастер/магистарски рад, у цјелини или у дијеловима, није био предложен за добијање било које дипломе према студијским програмима других високошколских установа,
- да су резултати коректно наведени и
- да нисам кршио/ла ауторска права и користио интелектуалну својину других лица.

У Бањој Луци 28.01.2025.

Потпис кандидата

Дејан Кесеровић

Изјава којом се овлашћује Електротехнички факултет/ Академија умјетности Универзитета у Бањој Луци да мастер/магистарски рад учини јавно доступним

Овлашћујем Електротехнички факултет/ Академију умјетности Универзитета у Бањој Луци да мој мастер/магистарски рад, под насловом

Аутоматизовано пројектовање базе података на основу говора

који је моје ауторско дјело, учини јавно доступним.

Мастер/магистарски рад са свим прилозима предао/ла сам у електронском формату, погодном за трајно архивирање.

Мој мастер/магистарски рад, похрањен у д и г и т а л н и р е п о з и т о р и ј у м Универзитета у Бањој Луци, могу да користе сви који поштују одредбе садржане у одабраном типу лиценце Креативне заједнице (*Creative Commons*), за коју сам се одлучио/ла.

1. Ауторство
2. Ауторство - некомерцијално
3. Ауторство - некомерцијално - без прераде
4. Ауторство - некомерцијално - дијелити под истим условима
5. Ауторство - без прераде
6. Ауторство - дијелити под истим условима

(Молимо да заокружите само једну од шест понуђених лиценци, кратак опис лиценци дат је на полеђини листа).

У Бањој Луци 28.01.2025.

Потпис кандидата

Ђејан Чекеролит

**Изјава о идентичности штампане и електронске верзије
мастер/магистарског рада**

Име и презиме аутора: Дејан Кесеровић

Наслов рада: Аутоматизовано пројектовање базе података на основу говора

Ментор: проф. др Дражен Брђанин

Изјављујем да је штампана верзија мог мастер/магистарског рада идентична електронској верзији коју сам предао/ла за дигитални репозиторијум Универзитета у Бањој Луци.

У Бањој Луци 28.01.2025.

Потпис кандидата

Дејан Кесеровић

Проф. др Зоран Ђурић
Електротехнички факултет Универзитета у Бањој Луци

Проф. др Дражен Брђанин
Електротехнички факултет Универзитета у Бањој Луци

Проф. др Јован Галић
Електротехнички факултет Универзитета у Бањој Луци

УНИВЕРЗИТЕТ У БАЊОЈ ЛУЦИ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ
БАЊАЛУКА

Број: 91

Датум: 15-1. 2025.

**НАУЧНО-НАСТАВНОМ ВИЈЕЋУ
ЕЛЕКТРОТЕХНИЧКОГ ФАКУЛТЕТА
УНИВЕРЗИТЕТА У БАЊОЈ ЛУЦИ**

Одлуком Научно-наставног вијећа Електротехничког факултета Универзитета у Бањој Луци, број 20/3.999-6/24 од 13.12.2024. године, именовани смо за чланове Комисије за завршни рад II циклуса студија, под називом "**Аутоматизовано пројектовање базе података на основу говора**", кандидата **Дејана Кесеровића**, дипл. инж. електротехнике. Након прегледа приложеног рада подносимо сљедећи

ИЗВЈЕШТАЈ

1. БИОГРАФСКИ ПОДАЦИ КАНДИДАТА

Дејан Кесеровић, дипл. инж. електротехнике, рођен је 27.11.1996. године у Бањој Луци. Основну школу завршио је у Бањој Луци 2011. године. Средњу економску школу, смјер Економски техничар, завршио је у Бањој Луци 2015. године.

Основне студије, на студијском програму Рачунарство и информатика, уписао је 2015. године на Електротехничком факултету Универзитета у Бањој Луци, а дипломирао у септембру 2022. године на тему "HIDS системи", са просјечном оцјеном студирања 7,37.

Други циклус студија уписао је 2022. године на Електротехничком факултету Универзитета у Бањој Луци, студијски програм Рачунарство и информатика, на којем је положио све испите са просјечном оцјеном 9,60. У октобру 2024. године изгубио је статус студента и поново уписао студије другог циклуса на истом студијском програму, уз признавање свих претходно положених испита.

Од априла 2022. године запослен је као програмер у компанији "Lanaco" у Бањој Луци.

Кандидат је до сада, као коаутор, објавио један научни рад у часопису међународног значаја:

1. D. Brdjanin, G. Banjac, **D. Keserovic**, N. Babic, N. Golubovic: Combining speech processing and text processing in conceptual database design, *Telfor Journal*, vol. 16, no. 1, pp. 8–13, 2024.

2. ОСНОВНИ ПОДАЦИ О РАДУ

Завршни рад II циклуса кандидата Дејана Кесеровића, под називом "Аутоматизовано пројектовање базе података на основу говора" садржи 77 нумерисаних страница, са 67 слика и 22 табеле, а организован је у седам глава:

1. Увод,
2. Аутоматизовано пројектовање базе података,
3. Препознавање говора,
4. Концептуално моделовање на основу говора,
5. Имплементација приступа,
6. Експериментални резултати,
7. Закључак.

Списак кориштене литературе садржи 47 цитираних извора.

3. АНАЛИЗА РАДА

У уводном дијелу рада прво је описана мотивација за израду рада, а затим су дефинисани предмет и циљеви истраживања, потом су описани методологија истраживања и главни доприноси. Основну мотивацију за истраживање и израду завршног рада представља чињеница да се у постојећим алатима за аутоматизовано пројектовање базе података као полазни основ за пројектовање користе текст и модели (који су вјештачки), за разлику од говора (који је природан и човјеку својствен). Једино *Speed* алат омогућава аутоматско генерисање концептуалног модела на основу снимљеног говора, али подржава само енглески језик и не омогућава директни инжењеринг базе података на основу генерисаног концептуалног модела. Зато је предмет истраживања у овом раду анализа могућности употребе алата за аутоматско препознавање говора и алата за провјеру граматике и правописа у препознатом говору у циљу унапређења *Speed* алата и проширења функционалности додавањем подршке за различите природне језике, омогућавањем обраде и *online* и *offline* говора, омогућавањем аутоматизоване провјере граматике и правописа те аутоматизацијом директног инжењеринга циљне базе података.

У другој глави прво је приказан типичан процес пројектовања релационе базе података, од концептуалног пројектовања до генерисања физичке шеме базе података, а затим су детаљније приказани постојећи приступи за аутоматизовано генерисање концептуалног модела на основу различитих извора. Детаљније су приказани најзначајнији алати, с фокусом на постојећу имплементацију *Speed* алата.

У трећој глави, након уводног прегледа историјског развоја приступа за препознавање говора, приказан је процес и описане су главне активности препознавања говора. Потом су приказане најзначајније технике и алати, с фокусом на *Kaldi* и *Vosk* који се користи у *Speed* алату и који је у фокусу истраживања.

У четвртој глави приказан је предложени приступ за аутоматизовано пројектовање концептуалног модела на основу говора, који комбинује технике за препознавање говора и технике за обраду препознатог говора. У првој фази врши се препознавање говора, при чему су подржани различити природни језици, након чега се врше корекције правописа и граматике у препознатом тексту. У другој фази врши се обрада текста и генерише кореспондентни концептуални модел базе података, који се репрезентује UML дијаграмом класа.

У петој глави описана је имплементација предложеног приступа и приказан унапријеђени *Speed* алат. У имплементацији је вишеструко примијењен *reuse* принцип – све главне функционалности омогућене су интеграцијом постојећих софтверских компонената: (а) за обраду говора користи се *Vosk* са десетак језичких модела за различите природне језике, (б) за корекције граматике и правописа омогућено је кориштење пет интерних (*Jazzy*, *LanguageTool*) и екстерних (*ProWritingAid*, *Sapling*, *TextGears*) сервиса, (в) за обраду препознатог говора и генерисање концептуалног модела користи се колекција *TexToData* сервиса, (г) за директни инжењеринг шеме циљне базе података користи се колекција *AMADEOS* сервиса. Након детаљног приказа серверске и клијентске стране унапријеђеног алата, дат је један илустративни примјер аутоматизованог пројектовања базе података на основу снимљеног говора.

Предложени приступ и имплементирано унапређење *Speed* алата евалуирани су у шестој глави. Евалуација је извршена на основу снимљених говора на енглеском и на њемачком језику, за два различита домена (*организациона структура* и *мрежна инфраструктура*), кориштењем великих и малих језичких модела, без или са примјеном алата за корекцију граматике и правописа. Спроведена квалитативна и квантитативна анализа показују да предложени приступ и имплементирани алат омогућавају аутоматизовану синтезу концептуалног модела са (релативно) високом комплетношћу и прецизношћу. Просјечна ефективност (*F-мјера*) у комплетном експерименту је 68%. Најмања ефективност синтезе комплетног модела је 50% (на основу снимљеног говора на енглеском језику, уз кориштење малог језичког модела и без корекција граматике и правописа), а највећа је 74% (на основу за снимљеног говора на њемачком језику, уз кориштење великог језичког модела и са корекцијама граматике и правописа).

У седмој глави дати су закључци спроведеног истраживања, након чега је указано на могућности наставка истраживања.

4. НАЈВАЖНИЈИ ДОПРИНОСИ

Комисија сматра да је кандидат, кроз спроведено истраживање, реализовао завршни рад II циклуса студија, који садржи више значајних доприноса, од којих су најважнији сљедећи:

1. Предложен је и у потпуности имплементиран приступ за аутоматизовано пројектовање релационе базе података на основу говора, што укључује синтезу концептуалног модела на основу *online* и *offline* говора на различитим природним језицима, уз могућност аутоматизоване корекције правописа и граматике, након чега је омогућен директни инжењеринг шеме циљне базе података.


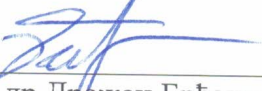

2. Извршен је реинжењеринг *Speed* алата, који је укључивао развој нових и интеграцију постојећих хетерогених софтверских компонената, чиме је функционалност алата проширена обрадом *online* говора, подршком за различите природне језике, корекцијом правописа и граматике те директним инжењерингом шеме релационе базе података за неколико савремених система за управљање релационим базама података, при чему се у комплетном процесу пројектовања користи стандардна UML нотација.
3. Приступ је експериментално верификован, а остварени резултати показују значајан потенцијал за практичне примјене и наставак истраживања.

5. ЗАКЉУЧАК И ПРИЈЕДЛОГ

Комисија сматра да завршни рад II циклуса, под називом "**Аутоматизовано пројектовање базе података на основу говора**", кандидата Дејана Кесеровића, садржи све потребне елементе и резултате којима су остварени постављени циљеви истраживања и са задовољством предлаже Научно-наставном вијећу Електротехничког факултета Универзитета у Бањој Луци да усвоји извјештај Комисије и одобри заказивање усмене јавне одбране.

Бања Лука, 13.1.2025. године

КОМИСИЈА

1. 
Проф. др Зоран Бурић, предсједник
2. 
Проф. др Дражен Брђанин, ментор
3. 
Проф. др Јован Галић, члан