

Računalna podrška za praćenje razvojnih timova pri korištenju sustava virtualne stvarnosti za pregled konstrukcije

Brnčić, Marko

Undergraduate thesis / Završni rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:235:733758>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-10-15**

Repository / Repozitorij:

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Marko Brnčić

Zagreb, 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

Računalna podrška za praćenje razvojnih timova pri korištenju sustava virtualne stvarnosti za pregled konstrukcije

Mentori:

Doc. dr. sc. Stanko Škec

Student:

Marko Brnčić

Zagreb, 2022.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru, doc. dr. sc. Stanku Škecu na pruženoj prilici, strpljenju, pomoći i pruženom vodstvu tijekom izrade ovog rada.

Zahvaljujem se i Nikoli Horvatu, mag. ing. mech. na savjetima, podršci i pomoći tijekom izrade ovog rada.

Marko Brnčić



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite
Povjerenstvo za završne i diplomske ispite studija strojarstva za smjerove:
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo
materijala i mehatronika i robotika

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa: 602 – 04 / 22 – 6 / 1	
Ur.broj: 15 - 1703 - 22 -	

ZAVRŠNI ZADATAK

Student: **Marko Brnčić** JMBAG: **0035215660**

Naslov rada na hrvatskom jeziku: **Računalna podrška za praćenje razvojnih timova pri korištenju sustava virtualne stvarnosti za pregled konstrukcije**

Naslov rada na engleskom jeziku: **Computer support for analysing design review teams while using virtual reality systems**

Opis zadatka:

Upotreba tehnologija za virtualnu stvarnost (VS) u području konstruiranja poznata je nekoliko desetljeća. S napretkom tehnologije, razvijeni su VS sustavi čiji se zasloni pozicioniraju na glavu korisnika (eng. Head-mounted display, HMD) te koji odnedavno omogućuju kvalitetno provođenje pregleda konstrukcija u virtualnom okruženju. S ciljem boljeg razumijevanja načina na koji virtualna stvarnost pomaže konstruktorima, važno je pratiti korištenje novih tehnologija tijekom razvojnih aktivnosti kako bi tvrtke mogle odlučiti koju tehnologiju upotrijebiti za svoje specifične situacije.

Kako je suradnja između članova konstrukcijskih timova jedan od glavnih aspekata koji je potrebno pratiti pri korištenju nove tehnologije, cilj ovog završnog rada razviti je i testirati računalnu podršku za praćenje pregleda konstrukcija tijekom korištenja virtualne stvarnosti.

U radu je potrebno:

- Proučiti i usporediti postojeće računalne alate za pregled konstrukcija u okruženju virtualne stvarnosti temeljem pregleda literature.
- Analizirati postupke pregleda konstrukcija u okruženju virtualne stvarnosti te usporediti s tradicionalnim načinima.
- Razviti računalnu podršku za analizu suradnje tijekom aktivnosti pregleda konstrukcija
- Predložiti preporuke za uporabu tehnologija virtualne stvarnosti za pregled konstrukcija u razvoju proizvoda.

Opseg računalne podrške dogovoriti će se tijekom izrade rada.

U radu navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

30. 11. 2021.

Datum predaje rada:

1. rok: 24. 2. 2022.
2. rok (izvanredni): 6. 7. 2022.
3. rok: 22. 9. 2022.

Predvideni datumi obrane:

1. rok: 28. 2. – 4. 3. 2022.
2. rok (izvanredni): 8. 7. 2022.
3. rok: 26. 9. – 30. 9. 2022.

Zadatak zadao:

Doc. dr. sc. Stanko Škec

Predsjednik Povjerenstva:

Prof. dr. sc. Branko Bauer

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	II
POPIS TABLICA.....	III
POPIS OZNAKA	IV
SAŽETAK.....	V
SUMMARY	VI
1. UVOD.....	1
2. PREGLED LITERATURE.....	3
2.2. Virtualna stvarnost	4
2.3. Praćenje kolaborativnih aktivnosti.....	5
2.4. Segmentacija govora	7
2.5. Zahtjevi za razvoj računalne podrške.....	10
3. RAČUNALNA PODRŠKA ZA ANALIZU SURADNJE	11
3.1. Priprema ulaznih podataka.....	11
3.2. Računalni program za analizu aktivnosti pregleda konstrukcije.....	12
3.2.1. Učitavanje podataka.....	13
3.2.2. Priprema podataka	16
3.2.3. Analiza podataka.....	21
3.2.4. Ispis rezultata	25
4. PRIMJENA PROGRAMA NA PRIMJERU	27
4.1. Priprema ulaznih podataka.....	27
4.2. Rad računalne podrške na primjeru	30
4.3. Rezultati analize primjera	31
5. DISKUSIJA	37
5.1. Osvrt na računalnu podršku	37
5.2. Dodane vrijednosti	38
5.3. Budući razvoj računalne podrške	39
6. ZAKLJUČAK.....	41
LITERATURA.....	42
PRILOZI.....	47

POPIS SLIKA

Slika 1	Kolaborativni pregled konstrukcije (lijevo – Creo View, desno – VRSmart) [17].	5
Slika 2	Dijagram toka pripreme ulaznih podataka	12
Slika 3	Dijagram toka programa.....	13
Slika 4	Primjer varijable <i>data_separate</i> i jedne ćelije u njoj	14
Slika 5	Primjer varijable <i>data</i> i jedne ćelije u njoj	14
Slika 6	Primjer varijable <i>raw_audio</i> i jedne ćelije u njoj	15
Slika 7	Primjer varijable <i>data_table</i> prije uklanjanja preklopa.....	15
Slika 8	Dijagram toka učitavanja podataka	16
Slika 9	Dijagram toka pripreme podataka	18
Slika 10	Algoritam za uklanjanje preklopa	20
Slika 11	Dijeljenje segmenta <i>n</i> na dva dijela za slučaj preklopa četiri.....	20
Slika 12	Primjer varijable <i>data_seconds</i>	23
Slika 13	Primjer varijable " <i>turns_table</i> "	24
Slika 14	Dijagram toka analize podataka	25
Slika 15	Dijagram toka ispisa rezultata	26
Slika 16	Pogledi korisnika u VS-u u jednom trenutku pregleda konstrukcije.....	27
Slika 17	Usporedba rezolucija automatske segmentacije	29
Slika 18	Usporedba kumulativnog trajanja govora ručne i automatske segmentacije na engleskom jeziku	30
Slika 19	Usporedba segmentacije za primjer na engleskom jeziku.....	30
Slika 20	Usporedba <i>data_table</i> prije (lijevo) i poslije (desno) uklanjanja preklopa	31
Slika 21	Graf kumulativnog vremena govora.....	32
Slika 22	Graf kumulativnog vremena govora normaliziran s obzirom na ukupno trajanje sesije	33
Slika 23	Graf kumulativnog vremena govora normaliziran s obzirom na ukupno trajanje govora svakog sudionika zasebno	33
Slika 24	Graf pokretnih prosjeka.....	34
Slika 25	Graf normaliziranih pokretnih prosjeka	35

POPIS TABLICA

Tablica 1	Tipovi preklopa	17
Tablica 2	Ukupno vrijeme trajanja govora sudionika	32
Tablica 3	Tablica interakcija članova tima.....	36

POPIS OZNAKA

Oznaka	Jedinica	Opis
<i>data</i>	-	Varijabla tipa "cell" koja sadrži n ćelija (jedna za svakog sudionika), informacije o početku, kraju i trajanju segmenata
<i>data_seconds</i>	-	$m \times n$ matrica gdje je m trajanje sesije u sekundama s dodatkom za sigurnost, a n je broj sudionika
<i>data_separate</i>	-	Varijabla tipa "cell" koja sadrži n ćelija (jedna za svakog sudionika i svaki dio sesije), informacije o početku, kraju i trajanju segmenata
<i>data_table</i>	-	Varijabla tipa "table" s pet stupaca, informacije o svim segmentima
<i>i</i>	-	Varijabla brojač
<i>n</i>	-	Varijabla brojač
<i>raw_audio</i>	-	Varijabla tipa "cell" koja sadrži n ćelija (jedna za svakog sudionika). Neobrađeni podaci audio zapisa
<i>turns_table</i>	-	Varijabla tipa "table" s pet stupaca, rezultati analize

SAŽETAK

Pregled konstrukcije je aktivnost koja se provodi tijekom razvoja proizvoda. Razvoj novih tehnologija omogućuje unaprjeđenje procesa pregleda konstrukcije. Jedna od tih tehnologija koja se posebno ističe je virtualna stvarnost (VS). VS omogućuje intuitivniji način vizualizacije i interakcije s pregledavanom konstrukcijom. Kako bi znali gdje i kada primijeniti VS za pregled konstrukcije, potrebno je pratiti razvojne timove pri korištenju VS-a. Jedan od aspekata koje je potrebno pratiti je suradnja razvojnih timova. Popularna metoda za analizu suradnje je analiza protokola. Analiza komunikacijske strukture je važan dio analize protokola koji nam pruža uvid u interakcije sudionika tijekom pregleda konstrukcije. Kako bi se olakšala analiza komunikacijske strukture razvojnih timova tijekom pregleda konstrukcije, razvijena je računalna podrška za provedbu analize. U ovom radu razvijen je MATLAB računalni program kao računalna podrška. Računalni program kao ulazne podatke koristi audio zapise i segmentirani govor pregleda konstrukcije, a daje nam rezultate analize komunikacijske strukture. Rezultati koje daje računalna podrška se ispisuju kao grafovi kumulativnog trajanja govora i pokretnih prosjeka (eng. moving average) govora te tablica s rezultatima Markovljeve analize. Na jednom primjeru pregleda konstrukcije četveročlanog tima je prikazan rad računalnog programa. Prikazani su i rezultati dobiveni analizom primjera pomoću računalne podrške. Dobiveni rezultati omogućavaju uspoređivanje različitih tipova pregleda konstrukcija i određivanje dominantnih sudionika tijekom pregleda. Prepoznata su tri potencijalna smjera za daljnji razvoj: prilagodba programa za praćenje suradnje timova u stvarnom vremenu, dodavanje mogućnosti analize s kontekstom i unaprjeđenje korisničkog iskustva.

Ključne riječi: Virtualna Stvarnost, pregled konstrukcija, komunikacijska struktura, računalna podrška, MATLAB

SUMMARY

Design review is a part of most product designs. The development of new technologies allows for improvements in the design review process. A new technology that stands out is virtual reality (VR). VR allows for more intuitive interactions with and visualizations of the product which is being reviewed. To determine how best to use VR for design reviews, it is important to study how design teams use VR. One aspect that should be studied is how they collaborate. A popular method for studying collaboration is protocol analysis. An important part of protocol analysis is analysing the communication structure. Analysing communicational structures allows insight into how teams interact during design reviews. To make analysing communicational structures of design teams during design reviews easier, a form of computer support has been developed for analysis. The computer support is a MATLAB computer program. The program uses audio files and segmented speech as input, based on which it gives us the results of communication structure analysis. The results are exported as cumulative talking time graphs, moving average graphs, and a spreadsheet with the results of Markov analysis. A single example of a four-member design team in a design review session has been used to demonstrate the working of the developed program. Results gathered from the analysis of the example are also shown. The results that are exported using the program allow us to compare different design review sessions as well as determine the dominant members during design reviews. Three paths for further development of the program have been recognized: adapting it to allow real-time analysis of design team collaboration, adding the ability to take context into account, and improving the user experience.

Key words: Virtual Reality, design review, communication structure, computer support, MATLAB

1. UVOD

Pregled konstrukcija je ključna aktivnost u procesu razvoja proizvoda [1] tijekom koje se utvrđuje koliko razvijena konstrukcija zadovoljava postavljene zahtjeve [2]. Ti zahtjevi se razlikuju od proizvođača do proizvođača, a mogu uključivati funkcionalne, geometrijske, proizvodne, ergonomske, sigurnosne, financijske i druge zahtjeve. Pregled konstrukcije se često provodi kao kolaborativna aktivnost, a tim za pregled konstrukcije može se sastojati od različitih članova. Česta podjela je na konstruktore i recenzente [3]. Konstruktori su članovi tima koji su sudjelovali u konstruiranju te su već upoznati s konstrukcijom, a recenzenti su članovi tima kojima je cilj utvrditi u kojoj mjeri konstrukcija zadovoljava zahtjeve, ili davati prijedloge za unaprjeđenja konstrukcije, a često vrše obje funkcije [3].

Tijekom razvoja proizvoda, uobičajeno je stvarati digitalne modele proizvoda [4]. Takvi digitalni modeli se obično koriste prilikom pregleda konstrukcije jer se pomoću njih može zaključiti puno bitnih informacija o konstrukciji. Već dugu niz godina se u svrhu stvaranja digitalnih modela koriste 3D CAD (eng. Computer Aided Design) računalni programi te danas imaju ulogu u razvoju gotovo svakog proizvoda. Takvi modeli pružaju vrlo prikladan način za razmjenu informacija između konstruktora unutar tvrtke, klijenata i različitih vanjskih suradnika koji sudjeluju u razvoju proizvoda. Također su vrlo pogodni za prikaz dosadašnjeg napretka klijentima koji nisu nužno stručnjaci u tehničkim područjima [5]. Unutar 3D CAD modela, osim što su sadržane informacije o geometriji modela, mogu se integrirati i informacije o materijalima, proizvodnji, nabavi itd.

Za potrebe pregleda konstrukcija potrebno je omogućiti intuitivnu interakciju s kreiranim CAD modelima. Od samog početka razvoja 2D i 3D CAD alata, koristili su se u razvoju konstrukcija te su i danas 3D CAD alati najčešće u uporabi za vizualizaciju i razmjenu informacija o konstrukciji u svrhu pregleda. U međuvremenu su razvijene i novije tehnologije za vizualizaciju od kojih se posebice ističe virtualna stvarnost (VS) [6].

VS tehnologije razvijaju se već desetljećima, a danas je različite oblike VS-a moguće pronaći i u kućanstvima. U procesu razvoja proizvoda VS se najčešće koristi tijekom pregleda konstrukcija [7]. VS se u kontekstu razvoja proizvoda često pojavljuje u obliku zaslona koji se pričvrsti na glavu korisnika (eng. Head-mounted display, HMD). HMD omogućava prikaz 3D

slike s jednim zaslonom na svakom oku i praćenje kretnji uređaja. Praćenje kretnji je važno kako bi se slika koja se prikazuje mogla dinamično mijenjati s obzirom na položaj glave korisnika. Prema Liu i sur. [6], tijekom pregleda arhitektonskih konstrukcija VS omogućuje intuitivniju interakciju s pregledavanom konstrukcijom te omogućuje korisnicima bolju percepciju prostora.

Kako je uporaba VS-a u pregledu konstrukcija još uvijek relativno nova primjena, potrebno je pratiti kako se koristi u industriji, na koje načine utječe na proces pregleda konstrukcija te koje su optimalne primjene te tehnologije. Za to je potrebno razvijati alate za praćenje razvojnih timova tijekom pregleda konstrukcija kako bi mogli što bolje procijeniti utjecaj uporabe VS-a u toj aktivnosti. Verbalna komunikacija je najdominantniji oblik komunikacije tijekom konstrukcijskih aktivnosti [8]. Zbog toga, važno je promatrati verbalnu komunikaciju sudionika.

U sklopu izrade ovog rada cilj je razviti i testirati računalni program koji će omogućiti računalnu podršku za analizu suradnje sudionika tijekom pregleda konstrukcije u virtualnom okruženju. U svrhu tog cilja, prvo će se proučiti aktivnost pregleda konstrukcije, dosadašnja saznanja o uporabi VS tehnologije u pregledu konstrukcija te na koji način se vrše analize aktivnosti pregleda konstrukcija. Zatim će se opisati razvoj računalnog programa u MATLAB programskom jeziku koji omogućava automatizaciju analize suradnje. Opisat će se sve funkcije programa i svi algoritmi koji su primijenjeni u programu te će se na primjeru prikazati rezultati dobiveni primjenom razvijenog programa. Nakon toga, bit će predložene moguće primjene programa i u kojem smjeru bi se mogla dalje istraživati primjena VS sustava.

2. PREGLED LITERATURE

Za razvijanje alata za analizu pregleda konstrukcija moramo odrediti zahtjeve koje alat treba ispuniti kako bi dobili kvalitetne rezultate. Prvo se moramo bolje upoznati sa samim procesom pregleda, proučiti na koji način se provodi i koji su njegovi doprinosi i ciljevi tijekom razvoja proizvoda. Zatim će se detaljnije opisati VS i istražiti njezina primjena u pregledu konstrukcija te će se na kraju proučiti postojeće metode analize pregleda konstrukcija.

2.1. Pregled konstrukcija

Pregled konstrukcija je ključna kontrolna aktivnost tijekom razvoja proizvoda [9]. Konstrukcije se uglavnom pregledavaju nekoliko puta tijekom razvoja te često služe kako bi se utvrdio dosadašnji napredak u razvoju i kvaliteta konstrukcije. Pregledom se mogu evaluirati potencijalni troškovi proizvodnje, kako se vrši kontrola kvalitete, tijekom proizvodnog procesa, potrebne tehnologije izrade, ekološki utjecaj proizvoda, funkcionalnost proizvoda i sl. [9].

Dhillon [10] prepoznaje 3 glavne vrste pregleda konstrukcija: preliminarni, srednji i krajnji pregled konstrukcije. U preliminarnom pregledu se najčešće pregledavaju koncepti danog proizvoda i preispituju se postavljeni kriteriji uspjeha te se uspoređuju postavljeni koncepti s podacima koji su dobiveni analizom tržišta. Zatim slijedi srednji pregled koji se često događa više puta tijekom razvoja jednog proizvoda, uglavnom tijekom detaljiranja proizvoda. On služi kako bi se kontinuirano pratio razvoj, utvrdili sljedeći koraci u razvoju te vršio osvrt na dosadašnje odluke. Posljednja vrsta, krajnji ili "kritični" pregled konstrukcije, jest onaj koji se događa pred kraj razvoja proizvoda. U tom kritičnom pregledu se provjeravaju konačne komponente kako bi se utvrdilo zadovoljavaju li sve zahtjeve i jesu li konstruirane optimalno u smislu troškova, proizvodnosti i sl. Sama aktivnost pregleda konstrukcija se obično provodi u tri ciklusa: razumijevanje konstrukcije, evaluacija konstrukcije i planiranje daljnjeg razvoja konstrukcije [11]. Razumijevanje je važno kako bi članovi tima koji se do sada nisu upoznali s konstrukcijom mogli kvalitetnije evaluirati konstrukciju.

Pregledi konstrukcija se mogu razlikovati tematski zbog različitih ciljeva pregleda. Ciljevi pregleda se mijenjaju s obzirom na sastav tima koji pregledava konstrukciju, a mogu biti drugačiji i u različitim fazama razvoja proizvoda [11]. U ranijim fazama razvoja se traže alternativna konceptualna rješenja, a u kasnijim fazama se više traže manje optimizacije rješenja [11]. U kasnijim fazama razvoja proizvoda se također češće uključuju recenzenti, koji nisu nužno stručnjaci za konstruiranje, kako bi utvrdili kvalitetu ispunjenosti različitih zahtjeva

[11]. Ti recenzenti mogu biti klijenti koji žele utvrditi ispunjenost njihovih zahtjeva funkcionalnosti, proizvođači kojima je cilj provjeriti proizvodnost elemenata itd.

Tvrtke kontinuirano pokušavaju unaprijediti procese razvoja proizvoda, uključujući i pregled konstrukcija [12]. Jedan od aspekata te aktivnosti koja se može poboljšati je omogućavanje bolje vizualizacije i intuitivnije interakcije s digitalnim modelom proizvoda. Zbog toga se VS sve češće koristi u svrhu pregleda konstrukcija kao alternativa tradicionalnim načinima vizualizacije [6].

2.2. Virtualna stvarnost

Iako razni oblici VS tehnologija postoje već desetljećima, moderne implementacije HMD tipa su relativno nova tehnologija. U kontekstu razvoja proizvoda su često primijenjeni i drugi oblici, kao npr. CAVE (eng. Cave Automatic Virtual Environment) [7] koji koristi "zidove" na kojima se projicira slika ili su sami zidovi zasloni. Korisnik CAVE-a nosi 3D naočale koje omogućuju i praćenje položaja korisnika, što onda omogućuje i dinamičnu promjenu slike na zidovima s obzirom na kretanje korisnika [13]. HMD i CAVE sustavi su dominantni u industriji u odnosu na druge implementacije (npr. 3D naočale s ekranom), a udio HMD-a još raste [14]. Tvrtke koje su implementirale VS često imaju više sustava različitih tipova koje koriste ovisno o potrebama [7]. Postoji i više tipova HMD sustava, a dvije su glavne podijele: na povezane i bežične te s praćenjem „iznutra“ ili „izvana“. Bežični sustavi su oni koji ne zahtijevaju fizičku vezu između HMD-a i okoline, a povezani, najčešće, imaju kabel za komunikaciju s računalom. Očita prednost bežičnih sustava je što korisnik ne mora biti fizički vezan za ništa, ali ograničava protok podataka jer su bežični komunikacijski protokoli još uvijek značajno sporiji u odnosu na one koje koriste kablove. Nadalje, sustavi s praćenjem izvana zahtijevaju postavljanje "kućica" koje prate položaj korisnika u prostoru. Takvi sustavi su precizniji u odnosu na one s praćenjem iznutra koji koriste samo senzore ugrađene u HMD uređaj. Prednost sustava s praćenjem iznutra jest da je prostor u kojemu se korisnik može kretati teoretski neograničen, ako je sustav i bežičan [15].

Kako je već spomenuto, najčešća primjena VS tehnologije u razvoju proizvoda je za pregled konstrukcija. VS omogućava vrlo intuitivan pristup pregledu digitalnih modela jer, osim što dozvoljava 3D prikaz i kretanje korisnika oko konstrukcije, može omogućiti i intuitivne interakcije s konstrukcijom [16]. Wolfartsberger [17] je u svom eksperimentu uspoređivao uspješnost pregleda konstrukcije između Creo View aplikacije s 2D ekranom i VRSmart

aplikacije s HMD VS sustavom [Slika 1]. U tom eksperimentu su se pregledavali veliki i kompleksni modeli industrijskih napajanja. Zaključio je da su grupe koje su koristile VS pronalazile više grešaka u modelu u odnosu na one koje su koristile klasične CAD alate, iako je razlika u broju pronađenih grešaka mala. U tom eksperimentu sudionici su u VS-u mogli hodati i "teleportirati" se kroz prostor oko modela, pregledavati model te vaditi i spajati elemente modela.



Slika 1 Kolaborativni pregled konstrukcije (lijevo – Creo View, desno – VRSmart) [17]

S druge strane, De Casenave i Lugo [15] u svojim eksperimentima gdje su sudionici tražili greške u sklopu, nisu dobili različite rezultate u pronađenim greškama između VS i CAD alata. Horvat i sur. [19] su u svojim eksperimentima zaključili da korisnici u VS-u mogu bolje procijeniti veličine objekata nego s CAD alatima. Freeman i sur. [20] predlažu da se korištenjem VS sustava bogatijih mogućnosti omogućava kvalitetniji pregled kompleksnijih proizvoda.

Istraživanja su često fokusirana na ishode pregleda konstrukcija. Za bolje razumijevanje prednosti i nedostataka uporabe VS-a u pregledu konstrukcija potrebno je razmatrati i kako se sama aktivnost pregleda mijenja kada se koristi VS u usporedbi s tradicionalnim CAD alatima. S obzirom na to da je pregled konstrukcije često kolaborativna aktivnost, korisno je pratiti interakcije sudionika tijekom pregleda.

2.3. Praćenje kolaborativnih aktivnosti

Do sada je najčešća metoda za analizu kolaborativnih aktivnosti, uključujući i konstrukcijskih aktivnosti, analiza protokola [21]. Analiza protokola uključuje proučavanje verbalnih interakcija kako bi bolje razumjeli ponašanja i procese razmišljanja ljudi [22]. Verbalna komunikacija je dominantni oblik komunikacije u konstrukcijskim aktivnostima [8]. Dosadašnja istraživanja su se uglavnom fokusirala na semantičku analizu, ali bitno je promatrati i komunikacijsku strukturu tima koji pregledava konstrukciju. Komunikacijska struktura

uključuje interakcije između sudionika, odnosno kojim se redoslijedom sudionici izmjenjuju, kao i trajanje govora različitih sudionika. Leenders i sur. [23] su u istraživanju s 44 razvojna tima zaključili da postoji povezanost između kreativnosti tima i komunikacijske strukture. Pokazali su da i veliki i mali broj interakcija smanjuje kreativnost te da je kreativnost najveća kada imamo umjeren broj interakcija. Razinu kreativnosti su određivali tako što su članovi timova sami ocjenjivali svoju kreativnost. Jiang i Gero [24] uspoređivali su komunikacijske strukture tijekom razvojne sesije tima od sedam članova s dvije različite metode. Jedna metoda promatra količinu interakcija između članova tima i ne uzima u obzir teme razgovora, a druga razmatra i o kojem konstrukcijskom problemu članovi tima razgovaraju. Pokazali su da obje metode daju slične komunikacijske strukture i da je promatranje govora bez obzira na sadržaj dobra ekonomična metoda za određivanje komunikacijske strukture. Kako analiza sadržaja zahtjeva transkripciju aktivnosti, analizom komunikacijske strukture neovisno o kontekstu možemo značajno smanjiti vrijeme potrebno za analizu (za transkripciju jednog sata potrebno je 10 do 100 sati transkribiranja [25]).

Za provođenje klasične analize protokola, van Someren i sur. [26] su podijelili proceduru na pet koraka: provođenje eksperimenata, transkripcija protokola, proučavanje segmenata, kodiranje prema kodnoj shemi i interpretacija kodiranog protokola. Način na koji se svaki pojedinačni korak provodi ovisi o svrsi istraživanja. Kada se analiziraju konstrukcijske aktivnosti, uključujući i pregled konstrukcija, uobičajeno je promatrati kontekst. Funkcija-ponašanje-struktura (eng. Function-Behaviour-Structure, FBS) se često koristi u svrhu kontekstualne analize konstrukcijskih aktivnosti [27]. FBS ontologija je prikladna za konstrukcijske aktivnosti jer razlaže govor o konstruiranim elementima na samo tri teme: funkciju (što neki element treba raditi), ponašanje (što neki element radi) i strukturu (od čega se neki element sastoji) [27]. S obzirom na to da se i bez uzimanja konteksta u obzir mogu dobiti dobre komunikacijske strukture [24], razvijena računalna podrška neće uzimati kontekst u obzir pa nije potrebno provoditi transkripciju. Segmentiranje protokola se može vršiti na više načina. Kan i Gero [28] predlažu da način segmentiranja konstrukcijskih protokola ovisi o ciljevima i opsegu istraživanja. Jedan od tih načina je segmentiranje prema pojavi nekog procesa kako bi se promatralo koliko često se taj proces pojavljuje. Jedna metoda koja se često koristi za interpretaciju učestalosti pojave tih procesa jest Markovljeva analiza. Osnovna pretpostavka Markovljeve analize je da redoslijed pojave procesa nije slučajan [28]. Primjenom

Markovljeve analize dobivamo normaliziranu učestalost pojave procesa pomoću kojih možemo doći do zaključaka o vjerojatnosti pojave procesa u nekom trenutku.

Kod konstrukcijskih aktivnosti promatrani procesi su često vezani za kontekst. Ovisno o cilju istraživanja segmentacija se može vršiti prema aktivnostima u kojima članovi tima sudjeluju, prema konstrukcijskim problemima o kojima raspravljaju, prema tome pokušavaju li definirati problem ili pronaći rješenje i slično [28]. Ako uzmemo da je proces koji promatramo govor, Ericson i Simon [29] predlažu segmentiranje prema pauzama u govoru, intonaciji, itd. Huet i sur. [2] predlažu da segmentiranje govora pomaže i u razumijevanju važnosti tema prilikom pregleda konstrukcije. Prema indikatorima glasnoće sudionika i trajanju govora vezanog za neku temu moguće je zaključiti koja je tema bila bitnija tijekom pregleda [2]. Postoji i mogućnost segmentiranja s fiksnom duljinom segmenata, ali taj pristup, iako je brži, može davati manje precizne rezultate zbog mogućnosti prekida segmenta tijekom govora i završetka segmenta nakon što govor stvarno završi [28]. Segmentiranje protokola prema govoru i pauzama u govoru se često naziva segmentacija govora.

2.4. Segmentacija govora

Iako neki istraživači koriste slične metode segmentacije, ne postoji jedna metoda koja je primjenjiva na svako istraživanje o pregledu konstrukcija [28]. Kan i Gero [28] su prepoznali da se segmentiranje konstrukcijskih aktivnosti uobičajeno svodi na dvije vrste podataka: kvalitativne i kvantitativne. Vrste kvalitativnih podataka, koji opisuju što su sudionici radili tijekom konstrukcijskih aktivnosti, mogu se značajno razlikovati od istraživanja do istraživanja, ovisno o cilju istraživanja. Kvantitativni podaci, osobito ako se promatra govor sudionika, su često segmentacija govora.

Segmentiranjem govora dobijemo, za danu aktivnost, periode u kojima se govori i u kojima se ne govori te je svaki period govora jedan segment. Kada se analizira aktivnost s više sudionika i stvaramo zasebne segmente za svakog sudionika imamo i informacije o redosljedu govora, to jest kojim redosljedom se izmjenjuju sudionici aktivnosti (eng. turn-taking, TT) [30]. Sacks i sur. [30] su dali nekoliko pretpostavki koje nam pomažu bolje definirati kakve informacije možemo imati u segmentiranom govoru i TT. Njihove pretpostavke su primjenjive na široki spektar različitih tipova razgovora, a ovdje su nabrojane samo neke koje vrijede i za analizu pregleda konstrukcija. Predlažu da se u razgovoru sudionici izmjenjuju, da najčešće priča samo jedan sudionik, ali da su situacije u kojima priča više sudionika česte i kratke. Prijelaz s jednog

sudionika na drugog je najčešće bez pauze i bez preklopa. Redoslijed izmjene sudionika i duljina njihovog govora nije unaprijed definirana. Neke od pretpostavki koje za pregled konstrukcija nisu točne su da trajanje i tema razgovora nije unaprijed definirana. Pregledi konstrukcija često su vremenski ograničeni, a cilj (tema razgovora) pregleda često može biti određen [11]. Kako bi rezultati segmentacije bili konzistentni, možemo pomoću pretpostavki koje su Sacks i sur. [30] naveli definirati nekoliko pravila za segmentirani govor.

Prvo, segment govora jednog sudionika će završiti kada drugi sudionik počne pričati, ili kada prvi sudionik prestane pričati. Dakako, u govoru često postoje kratke pauze te je potrebno definirati koliko treba biti duga pauza između dva segmenta govora istog sudionika da bi se segment podijelio na dva dijela. Barthel i sur. [31] su zaključili da je za planiranje govora potrebno oko jednu sekundu. Prema tome, valja sve segmente jednog sudionika koji između nemaju pauzu veću od jedne sekunde spojiti u jedan.

Nadalje, s obzirom na to da segment završava kada drugi sudionik započne svoj govor, u segmentima nema preklopa. Fredouille i Evans [32] su promatrali utjecaj uklanjanja preklopa govora u automatski segmentiranom govoru snimaka sastanaka. Došli su do zaključka da je kvaliteta segmentacije, s obzirom na poklapanje s ručnom segmentacijom, vrlo konzistentna neovisno o tome uklanjaju li se preklopi ili ne. Također su uočili da je teško zaključiti koliki je utjecaj uklanjanja preklopa na broj segmenata. Za uklanjanje preklopa potrebno je odrediti koji od sudionika imaju veći utjecaj na razgovor dok govore u isto vrijeme.

Razvijeno je nekoliko metoda za određivanje dominantnog sudionika u govoru. Jedan od pristupa koji su ispitali Hung i sur. [33] je korištenje duljine govora za određivanje dominantnih govornika. Oni su uspoređivali metode koje nakon prepoznavanja govora pokušavaju odrediti koji sudionik je dominantan primarno gledajući duljinu govora svakog sudionika. Uočili su da se takav pristup bolje poklapa s ljudskom procjenom ako se koristi samo jedan mikrofonski audio izvor za sve sudionike u odnosu na situaciju kada svaki sudionik ima zaseban mikrofonski izvor. Volfen i Cohen [34] su razvili vlastitu metodu za određivanje dominantnog govornika koja također koristi duljinu govora kao primarnu varijablu. Njihova metoda je namijenjena za situacije kada imamo zasebne mikrofonske izvore za svakog sudionika. Prema trajanju aktivnosti (glasnoća iznad definirane vrijednosti u audio frekvencijama ljudskog govora) na svakom mikrofonskom izvoru odlučuje se koji sudionik je dominantan. Oni su u svojem istraživanju svoju metodu usporedili s nekoliko drugih metoda, uključujući određivanje dominantnog govornika prema glasnoći i postojećim alatima za prepoznavanje govora [35]. Volfen i Cohen [34] su prepoznali

da određivanje dominantnog govornika samo prema glasnoći daje puno krivih odluka o tome koji sudionik je dominantan zbog osjetljivosti na buku u signalu koja nije govor, a metode koje uzimaju u obzir duljinu govora daju bolje rezultate. Otkrili su i da se proširenjem intervala odluke (minimalna duljina zvuka potrebna za promjenu govornika) sa 0,1 na 0,5 sekundi značajno poboljšava rezultate svake ispitane metode.

Segmentacija govora se tradicionalno provodi ručno pomoću alata kao što je ELAN¹. Takvim pristupom korisnik ručno označava početak i/ili kraj govora u video ili audio zapisu te kasnije te podatke može dobiti u obliku vremenskih oznaka (vremena početaka, krajeva i trajanja segmenata). Automatiziranje prepoznavanja govora je već dulje vrijeme područje velikog interesa za istraživače. Razvijene su metode s dobrim uspjehom prepoznavanja govora i segmentiranja tradicionalnim programiranjem [36], a i korištenje metoda umjetne inteligencije (strojno učenje) se pokazalo vrlo uspješnim za prepoznavanje govora [37], [38]. Tradicionalno programiranje podrazumijeva ručno razvijanje pravila i algoritama koji se koriste za prepoznavanje govora. Metodama umjetne inteligencije ti se algoritmi razvijaju tako da algoritam za strojno učenje analizira set podataka. Ti podaci su audio zapisi i pripadajući segmentirani govor. Ovisno o svrsi, može biti uključen i sadržaj govora ako se želi, na primjer, napraviti alat za automatsku transkripciju. Prepoznavanje govora je jedna od najuspješnijih implementacija umjetne inteligencije. Danas prepoznavanje govora umjetnom inteligencijom daje odlične rezultate, a prepoznavanje riječi je iz dana u dan sve pouzdanije [39]. Adlesee i sur. [39] su uočili da i najsofisticiranije umjetne inteligencije za automatizirano prepoznavanje govora (Google Cloud, IBM Watson) još uvijek loše prepoznaju sudionike ako u audio zapisu u isto vrijeme govori više njih. Ručnom segmentacijom korisnik lako može prepoznati i definirati različite sudionike u razgovoru. Prema tome, iako je automatska segmentacija značajno brža [40], ako imamo samo jedan audio zapis za sve sudionike u pregledu konstrukcije ograničeni smo na ručnu segmentaciju. Kako većina HMD VS sustava u sebi ima integriran mikrofoni za snimanje zvuka [15], lako je zasebno snimati sudionike pregleda konstrukcije ako koriste VS.

¹ ELAN - besplatna programska podrška namijenjena kreiranju bilješki na audio i video zapisima. Razvijena je na Max Planck institutu za psiholingvistiku, Nijmegen (Nizozemska). Dostupno na: <https://tla.mpi.nl/tools/tla-tools/elan/>

Temeljem pregleda literature može se pronaći nekoliko zahtjeva za računalnu podršku za praćenje razvojnih timova pri korištenju VS sustava za pregled konstrukcije.

2.5. Zahtjevi za razvoj računalne podrške

Razvijena računalna podrška treba pomoću audio zapisa i segmentiranog govora sesije pregleda konstrukcije dati analizu komunikacijske strukture aktivnosti, bez uključivanja konteksta. Oblik rezultata će biti Markovljeva analiza i vizualizacije govora tijekom pregleda.

Tijekom pregleda literature ustanovljeni su zahtjevi za razvoj računalne podrške:

- Ulazni podaci su zasebni audio zapisi i segmentirani govor svakog sudionika
- Ulazni segmentirani govor ne smije imati pauze u govoru jednog sudionika dulje od jedne sekunda [31]
- Računalna podrška treba ukloniti preklope u segmentiranom govoru, ako oni postoje
- Kriterij za uklanjanje preklopa je da dominantniji sudionik ostaje
- Kriterij dominantnosti sudionika je glasnoća [34]

3. RAČUNALNA PODRŠKA ZA ANALIZU SURADNJE

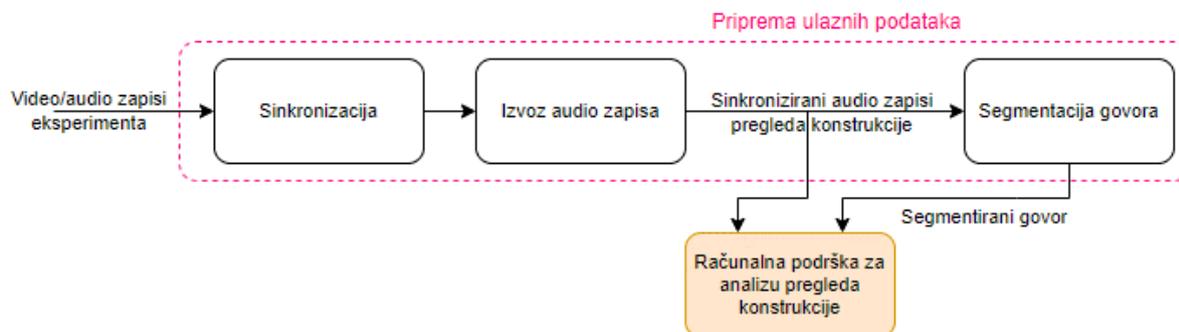
Razvijena računalna podrška je računalni program izrađen u MATLAB-u. MATLAB je računalna platforma i programski jezik namijenjen za znanstvene i inženjerske probleme [41]. Pruža alate za strojno učenje, obradu signala, analizu i vizualizaciju podataka i sl. MATLAB je optimiziran za iterativnu analizu podataka pomoću vlastitog programskog jezika [41].

Kao što je navedeno u poglavlju 2.5, računalna podrška kao ulazne podatke koristi audio zapise i segmentaciju govora jedne sesije pregleda konstrukcije. Za svakog sudionika u pregledu konstrukcije treba biti zaseban audio zapis na kojemu se čuje samo taj sudionik. Također, treba biti segmentacija govora svakog audio zapisa zasebno. Drugo pravilo za ulazni segmentirani govor je da u sebi ne smije sadržavati pauze govora kraće od jedne sekunde. Ispunjenost svih ostalih zahtjeva osigurava razvijena računalna podrška. Ulazni audio zapisi, kao i segmentirani govor, moraju biti sinkronizirani. To znači da svi audio zapisi trebaju početi i završiti u istom trenutku tijekom sesije.

Računalna podrška je namijenjena za analizu komunikacijske strukture razvojnog tima tijekom pregleda konstrukcije. U ovom poglavlju je detaljno opisan redoslijed operacija razvijene računalne podrške, počevši s pripremom ulaznih podataka. Kako bi se bolje opisao rad programa, prikazivat će se primjeri varijabli koje se koriste. Svi korišteni primjeri u ovom poglavlju odgovaraju slučaju koji je opisan u poglavlju 4. Sesija u primjeru traje sat vremena i podijeljena je na dva dijela od kojih su oba trajala 30 minuta. Kako u primjeru imamo četiri sudionika, koristimo 8 parova ulaznih audio zapisa i segmentacija govora – jedan za svakog člana tima za svaku polovicu sesije.

3.1. Priprema ulaznih podataka

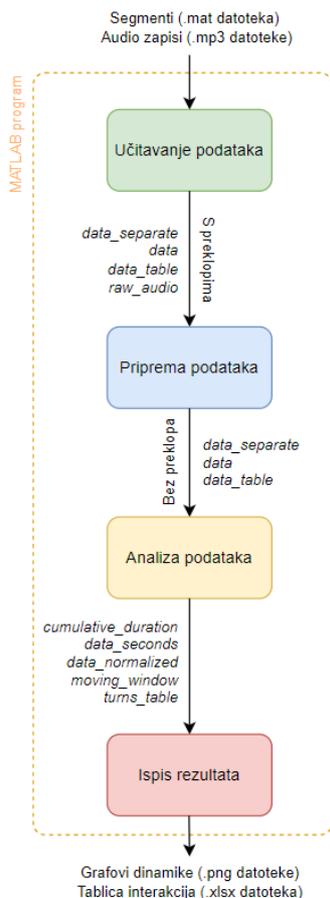
Pretpostavka je da se aktivnost pregleda konstrukcije snima. Mogu se snimati video zapisi sa zvukom ili samo audio zapisi. U oba slučaja, potrebno je prvo sinkronizirati video/audio zapise i odrediti početak i kraj sesije koja će se analizirati. Sinkronizacija je bitna kako bi se segmentacija govora svakog sudionika odnosila na isti vremenski period. Nakon sinkronizacije, potrebno je uzeti samo audio zapise sesije i na njima izvršiti segmentaciju govora [Slika 2]. Segmentaciju je moguće izvršiti ručno ili automatski. S obzirom na to da se koriste zasebni audio zapisi za svakog sudionika preporučuje se automatska segmentacija [39]. Dobiveni audio zapisi i segmentirani govor su ulazni podaci potrebni za računalnu podršku.



Slika 2 Dijagram toka pripreme ulaznih podataka

3.2. Računalni program za analizu aktivnosti pregleda konstrukcije

Razvijena računalna podrška započinje s učitavanjem podataka dobivenih pripremom ulaznih podataka, a zatim slijedi priprema podataka, analiza te ispis rezultata [Slika 3]. Učitavanje podataka podrazumijeva spremanje svih ulaznih podataka unutar programa te oblikovanje informacija kako bi se mogle dalje obrađivati. Tijekom pripreme podataka uklanjaju se svi preklopi u segmentaciji. Nakon toga, tijekom analize podataka, izračunavaju se potrebne vrijednosti. To uključuje provođenje Markovljeve analize, izračun vremena govora sudionika i pokretnih prosjeka govora. Na kraju se ispisuju rezultati dobiveni analizom podataka. Ispis se sastoji od izrade i spremanja grafova ukupnog vremena i pokretnih prosjeka govora kao .png slikovne datoteke te spremanje rezultata Markovljeve analize kao .xlsx tablica.



Slika 3 Dijagram toka programa

3.2.1. Učitavanje podataka

Ulazne podatke (u ovom slučaju 8 audio zapisa i 8 segmentacija govora) učitavamo u MATLAB i spremamo u dvije varijable tipa "cell" [Slika 8]. U jednu varijablu se spremaju audio zapisi, a u drugu segmenti (*data_separate*, Slika 4). Varijabla *data_separate* sa segmentima sadrži početke i krajeve svih segmenata. U navedenim "cell" varijablama imamo osam ćelija, jednu za svaku datoteku, tako da su odvojeni i sudionici i polovice sesija. Nakon toga stvaramo nove "cell" varijable s četiri ćelije gdje će svaka ćelija sadržavati informacije o segmentima kompletne sesije jednog sudionika (*data*, Slika 5). Te informacije su početak, kraj i duljina segmenta. Također stvaramo varijablu u koju spremamo informacije o audio zapisima (*raw_audio*, Slika 6). Ta varijabla sadrži neobrađene (eng. raw) podatke .mp3 datoteka koji su nam potrebni kasnije u programu za određivanje glasnoće pojedinih sudionika. Zbog kasnije analize i jednostavnosti pozivanja podataka, stvaramo i jednu varijablu tipa "table" sa stupcima *Start*, *End*, *Duration*, *Participant* i *Index* (*data_table*, Slika 7). Stupci *Start*, *End* i *Duration* su dobiveni iz "cell" varijable koja sadrži informacije o segmentima (početak, kraj i trajanje), a za

Participant i *Index* stupce su stvoreni vektori koji sadrže informacije o tome na kojeg sudionika se odnosi pojedini segment (*Participant*) i koji je to redni broj segmenta za tog sudionika (*Index*). Varijabla *data_table* se sortira prema stupcu *Start* tako da su podaci kronološki poredani.

```
>> data_separate

data_separate =

    1×8 cell array

    {248×2 double}    {165×2 double}    {212×2 double}    {181×2 double}

>> data_separate{1}

ans =

    1.0e+03 *

    0.0008    0.0017
    0.0037    0.0233
    0.0250    0.0335
    0.0366    0.0386
    0.0558    0.0645
    0.0663    0.0668
    0.0803    0.0833
    0.0897    0.0928
    0.0970    0.0973
    0.1074    0.1130
    0.1223    0.1237
    0.1315    0.1317
    0.1331    0.1359
```

Slika 4 Primjer varijable *data_separate* i jedne ćelije u njoj

```
data =

    1×4 cell array

    {369×3 double}    {319×3 double}    {292×3 double}    {282×3 double}

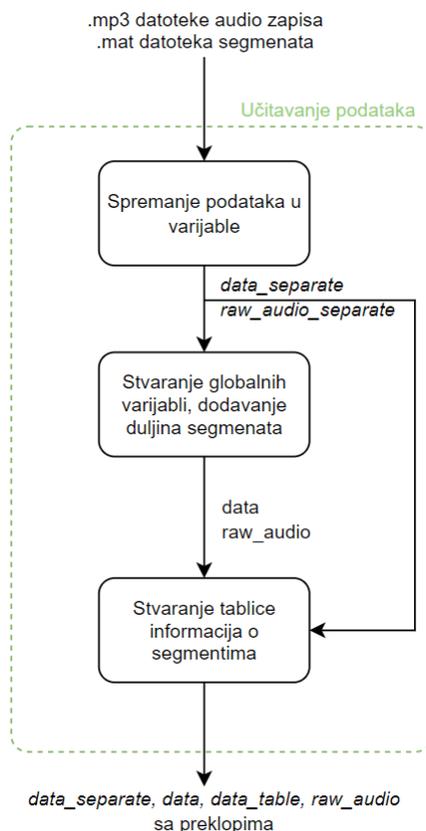
>> data{1}

ans =

    1.0e+03 *

    0.0037    0.0090    0.0053
    0.0095    0.0184    0.0089
    0.0191    0.0233    0.0042
    0.0250    0.0274    0.0024
    0.0276    0.0335    0.0059
    0.0382    0.0386    0.0004
    0.0558    0.0566    0.0008
    0.0577    0.0585    0.0008
    0.0590    0.0645    0.0055
    0.0663    0.0668    0.0005
    0.0803    0.0829    0.0026
    0.0897    0.0924    0.0027
    0.1100    0.1130    0.0029
    0.1387    0.1438    0.0051
```

Slika 5 Primjer varijable *data* i jedne ćelije u njoj



Slika 8 Dijagram toka učitavanja podataka

3.2.2. Priprema podataka

Prije analize, potrebno je iz segmentiranog govora ukloniti preklape. Kada se preklap dogodi, mora se odbaciti preklapljeni dio segmenta onog sudionika koji ima manji utjecaj na razgovor. To se odlučuje tako što se provjerava koji sudionik preklopa je bio glasniji, uz pretpostavku da je glasniji sudionik utjecajniji. Kao što je spomenuto u pregledu literature, određivanje dominantnosti prema glasnoći je najjednostavnija metoda. Prema Volfin i Cohen [34] najveći nedostatak određivanja dominantnosti prema glasnoći je utjecaj šumova. Kako je već provedeno prepoznavanje govora prilikom segmentacije, provjeravat će se glasnoća samo onih dijelova audio zapisa za koje je već poznato da sadrže govor. Zbog toga, šumovi nemaju utjecaj na rezultate.

Prepoznata su dva osnovna tipa preklopa: ugniježđeni preklap i isprepleteni preklap. Kod ugniježđenog preklopa jedan sudionik započne i završi s govorom dok neki drugi sudionik priča. Kod isprepletenog preklopa jedan sudionik započne govor, a zatim drugi sudionik započne svoj govor. Nakon toga prvi govornik završi govor i poslije njega drugi sudionik završi

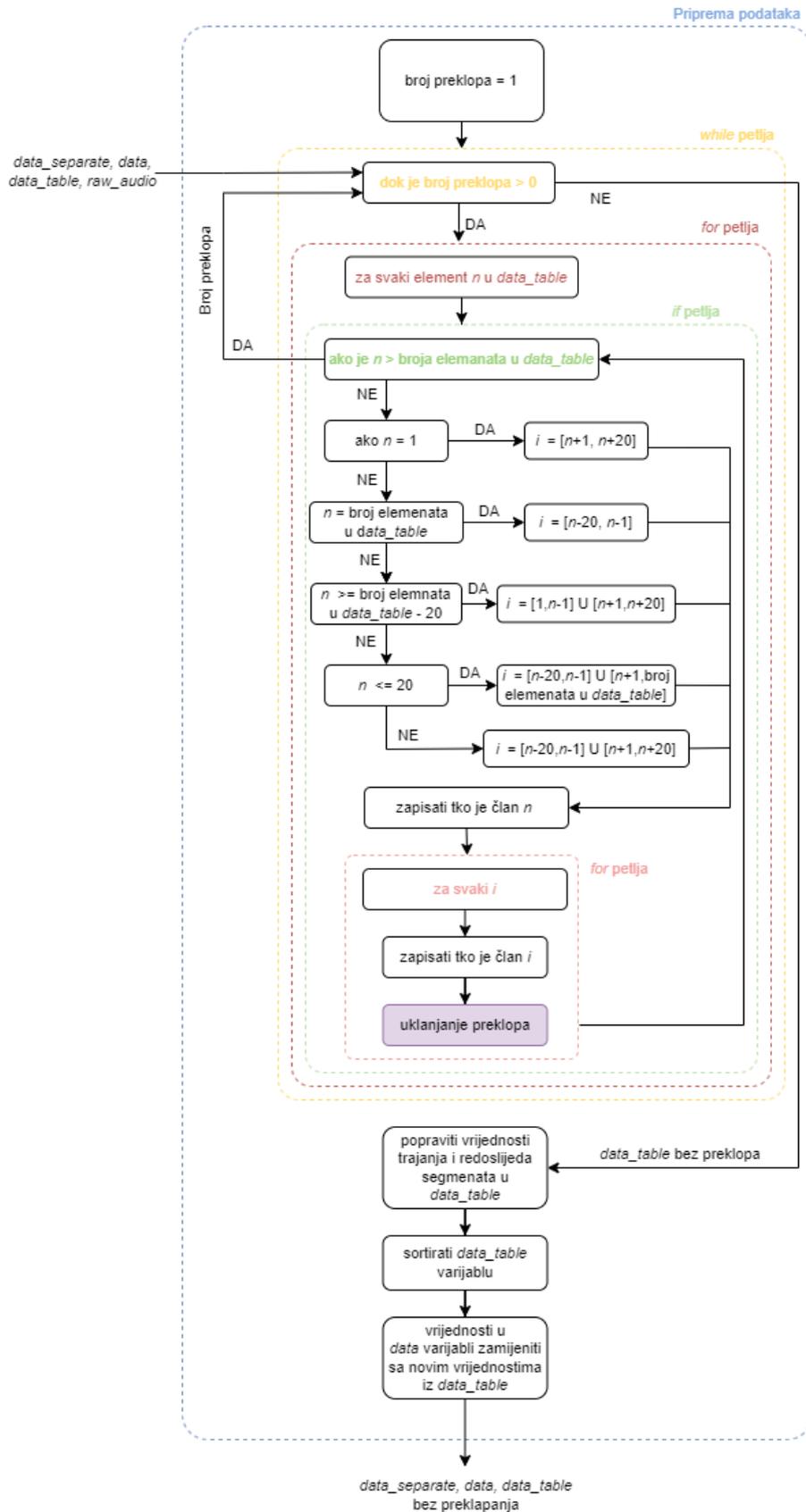
govor. U programu se razlikuje šest slučaja preklopa, četiri ugniježđena i dva isprepletana [Tablica 1].

Tablica 1 Tipovi preklopa

	Tip preklopa	Grafički prikaz
Ugniježđeno	1. Oba segmenta počnu i završe u istom trenutku	
	2. Oba segmenta počnu u istom trenutku	
	3. Oba segmenta završe u istom trenutku	
	4. Oba segmenta počnu i završe u različitim trenucima	
Isprepletano	5. Sudionik n je prvi započeo svoj segmente	
	6. Sudionik i je prvi započeo svoj segment	

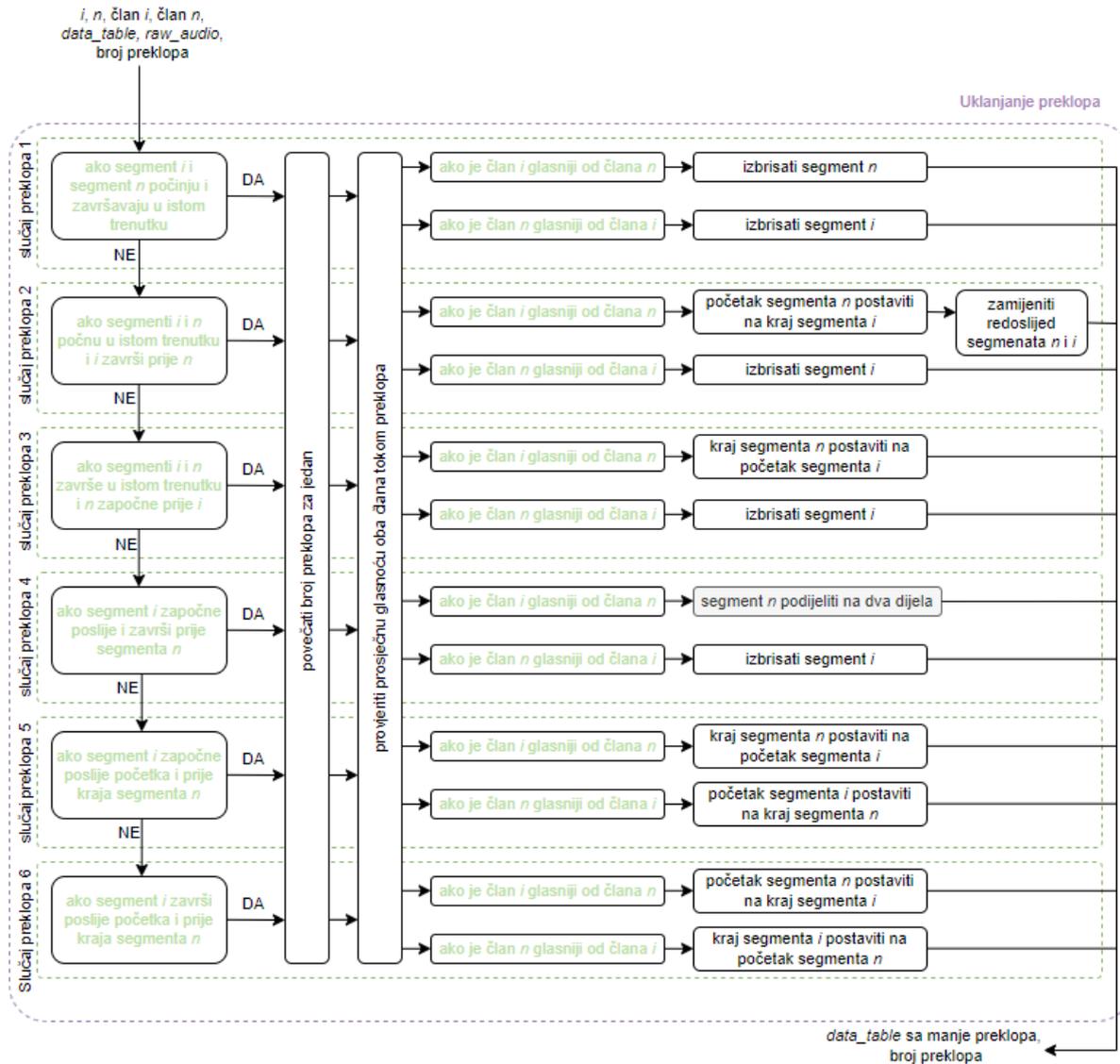
Tih šest podtipova preklopa se razlikuje za potrebe algoritma za uklanjanje preklopa. Svaki od šest podtipova zahtjeva različite aktivnosti za uklanjanje preklopa.

Algoritam za pripremu podataka [Slika 9] započinje definiranjem potrebnih varijabli. Prvo, određuje se raspon u kojem tražimo preklop. Program kasnije za svaki redak varijable *data_table* provjerava postoji li preklop s tim segmentom n (u tablici 1 označeno plavom bojom), pa je potrebno odrediti koliko prije, odnosno poslije, zadanog segmenta treba tražiti preklope. Uočeno je da bi se rasponom od 10 segmenata prije i poslije segmenta n mogli ukloniti svi preklopi u primjeru. Zbog toga, u programu se provjerava 20 segmenata prije i 20 segmenata poslije segmenta n zbog sigurnosti. Segment za kojeg se provjerava preklop sa segmentom n zapisan je varijablom i (u tablici 1 označeno narančastom bojom). Zatim, traži se o kojim se sudionicima radi te se zapisuje koji je sudionik n i koji je sudionik i . Nakon toga se vrši spomenuta provjera postojanja preklopa redosljedom koji je prikazan u tablici 1. Ako se preklop uoči, onda se uklanjaju sljedećim algoritmom [Slika 10].

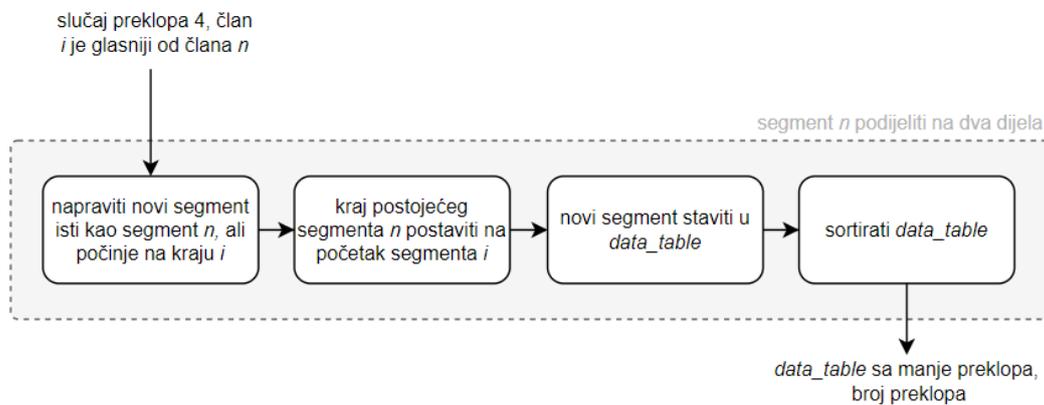


Slika 9 Dijagram toka pripreme podataka

Ako imamo preklop tipa jedan, onda se samo obriše tiši segment. Ako je preklop tipa dva ili tri, ovisno o tome koji sudionik je glasniji, mora se ili obrisati kraći segment (ako je glasniji sudionik n čiji segment traje dulje) ili pomaknuti početak, odnosno kraj, duljeg segmenta (ako je glasniji sudionik i čiji segment traje kraće). Ako je glasniji sudionik i u slučaju dva, potrebno je početak duljeg segmenta postaviti na kraj kraćeg. Ako je glasniji sudionik i u slučaju tri, potrebno je kraj duljeg segmenta postaviti na početak kraćeg. U slučaju četiri, ponovno se samo obriše kraći segment i ako je glasniji sudionik n koji dulje priča. Ako je glasniji sudionik i koji ima kraći segment, onda se mora segment n podijeliti na dva dijela [Slika 11]. To se obavlja tako što se stvara privremena varijabla u koju se zapisuje kraj duljeg segmenta. Nakon toga, za vrijednost kraja duljeg segmenta postavlja se vrijednost početka kraćeg segmenta, kao u slučaju tri. Zatim, stvara se novi segment. Novi segment započinje u trenutku u kojemu završava kraći segment, a završava u trenutku u kojemu je završavao dulji segment (privremena varijabla). Nakon toga, zbog novog segmenta, ponovno se sortira varijabla *data_table*.



Slika 10 Algoritam za uklanjanje preklopa



Slika 11 Dijeljenje segmenta n na dva dijela za slučaj preklopa četiri

Ako se prepozna peti ili šesti slučaj preklopa, mora se pomaknuti početak ili kraj segmenta tišeg sudionika na kraj, odnosno početak, glasnijeg.

Glasnoća se provjerava prema ISO 532-1:2017, tzv. "Zwicker" metodom. ISO 532 su standardi koji definiraju procedure za procjenu glasnoće koju osjete ljudi s normalnim sluhom [42]. Za provjeru glasnoće u traženim intervalima koristi se varijabla s audio zapisima *raw_audio*.

Uzmimo za primjer šesti slučaj preklopa. U tom slučaju, preklop traje od početka segmenta sudionika *n* do kraja segmenta sudionika *i*. Vrijeme početka i kraja preklopa uzima se iz *data_table* varijable. Kako bi program znao koju ćeliju u *raw_audio* varijabli treba, koriste se prethodno napravljene varijable koje povezuju naziv sudionika u *data_table.Participant* stupcu s rednim brojem ćelije u *raw_audio* varijabli. Zatim se koriste informacije o početku i kraju preklopa kako bi se provjerila glasnoća samo u tom dijelu sesije.

Kada se prođe kroz svaki redak *data_table* varijable, provjerava se koliko preklopa je do sada pronađeno. Ako je pronađeno više od nula preklopa, ponovno se prolazi kroz *data_table* uklanjajući preostale preklope. Kada se jednom prođe kroz cijelu tablicu bez ijednog preklopa, tada se izlazi iz petlje i nastavlja dalje s programom. Takav pristup se koristi zbog toga što mogu pričati i više od dva sudionika u isto vrijeme, pa se takvi slučajevi postepeno razriješavaju.

Nakon rješavanja svih preklopa, stupci *Duration* i *Index* više ne odgovaraju stvarnim vrijednostima, kao ni *data* varijabla koja sadrži segmente podijeljene na sudionike. Sljedeći korak je zato ispravljanje tih vrijednosti. Te vrijednosti se jednostavno ispravljaju. *Duration* stupac se prepravi oduzimanjem početka segmenta od kraja. Za ispravljanje *Index* stupca program prolazi kroz *data_table* i provjerava o kojem se sudioniku radi. Program pamti koliko puta do sada se taj sudionik pojavio i mijenja odgovarajuću vrijednost u *Index* stupcu na taj broj. Zatim, kako bi se prepravila *data* varijabla, *data_table* se sortira prema sudionicima pa prema indeksima. Tako imamo tablicu koja je prvotno poredana po sudionicima, a isti sudionici su kronološki poredani. Zatim se samo uzimaju vrijednosti koje trebamo (početak, kraj i trajanje) i mijenjaju se vrijednosti u ćelijama s novim vrijednostima.

3.2.3. Analiza podataka

U ovom trenutku u programu se nalaze svi potrebni podaci te se može započeti s analizom podataka. Cilj je analizirati komunikacijsku strukturu sesije. Kako bi se ispunio taj cilj potrebno je napraviti Markovljevu analizu sesije i dobiti raspodjelu govora tijekom sesije. Program nam

daje dva osnovna tipa vizualizacije raspodjele govora: kumulativno vrijeme govora i pokretni prosjeci (eng. moving average) govora.

Kumulativno vrijeme govora se prikazuje apsolutno, normalizirano s obzirom na trajanje sesije i normalizirano s obzirom na ukupno vrijeme govora pojedinog sudionika. Apsolutni graf kumulativnog vremena govora nam pomaže u razumijevanju udjela govora pojedinog sudionika te na intuitivan način možemo vidjeti koliko su sudionici pričali tijekom sesije. Normalizacijom s obzirom na ukupno trajanje sesije dobivamo mogućnost usporedbe različitih sesija. Normalizacija kumulativnog trajanja govora s obzirom na vrijeme govora pojedinog sudionika omogućuje da vidimo u kojim dijelovima sesije su sudionici više pričali.

Pokretni prosjeci omogućavaju vizualizaciju interakcija između sudionika i dominantnost sudionika u vremenu s obzirom na duljinu govora. Pomoću njih možemo jasnije vidjeti ako je jedan sudionik pričao značajno više od drugih u određenim periodima i sl. Također imamo apsolutni i normalizirani graf kako bi mogli uspoređivati rezultate različitih sesija.

Markovljevom analizom ćemo dobiti vrijednosti Markovljevih lanaca nultog, prvog i drugog reda. To je metoda koja služi za predviđanje pojave nekih događaja, a dobivene vrijednosti su vjerojatnosti pojave događaja. Markovljevom analizom se može procijeniti kolika je vjerojatnost da će u proizvoljnom trenutku govoriti neki sudionik (nulti red). Moći ćemo i zaključiti kolika je vjerojatnost da neka dva sudionika imaju međusobnu interakciju, ili da jedan sudionik ima dva segmenta za redom (prvi red). Također, vrijednosti Markovljevog lanca drugog reda govore o vjerojatnosti da se dogodi neki slijed događaja koji uzima tri segmenta u obzir. To može biti slučaj gdje jedan sudionik govori tri puta za redom ili gdje se nekim redoslijedom izmjenjuju dva ili tri sudionika.

Dijagram toka analize podataka je prikazan na Slika 14. Prvo, stvara se nova varijabla tipa "cell" kumulativnog trajanja govora za svakog sudionika kako bi imali vrijednosti potrebne za taj graf. Svaka ćelija u toj varijabli sadrži vektor čije vrijednosti odgovaraju dosadašnjem ukupnom trajanju govora tog sudionika.

Zatim, potrebno je preoblikovati podatke tako da se može izraditi graf pokretnog prosjeka govora za svakog sudionika. Stvara se nova varijabla tipa matrica (*data_seconds*) s četiri stupca (jedan stupac za jednog sudionika) i 4.000 redaka (jedan redak za svaku sekundu u sesiji s dodatkom za sigurnost). Jedan sat traje 3.600 sekundi, a u slučaju da sesija traje dulje dodajemo 400 sekundi za sigurnost. U toj varijabli će se zapisivati koliko dugo sudionici pričaju u svakoj

sekundi. Za svakog sudionika i svaki segment provjerava se počinje li sudionik govoriti u toj sekundi. Za segment koji počinje u danoj sekundi, prvo se provjeri završava li segment u toj istoj sekundi. Ako segment završava u istoj sekundi, onda se odgovarajućem retku i stupcu *data_seconds* matrice dodaje vrijednost trajanja tog segmenta. Ako segment ne završi u istoj sekundi, onda se prvo u odgovarajućoj ćeliji u *data_seconds* dodaje koliko dugo sudionik priča u sekundi u kojoj je započeo govor. Zatim, svakoj sljedećoj ćeliji u *data_seconds* varijabli se dodaje vrijednost jedan sve dok se ne dođe do sekunde u kojoj govor završava. Odgovarajućoj ćeliji u kojoj segment završi pridodaje se preostali dio trajanja segmenta.

```

data_seconds =
      0      0      0      0
      0      0      0      0
      0      0      0      0
      0      0      0      0
      0      0      0      0
      0      0      0      0
      0      0      0      0
      0      0      0      0
0.5000      0      0      0
1.0000      0      0      0
1.0000      0      0      0
1.0000      0      0      0
1.0000      0      0      0
1.0000      0      0      0
1.0000      0      0      0
1.0000      0      0      0
1.0000      0      0      0
0.3800      0      0.6200      0
0.9300      0      0.0700      0
1.0000      0      0      0
1.0000      0      0      0
1.0000      0      0      0
0.2900      0.7100      0      0
0.0200      0.9800      0      0
1.0000      0      0      0
1.0000      0      0      0
0.7700      0      0.2300      0
1.0000      0      0      0
1.0000      0      0      0
1.0000      0      0      0
1.0000      0      0      0

```

Slika 12 Primjer varijable *data_seconds*

Nakon toga, da bi se dobili normalizirani pokretni prosjeci, stvara se nova matrica s četiri stupca (jedan za svakog sudionika) i 1.000 redaka (kako bi imali korake govora od 0,1%). Kako bi se dobile vrijednosti za tu matricu, prvo se izračunava koliko traje 0,1% sesije te se tolikim koracima kreće kroz *data_seconds* i zapisuje koliki dio koraka sudionik govori u danih 0,1% sesije.

Osim trajanja i dinamike govora, vrši se i analiza interakcija između sudionika, a za to je potrebno prebrojati koliko interakcija postoji između svakog sudionika. U programu se vrši

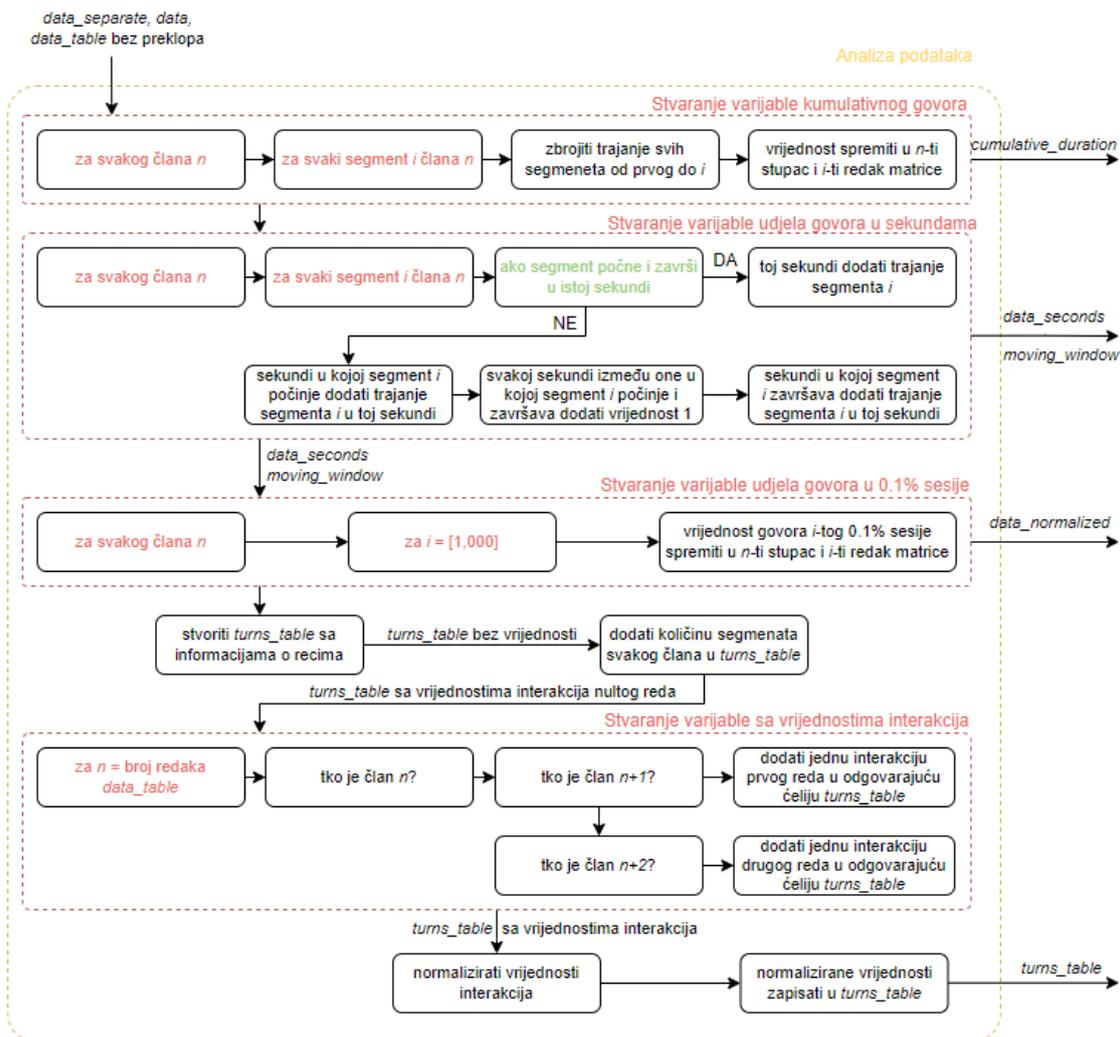
analiza nultog, prvog i drugog reda interakcije. To znači da se za svaki segment mora pogledati tko trenutno priča (nulti red), tko priča poslije njega (prvi red) te koji sudionik je dva segmenta poslije (drugi red). Za spremanje tih podataka stvara se nova varijabla tipa "table" (*turns_table*, Slika 13) s pet stupaca: *Values*, *Markov*, *Type*, *Participant_order* i *Participant_type_order*. U stupac *Type* zapisuje se red interakcije (nulti, prvi ili drugi), u *Participant_order* zapisuje se redosljed sudionika, a u *Participant_type_order* se također zapisuje redosljed sudionika, ali samo tip sudionika, odnosno radi li se o konstruktoru ili recenzentu. Stupac *Values* sadrži informacije o tome koliko puta se dogodila dana interakcija. Da bi dobili vrijednosti za taj stupac, potrebno je ponovno proći kroz *data_table* varijablu i za svaki redak provjeriti tko priča u iduća dva segmenta. Zatim, potrebno je u odgovarajuće retke *turns_table* varijable zapisati da se dogodila ta interakcija. Stupac *Markov* sadrži normalizirane vrijednosti interakcija koje odgovaraju vrijednostima Markovljevih lanaca.

```
turns_table =
```

```
84×5 table
```

Values	Markov	Type	Participant_order	Participant_type_order
369	0.29239	0	"D1"	"D"
319	0.25277	0	"D2"	"D"
292	0.23138	0	"R1"	"R"
282	0.22345	0	"R2"	"R"
24	0.019033	1	"D1-D1"	"DD"
128	0.10151	1	"D1-D2"	"DD"
118	0.093577	1	"D1-R1"	"DR"
99	0.078509	1	"D1-R2"	"DR"
134	0.10626	1	"D2-D1"	"DD"
21	0.016653	1	"D2-D2"	"DD"
80	0.063442	1	"D2-R1"	"DR"
83	0.065821	1	"D2-R2"	"DR"
93	0.073751	1	"R1-D1"	"RD"
89	0.070579	1	"R1-D2"	"RD"
22	0.017446	1	"R1-R1"	"RR"
88	0.069786	1	"R1-R2"	"RR"
118	0.093577	1	"R2-D1"	"RD"
80	0.063442	1	"R2-D2"	"RD"
72	0.057098	1	"R2-R1"	"RR"
12	0.0095163	1	"R2-R2"	"RR"
2	0.0015873	2	"D1-D1-D1"	"DDD"
7	0.0055556	2	"D1-D1-D2"	"DDD"
7	0.0055556	2	"D1-D1-R1"	"DDR"
8	0.0063492	2	"D1-D1-R2"	"DDR"

Slika 13 Primjer varijable "turns_table"

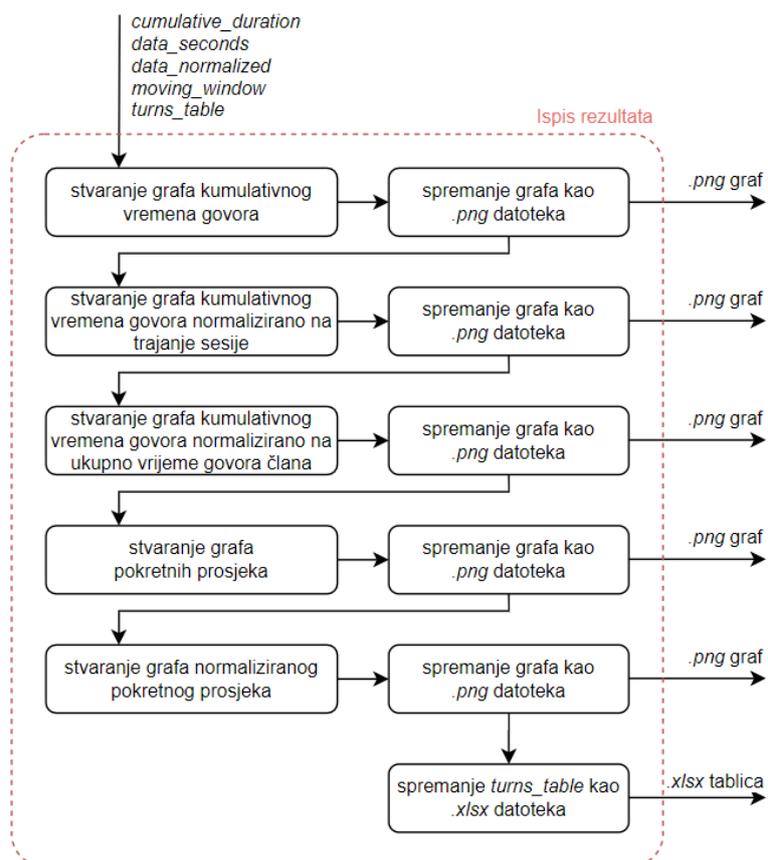


Slika 14 Dijagram toka analize podataka

3.2.4. Ispis rezultata

Na kraju programa ispisuju se rezultati analize [Slika 15]. Stvara se pet grafova: graf kumulativnog vremena govora, graf kumulativnog vremena govora koji je normaliziran s obzirom na trajanje sesije, graf kumulativnog vremena govora koji je normaliziran s obzirom na ukupno vrijeme govora svakog sudionika, graf pokretnih prosjeka i graf normaliziranih pokretnih prosjeka. Svaki graf se sprema kao slikovna ".png" datoteka.

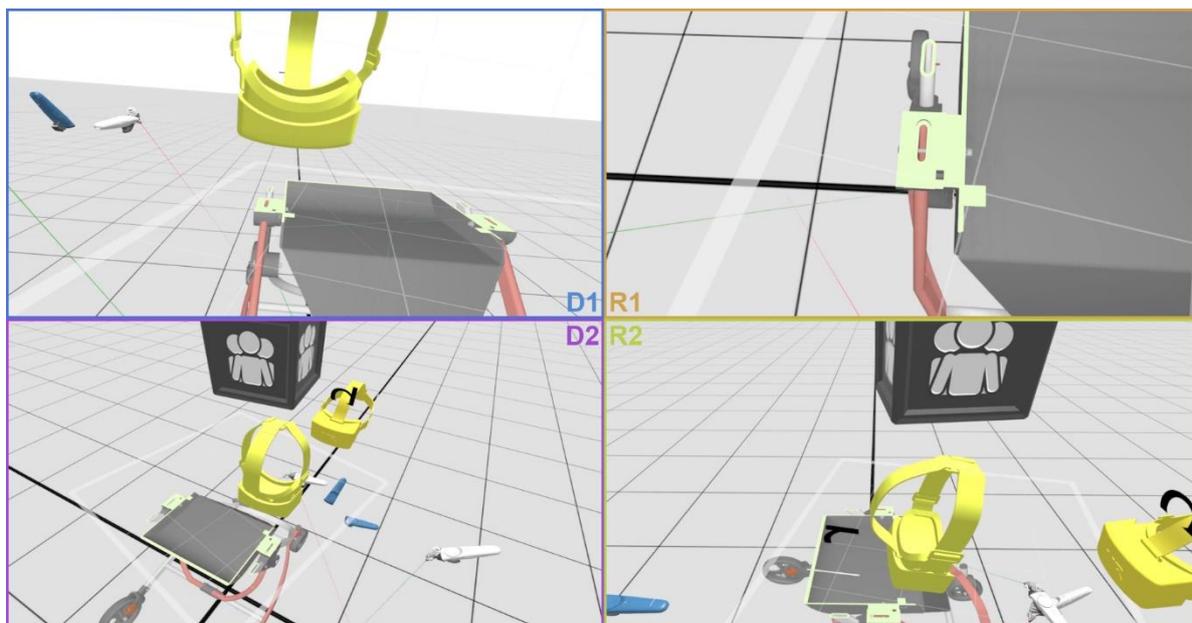
Osim grafova, *turns_table* varijabla, koja sadrži informacije o interakcijama tijekom sesije, sprema se kao Excel tablica, odnosno ".xlsx" datoteka.



Slika 15 Dijagram toka ispisa rezultata

4. PRIMJENA PROGRAMA NA PRIMJERU

U ovom poglavlju će se pokazati rad računalne podrške na primjeru jedne sesije pregleda konstrukcije u VS-u. Tim u primjeru se sastoji od četiri člana, dva konstruktora i dva recenzenta. Konstruktori su osobe koje su sudjelovali u razvoju dječjih kolica koja se pregledavaju. To su studenti preddiplomskog studija strojarstva na Fakultetu strojarstva i brodogradnje. Recenzenti se prvi put upoznaju s konstrukcijom. To su osobe koje rade kao inženjeri strojarstva u lokalnim tvrtkama. Svi sudionici su bili u zasebnim prostorijama. Pregled konstrukcije se provodio online u Siemens NX VS okruženju, a komunicirali su putem Microsoft Teams sastanka. Tijekom pregleda su se zasebno snimali video zapisi za svakog sudionika na kojima se vidi pogled u VS-u i čuje govor samo tog člana tima [Slika 16]. Tijekom pregleda, sudionici su se fokusirali na nedostatke postojeće konstrukcije i pokušavali su pronaći potencijalna unaprjeđenja. Takav pregled je, prema Dhillon-u [10], srednjeg tipa.



Slika 16 Pogledi korisnika u VS-u u jednom trenutku pregleda konstrukcije

Prvo je detaljno opisana priprema podataka primjera, a zatim i rad računalne podrške. Nakon toga, opisani su rezultati dobiveni analizom primjera.

4.1. Priprema ulaznih podataka

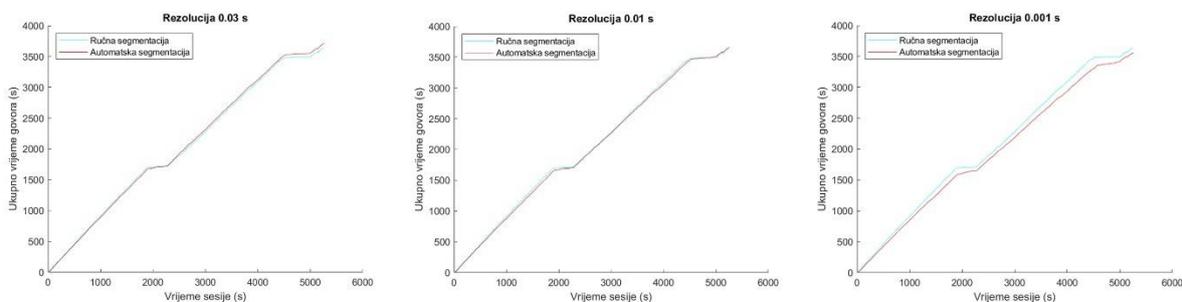
Priprema ulaznih podataka je provedena procedurom koja je prikazana na Slika 2. Prije svega, potrebno je sinkronizirati video zapise pregleda konstrukcije. Sinkronizacija je provedena u Adobe Premiere Pro, programu za video montažu koji je dio Adobe Creative Cloud platforme.

Nakon učitavanja video zapisa u program, potrebno je pronaći zajedničku sinkronizacijsku točku u zapisu svakog sudionika prema kojoj će se vršiti sinkronizacija. Na početku video zapisa određena je vizualna (odbrojavanje na dijeljenom zaslonu unutar Microsoft Teams sastanka) i auditorna (pljeskanje istraživača) sinkronizacijska točka. U Adobe Premiere Pro potrebno je video zapise postaviti tako da su im sinkronizacijske točke u istom položaju na vremenskoj liniji (eng. timeline) programa. Tako se osigurava sinkronizacija video zapisa. Sinkronizacijska procedura se događa na početku video zapisa, a pregled konstrukcije započne kasnije. S obzirom na to da nas zanima samo komunikacija sudionika tijekom pregleda konstrukcije, potrebno je pronaći početak i kraj pregleda unutar video zapisa. Za to se oslanjamo na promatranje aktivnosti sudionika, npr. pozdravljanje istraživača ili neke druge znakove da je pregled konstrukcije započeo, odnosno završio. Nakon označavanja početka i kraja pregleda konstrukcije na sinkroniziranim video zapisima, koristeći Adobe Premiere Pro iz video zapisa izvozimo samo audio zapise. To činimo tako da audio zapis svakog sudionika spremamo u zasebnu .mp3 datoteku.

Sada kada imamo audio zapise na kojima se čuje samo jedan sudionik, potrebno je provesti segmentaciju govora. Kao što je spomenuto u pregledu literature, to je moguće učiniti ručno ili automatski. Na ovom primjeru će se vršiti automatska segmentacija. Za provođenje automatske segmentacije potrebno je definirati dva parametra: minimalno trajanje tišine i rezolucija segmentacije. Minimalno trajanje tišine se odnosi na to kolika mora biti minimalna pauza u govoru između dva segmenta da se ti segmenti ne spoje u jedan. Minimalna tišina je određena prema istraživanju Barthel i sur. [31] kao jedna sekunda. Rezolucija segmentacije se odnosi na korak segmentacije. Automatskom segmentacijom se uzima jedan dio, odnosno korak, audio zapisa i provjerava se da li se u tom koraku nalazi govor. Ako se nalazi govor, algoritam za automatsku segmentaciju označava taj dio kao segment govora i nastavlja na sljedeći dio, a ako nema govora onda samo prijeđe na idući dio audio zapisa. Nakon što označi svaki korak audio zapisa u kojemu se nalazi govor, sve segmente koji su udaljeni manje od definirane minimalne tišine (jedna sekunda) spaja u jedan. Rezolucija, odnosno korak segmentacije, određen je eksperimentalno kao 0.01 sekunda.

Na jednom primjeru sesije pregleda konstrukcije izvršena je prvo ručna, a zatim automatska segmentacija s različitim rezolucijama segmentacije. Ručna segmentacija koja je korištena za utvrđivanje uspješnosti automatske je izvršena pomoću ELAN-a.

U grafovima [Slika 17] vidimo podudaranje kumulativnog trajanja govora ručne i automatske segmentacije s različitim rezolucijama. Za prvi pokušaj automatske segmentacije postavljena je rezolucija od 0,03 sekunde jer je tako postavljen parametar prilikom pokretanja segmentacije. Zatim se, zbog veće preciznosti, rezolucija smanjila na 0,001 sekundu.



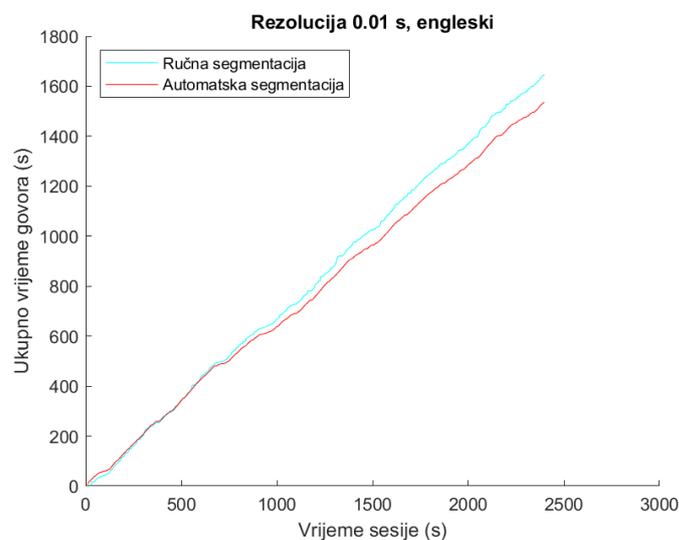
Slika 17 Usporedba rezolucija automatske segmentacije

S rezolucijom od 0,03 sekunde ukupno trajanje govora dobiveno automatskom segmentacijom je 1,9% dulje od trajanja dobivenog ručnom segmentacijom. S rezolucijom od 0,001 sekunde ručna segmentacija ima dulje ukupno trajanje govora od automatske za 2,3%. Ako se ipak uzme rezolucija od 0,01 sekunde, ukupna duljina govora dobivena automatskom segmentacijom je samo 0,6% dulja od one dobivene ručnom segmentacijom. To nam govori da rezolucija od 0,01 sekunde daje segmente koji su vrlo slični ručnoj segmentaciji. Kod ručne segmentacije korisnik može segmentirati samo onom rezolucijom kojom su snimani video zapisi. To znači da za video sa 60 sličica u sekundi ručna segmentacija ima rezoluciju od 0,0167 sekundi. Prema tome, ima smisla da automatska segmentacija daje slične rezultate kao i ručna segmentacija kada se približava rezoluciji ručne segmentacije.

Kako bi se verificirala rezolucija od 0,01 s, provedena je automatska segmentacija na još jednom primjeru [Slika 18]. U tom primjeru se također radi o pregledu konstrukcije, ali na engleskom jeziku i druga je osoba vršila ručnu segmentaciju. Za rezoluciju od 0,01 sekunde, ručna segmentacija je dulja od automatske za 7,4%. Usporedbom segmenata dobivenih ručnom i automatskom segmentacijom uočeno je nekoliko stvari. Uočeno je da je ručna segmentacija primjera s engleskim jezikom bila tolerantnija prema pauzama, to jest postoji puno slučajeva gdje korisnik nije prekinuo segment iako je pauza dulja od jedne sekunde (Slika 19, zelena oznaka). To djelomice objašnjava zašto je razlika u trajanju govora između ručne i automatske segmentacije tako velika u tom primjeru.

U oba primjera je tijekom usporedbe segmenata uočeno da je ručna segmentacija tolerantnija prema šumovima u govoru (npr. riječi koje završavaju na s), odnosno automatska segmentacija

prekine segment ranije u takvim situacijama (Slika 19, narančaste oznake). To djelomice objašnjava zašto je ručna segmentacija dulja u oba slučaja.



Slika 18 Usporedba kumulativnog trajanja govora ručne i automatske segmentacije na engleskom jeziku

Na Slika 19 vidimo razliku između segmenata ručne i automatske segmentacije na primjeru s engleskim jezikom. Na toj slici su označene i razlike između ručne i automatske segmentacije. Segmenti u gornjem redu su dobiveni ručnom segmentacijom, a donji segmenti automatskom segmentacijom. Valja napomenuti da se spajanje segmenata ručne segmentacije koji su razmaknuti manje od jedne sekunde vršilo prilikom usporedbe trajanja govora te da na Slika 19 takvi segmenti nisu prikazani spojeno (plave oznake).



Slika 19 Usporedba segmentacije za primjer na engleskom jeziku

4.2. Rad računalne podrške na primjeru

Analiza podataka pomoću razvijene računalne podrške provedena je na Lenovo Legion Y530 laptopu (Intel® Core™ i5-8300H @ 2,3 GHz procesor i 16 GB DDR4 @ 2667 MHz RAM). Tijekom rada programa procesor je većinu vremena radio na 3,9 GHz sa snagom od 45 W. Ukupno vrijeme potrebno za analizu sesije pomoću računalne podrške je 5,2 minute. Najviše vremena je utrošeno na učitavanje audio zapisa i uspoređivanje glasnoća audio zapisa prilikom

uklanjanja preklopa. Vrijeme potrebno za uspoređivanje glasnoća odgovara 56% ukupnog vremena potrebnog za rad programa, a učitavanje audio zapisa 22%. Kako je jedini razlog za učitavanje audio zapisa uklanjanje preklopa, vrijeme rada programa moglo bi se značajno smanjiti kada se ne bi vršilo uklanjanje preklopa. Kako bi se izbjeglo uklanjanje preklopa, bilo bi potrebno provoditi ručnu segmentaciju audio zapisa. Provođenje ručne segmentacije za aktivnost koja traje jedan sat može trajati i do 10 sati [40]. Prema tome, ovaj pristup, koji koristi automatsku segmentaciju i uklanjanje preklopa, je svakako puno brži od pripreme ulaznih podataka ručnom segmentacijom.

Računalni program je prilikom pripreme podataka uklonio 1104 slučaja preklopa, a ukupni broj segmenata se smanjio za 16%. Usporedbom segmenata u *data_table* varijabli prije i poslije uklanjanja preklopa [Slika 20] uočeno je da većina uklonjenih segmenata odgovara vrlo kratkim izričajima poput "mhm", "da", "aha" i slično (označeno na Slika 20). Takvi izričaji služe kako bi sudionici signalizirali da obraćaju pažnju na dominantnog sudionika i nemaju veliki utjecaj na tijek razgovora [43].

data_table =					data_table =				
1503×5 table					1262×5 table				
Start	End	Duration	Participant	Index	Start	End	Duration	Participant	Index
0.42	3.07	2.65	"D2"	1	0.42	3.07	2.65	"D2"	1
0.61	0.67	0.06	"R2"	1	3.71	9.01	5.3	"D1"	1
0.85	1.69	0.84	"D1"	1	9.01	9.5	0.49	"R1"	1
3.71	23.33	19.62	"D1"	2	9.5	18.38	8.88	"D1"	2
9.01	9.5	0.49	"R1"	1	18.38	19.07	0.69	"R1"	2
15.89	16.12	0.23	"R1"	2	19.07	23.29	4.22	"D1"	3
18.38	19.07	0.69	"R1"	3	23.29	24.98	1.69	"D2"	2
21.23	21.57	0.34	"R1"	4	24.98	27.41	2.43	"D1"	4
23.29	25.69	2.4	"D2"	2	27.41	27.64	0.23	"R1"	3
23.34	26.11	2.77	"R2"	2	27.64	33.49	5.85	"D1"	5
23.63	23.71	0.08	"R1"	5	33.49	35.36	1.87	"D2"	3
24.85	25.32	0.47	"R1"	6	36.29	36.39	0.1	"R2"	1
24.98	33.49	8.51	"D1"	3	36.39	38.23	1.84	"R1"	4
27.41	27.64	0.23	"R1"	7	38.23	38.62	0.39	"D1"	6
33.16	35.36	2.2	"D2"	3	38.62	40.49	1.87	"R2"	2
34.69	34.8	0.11	"R2"	3	40.49	40.68	0.19	"R1"	5
36.29	47.25	10.96	"R2"	4	40.68	47.25	6.57	"R2"	3
36.39	38.23	1.84	"R1"	8	48.4	53.78	5.38	"R2"	4
36.6	38.62	2.02	"D1"	4					

Slika 20 Usporedba *data_table* prije (lijevo) i poslije (desno) uklanjanja preklopa

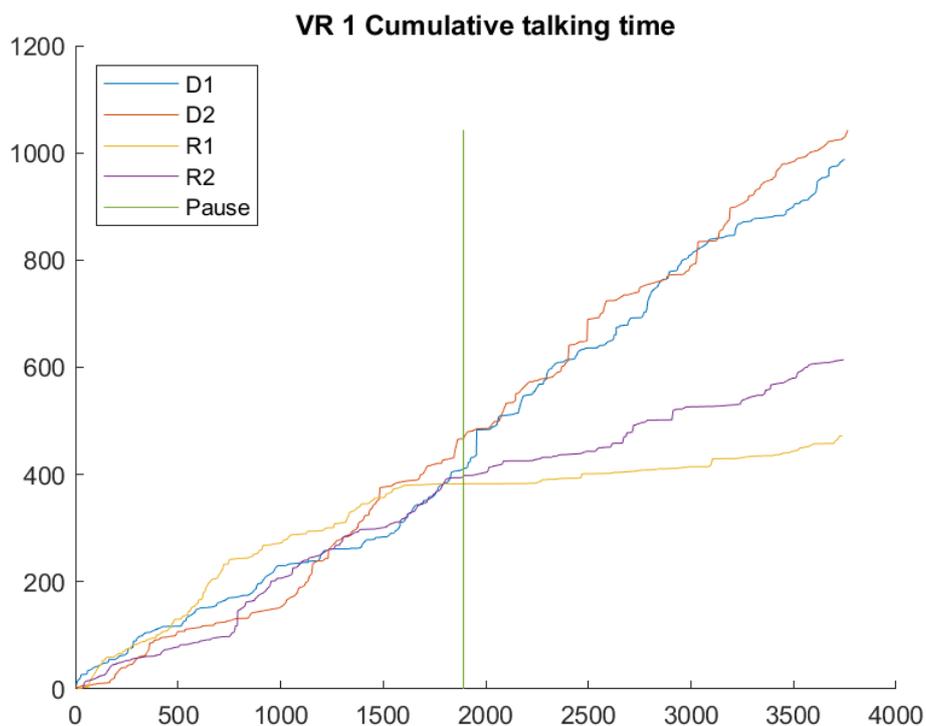
4.3. Rezultati analize primjera

U rezultatima su recenzenti naznačeni s "R1" i "R2", a konstruktori s "D1" i "D2" (eng. designer). U grafovima kumulativnog trajanja govora [Slika 21, Slika 22, Slika 23] možemo uočiti da oba konstruktora ujednačeno sudjeluju u razgovoru tijekom cijelog pregleda konstrukcije. Također, vidi se da recenzenti u drugoj polovici sesije manje sudjeluju u

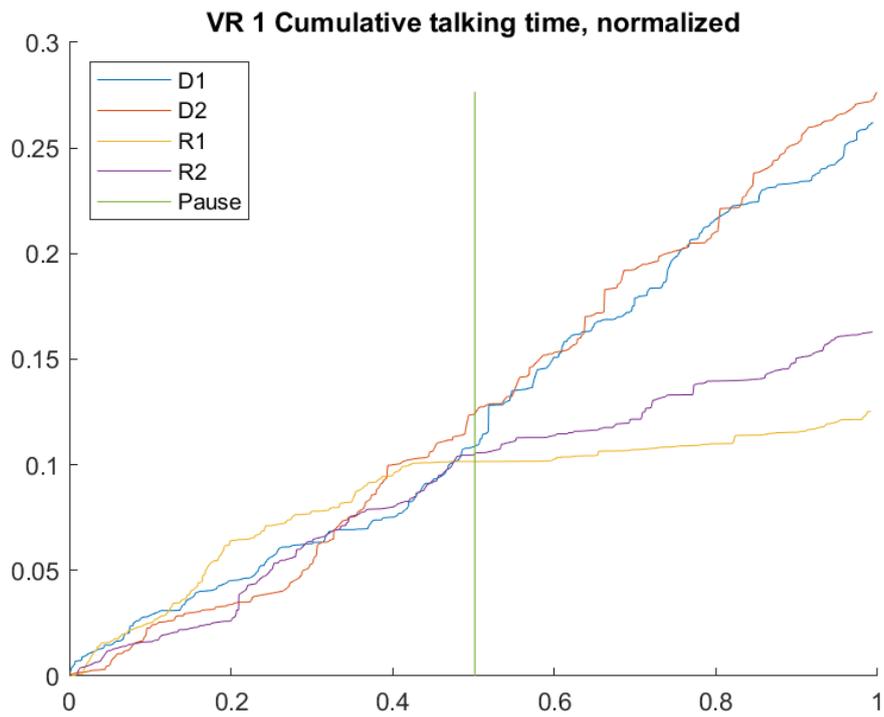
razgovoru, posebice recenzent "R1". Ukupno vrijeme trajanje govora svih sudionika nalazi se u Tablica 2.

Tablica 2 Ukupno vrijeme trajanja govora sudionika

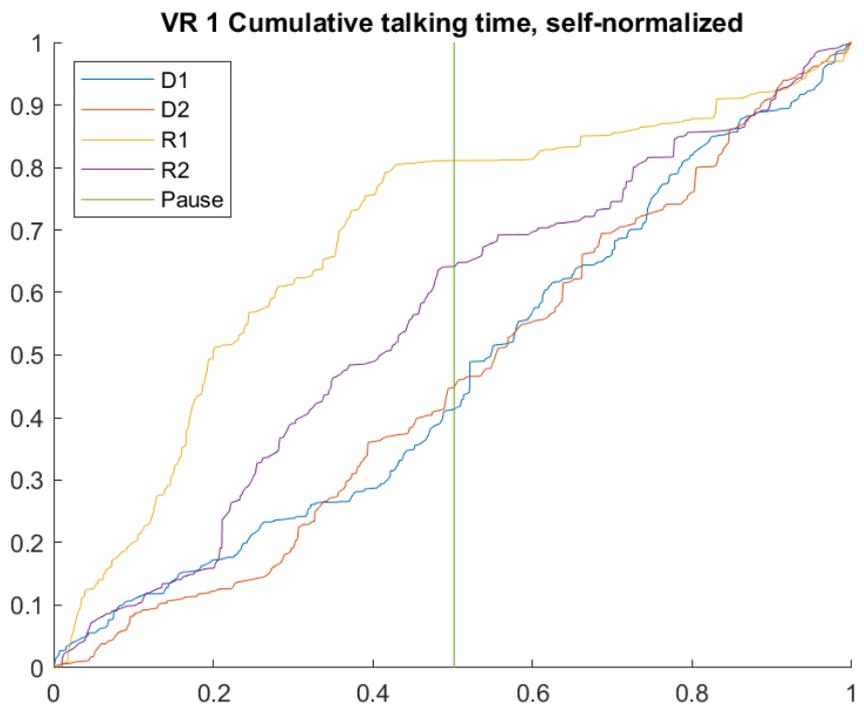
SUDIONIK	UKUPNO VRIJEME TRAJANJA GOVORA (s)
D1	987,7
D2	1042,6
R1	472,1
R2	614,7



Slika 21 Graf kumulativnog vremena govora

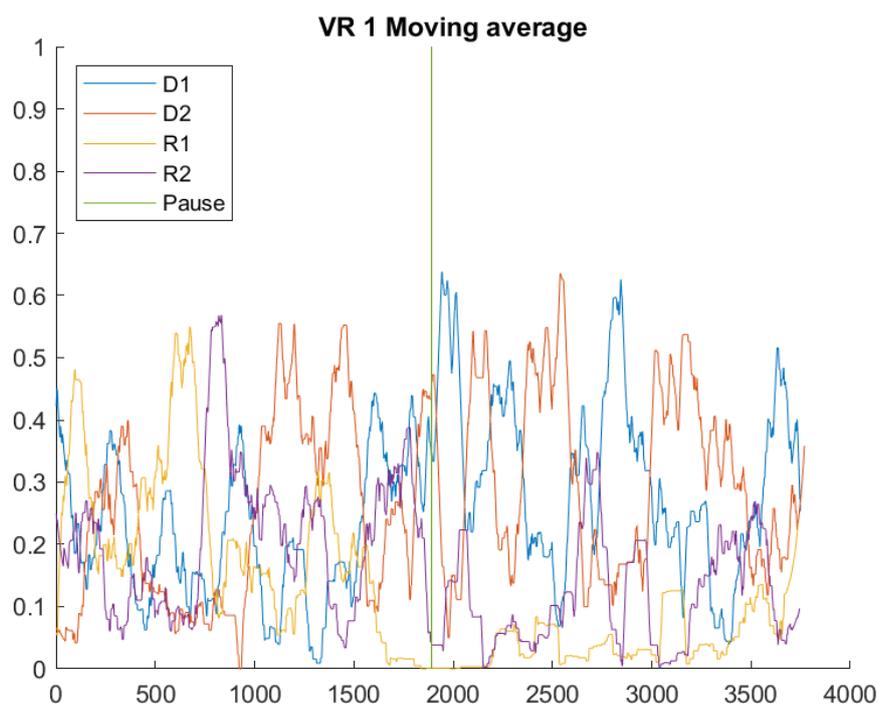


Slika 22 Graf kumulativnog vremena govora normaliziran s obzirom na ukupno trajanje sesije

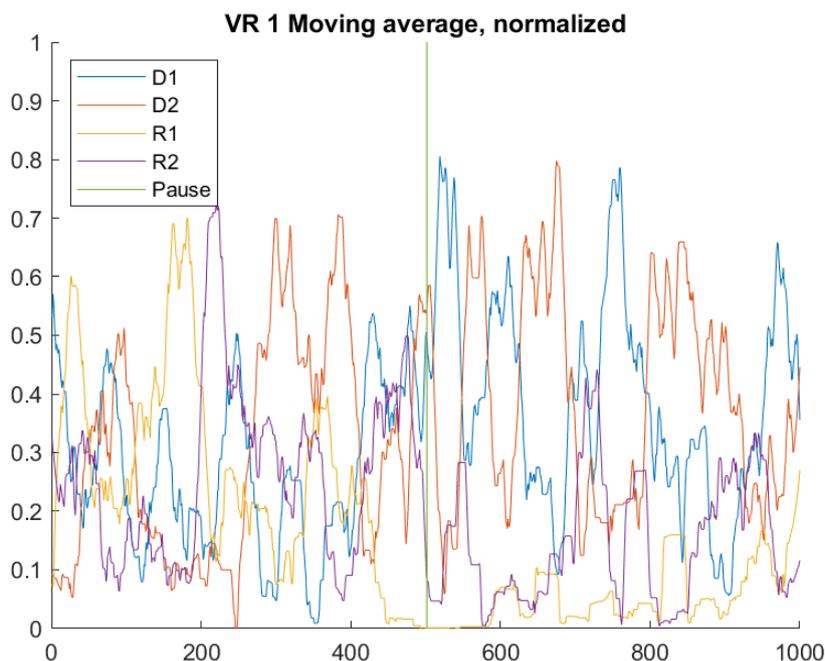


Slika 23 Graf kumulativnog vremena govora normaliziran s obzirom na ukupno trajanje govora svakog sudionika zasebno

U grafovima pokretnih prosjeka [Slika 24, Slika 25], jasnije se vide interakcije između sudionika. Zanimljivo je da često u razgovoru sudjeluje samo jedan recenzent i samo jedan konstruktor. To možemo primijetiti i u tablici interakcija [Tablica 3]. U stupcu "Markov" vidimo da sve interakcije prvog reda u kojima se nalazi samo jedan tip sudionika ("DD", "RR") označavaju 39% svih interakcija.



Slika 24 Graf pokretnih prosjeka



Slika 25 Graf normaliziranih pokretnih prosjeka

Zanimljivo je da se razlike između poklapanja segmenata i ukupnog trajanja govora ne poklapaju. Iz tablice interakcija članova tima [Tablica 3] možemo vidjeti da je najveća razlika količine segmenata između sudionika D1 i R2 ($D1 = 369$, $R2 = 282$ segmenta) i ona iznosi 23,6%. Razlika između ukupnog govora D1 i R2 [Tablica 2] je 37,8%. To znači da sudionik D1 ima više segmenata i dulje segmente od R2.

Analizom rezultata Markovljeve analize drugog reda [Tablica 3] možemo uočiti da interakcije tipa "RDR" i "DRD" odgovaraju 42% svih interakcija drugog reda, iako je to samo četvrtina svih tipova interakcija drugog reda.

Tablica 3 Tablica interakcija članova tima

Values	Markov	Type	Participant order	Participant type order
369	0.292393027	0	D1	D
319	0.252773376	0	D2	D
292	0.231378764	0	R1	R
282	0.223454834	0	R2	R
24	0.019032514	1	D1-D1	DD
128	0.101506741	1	D1-D2	DD
118	0.093576527	1	D1-R1	DR
99	0.07850912	1	D1-R2	DR
134	0.106264869	1	D2-D1	DD
21	0.01665345	1	D2-D2	DD
80	0.063441713	1	D2-R1	DR
83	0.065820777	1	D2-R2	DR
93	0.073750991	1	R1-D1	RD
89	0.070578906	1	R1-D2	RD
22	0.017446471	1	R1-R1	RR
88	0.069785884	1	R1-R2	RR
118	0.093576527	1	R2-D1	RD
80	0.063441713	1	R2-D2	RD
72	0.057097542	1	R2-R1	RR
12	0.009516257	1	R2-R2	RR
2	0.001587302	2	D1-D1-D1	DDD
7	0.005555556	2	D1-D1-D2	DDD
7	0.005555556	2	D1-D1-R1	DDR
8	0.006349206	2	D1-D1-R2	DDR
76	0.06031746	2	D1-D2-D1	DDD
6	0.004761905	2	D1-D2-D2	DDD
17	0.013492063	2	D1-D2-R1	DDR
29	0.023015873	2	D1-D2-R2	DDR
56	0.044444444	2	D1-R1-D1	DRD
29	0.023015873	2	D1-R1-D2	DRD
5	0.003968254	2	D1-R1-R1	DRR
28	0.022222222	2	D1-R1-R2	DRR
49	0.038888889	2	D1-R2-D1	DRD
21	0.016666667	2	D1-R2-D2	DRD
23	0.018253968	2	D1-R2-R1	DRR
6	0.004761905	2	D1-R2-R2	DRR
14	0.011111111	2	D2-D1-D1	DDD
61	0.048412698	2	D2-D1-D2	DDD
38	0.03015873	2	D2-D1-R1	DDR
21	0.016666667	2	D2-D1-R2	DDR
9	0.007142857	2	D2-D2-D1	DDD
6	0.004761905	2	D2-D2-D2	DDD

1	0.000793651	2	D2-D2-R1	DDR
4	0.003174603	2	D2-D2-R2	DDR
13	0.01031746	2	D2-R1-D1	DRD
36	0.028571429	2	D2-R1-D2	DRD
6	0.004761905	2	D2-R1-R1	DRR
25	0.01984127	2	D2-R1-R2	DRR
26	0.020634921	2	D2-R2-D1	DRD
40	0.031746032	2	D2-R2-D2	DRD
15	0.011904762	2	D2-R2-R1	DRR
2	0.001587302	2	D2-R2-R2	DRR
3	0.002380952	2	R1-D1-D1	RDD
27	0.021428571	2	R1-D1-D2	RDD
39	0.030952381	2	R1-D1-R1	RDR
24	0.019047619	2	R1-D1-R2	RDR
20	0.015873016	2	R1-D2-D1	RDD
8	0.006349206	2	R1-D2-D2	RDD
46	0.036507937	2	R1-D2-R1	RDR
15	0.011904762	2	R1-D2-R2	RDR
6	0.004761905	2	R1-R1-D1	RRD
3	0.002380952	2	R1-R1-D2	RRD
7	0.005555556	2	R1-R1-R1	RRR
6	0.004761905	2	R1-R1-R2	RRR
36	0.028571429	2	R1-R2-D1	RRD
17	0.013492063	2	R1-R2-D2	RRD
33	0.026190476	2	R1-R2-R1	RRR
2	0.001587302	2	R1-R2-R2	RRR
5	0.003968254	2	R2-D1-D1	RDD
33	0.026190476	2	R2-D1-D2	RDD
34	0.026984127	2	R2-D1-R1	RDR
46	0.036507937	2	R2-D1-R2	RDR
28	0.022222222	2	R2-D2-D1	RDD
1	0.000793651	2	R2-D2-D2	RDD
16	0.012698413	2	R2-D2-R1	RDR
35	0.027777778	2	R2-D2-R2	RDR
18	0.014285714	2	R2-R1-D1	RRD
21	0.016666667	2	R2-R1-D2	RRD
4	0.003174603	2	R2-R1-R1	RRR
29	0.023015873	2	R2-R1-R2	RRR
7	0.005555556	2	R2-R2-D1	RRD
2	0.001587302	2	R2-R2-D2	RRD
1	0.000793651	2	R2-R2-R1	RRR
2	0.001587302	2	R2-R2-R2	RRR

5. DISKUSIJA

5.1. Osvrt na računalnu podršku

Razvijena računalna podrška olakšava analizu segmentiranog govora kolaborativnih aktivnosti, pogotovo ako se u segmentaciji nalaze preklopi sudionika. Računalna podrška je implementirana tako da se računalni program može prilagođavati za razne svrhe. Takav pristup ima i svoje prednosti i nedostatke. Ako je korisnik upoznat s MATLAB računalnim jezikom, program je moguće izmijeniti kako bi se analizirali različiti tipovi segmenata (npr. FBS ontologija). Ta fleksibilnost za napredne korisnike je najveća prednost ovakve implementacije. Najveći nedostatak je to što je potrebno znanje MATLAB-a i za male prilagodbe, npr. za analizu aktivnosti s različitim brojem članova. U daljnjem razvoju bi svakako trebalo poboljšati korisničko iskustvo kako bi se računalna podrška mogla lakše primjenjivati u istraživanjima.

Najbitnija funkcija ove računalne podrške je uklanjanje preklopa segmenata govora. Najlakša metoda za segmentaciju govora je uporaba alata koji primjenjuju strojno učenje [37], čiji je problem loše prepoznavanje sudionika [39]. Prema tome, ako se vrši zasebna automatska segmentacija audio zapisa za svakog sudionika (svaki audio zapis sadrži govor samo jednog sudionika) preostaje problem preklopa. Ako se preklopi uklanjaju prema tome koji sudionik je dominantniji, onda je najjednostavnija metoda usporedba glasnoće sudionika. Najveći problem te metode je utjecaj buke na rezultate [34]. Kako je već određeno koji dijelovi audio zapisa sadržavaju govor automatskom segmentacijom, buka nije uključena u analizu. Prema tome, primijenjeni algoritam uklanjanja preklopa prema kriteriju glasnoće omogućuje uporabu automatske segmentacije, što značajno olakšava pripremu ulaznih podataka.

Becattini i Alberio [21] su razvili računalnu podršku koja također analizira broj segmenata sudionika i trajanje govora sudionika. Njihova podrška kao metriku trajanja govora uzima prosječno trajanje segmenata sudionika, a u ovom radu se promatra ukupno trajanje govora sudionika. U njihovoj računalnoj podršci je integrirana i segmentacija govora, tako da su ulazni podaci samo audio zapisi sesije što još više olakšava proceduru analize komunikacijskih struktura. Takav pristup, iako pojednostavljuje pripremu ulaznih podataka, ograničava korisnika samo na njihovu metodu segmentacije. Prednost računalne podrške opisane u ovom radu je omogućavanje uporabe i ručne segmentacije. To je važno jer se automatskom segmentacijom još uvijek dobivaju kvalitetnije segmentacije u odnosu na automatsku segmentaciju [44].

Gero i sur. [45] su razvili alat za analizu pomoću FBS ontologije. Njihov alat je ipak orijentiran prema kontekstualnoj analizi komunikacije. Najveća prednost njihovog alata je grafičko sučelje što uvelike olakšava korištenje računalne podrške. To omogućava uporabu računalne podrške i osobama koje nemaju nikakvog iskustva s računalnim programiranjem. Kao što je već spomenuto, prednost pristupa opisanog u ovome radu je to što omogućava veću fleksibilnost korisnicima koji su poznati s MATLAB računalnim jezikom.

5.2. Dodane vrijednosti

Primarni cilj ovog rada je omogućiti bolje razumijevanje korištenja VS-a tijekom pregleda konstrukcije. Analizom komunikacije razvojnih timova unutar VS-a moguće je istaknuti potencijalne probleme. Na primjer, moguće je prepoznati situacije u kojima neki članovi tima previše ili premalo pričaju. Kan i Gero [46] predlažu da kolaborativne konstrukcijske aktivnosti s većom izmjenom informacija rezultiraju kreativnijim rješenjima. Prema tome, moglo bi biti korisno prilikom pregleda konstrukcije poticati sudionike da svi sudjeluju u razgovoru. Pomoću rezultata koje dobijemo računalnom podrškom možemo odrediti koje sudionike treba poticati da sudjeluju više u razgovoru, a koje manje. Informacije o interakcijama između sudionika nam pružaju uvid u to koliko koji sudionici komuniciraju međusobno. Tako bi se moglo uočiti ako neki sudionici međusobno premalo ili previše komuniciraju, temeljem čega bi se moglo također potaknuti sudionike na ravnomjerniju suradnju. Moguće je uočiti i postojanje podtimova ako se primijeti da određena grupa sudionika ima visoku količinu međusobnih interakcija. Ako bi prilagodili ovakvu računalnu podršku za praćenje konstrukcijskih timova u stvarnom vremenu, omogućilo bi se automatizirano moderiranje pregleda konstrukcije. Kako bi se bolje razumio utjecaj interakcija sudionika na pregled konstrukcije, potrebno je napraviti veći broj istraživanja. Prvi korak u tome jest razvoj metodologija i alata za podršku istraživanja, kao što je računalna podrška opisana u ovome radu.

S obzirom na to da većina VS sustava ima u sebi ugrađen mikrofon [15], jednostavno je snimati zasebne audio zapise svakog sudionika. To olakšava uporabu automatske segmentacije [40] što značajno ubrzava dobivanje rezultata. Zbog toga je računalna podrška prilagođena za analizu segmentiranog govora dobivenog automatskom segmentacijom, ali omogućava i korištenje ručne segmentacije.

Primjenom računalne podrške na više sesija omogućava usporedbu rezultata različitih situacija. Na primjer, ako se želi promatrati razlika između pregleda konstrukcija koje koriste

tradicionalne CAD alate i VS sustave, razvijena računalna podrška omogućava brz način analize komunikacijske strukture oba okruženja. Također, za uspoređivanje je važno imati i ujednačenu metodologiju za ispitivanje svake sesije pregleda konstrukcije. Uporabom računalnog programa osigurava se da je svaka sesija analizirana na isti način, pogotovo ako se koristi i ista metoda automatske segmentacije govora za svaku sesiju.

Kako razvijena računalna podrška ne uzima u obzir kontekst, nije ograničena samo na analiziranje pregleda konstrukcija. Program je moguće primijeniti za analizu raznih vrsta kolaborativnih aktivnosti s malo prilagodbe (promjena broja sudionika, nazivi sudionika, itd.).

5.3. Budući razvoj računalne podrške

Kako je spomenuto u prethodnim potpoglavljima, jedan smjer daljnjeg razvoja je praćenje kolaborativnih aktivnosti u stvarnom vremenu. Praćenje pregleda konstrukcije u stvarnom vremenu bi omogućilo da članovi tima imaju uvid u dominantnost sudionika. Ako je poznato koji sudionici manje sudjeluju u razgovoru, onda bi ih se moglo tijekom sesije potaknuti da pokušaju ujednačiti svoj doprinos pregledu konstrukcije.

Drugi smjer u kojemu bi valjalo nastaviti razvijati računalnu podršku je uključivanje konteksta. Kada bi se uključile, na primjer, informacije o temama razgovora, moglo bi se još detaljnije analizirati sudjelovanje članova razvojnog tima. Uspoređivanjem aktivnosti sudionika tijekom rasprave o različitim temama tijekom pregleda konstrukcije moglo bi se ustanoviti koji članovi tima su stručniji u kojem području, koji previše razgovaraju o određenim temama, koji mijenjaju teme razgovora i sl. Svi takvi podaci mogu pružiti još kvalitetniju analizu kolaboracijskih aktivnosti pomoću koje se mogu stvarati prijedlozi za provođenje kolaboracijskih aktivnosti, stvaranje kreativnijih timova i sl. Na primjer, ako bi se uočilo da jedan član tima često puno govori o ergonomiji, a nije još uključen u preglede tijekom kojih je primarni cilj analiza ergonomije, bilo bi dobro ubuduće i njega uključiti u takve aktivnosti. Ako se vidi da jedan sudionik često ne sudjeluje puno u razgovoru o mogućnostima proizvodnje i pokušava promijeniti temu s proizvodnje, to bi moglo značiti da taj član tima nije zainteresiran za tu temu.

Treći aspekt koji bi valjalo svakako poboljšati je lakoća korištenja računalne podrške. S obzirom na to da istraživači nisu nužnoiskusni u računalnom programiranju, stvaranjem korisničkog sučelja i omogućavanjem lakše izmjene barem osnovnih parametara (broj

sudionika, vrste sudionika i sl.) bi se uvelike proširilo broj korisnika koji bi mogli primijeniti računalnu podršku.

6. ZAKLJUČAK

Razvijena računalna podrška je računalni program koji služi za analizu komunikacijske strukture razvojnog tima tijekom pregleda konstrukcije u VS-u. Ulazni podaci za računalnu podršku su video zapisi i segmentirani govor svakog člana tima u zasebnim datotekama. Ti podaci se učitavaju u program te se oblikuju za daljnju obradu. Zatim se uklanjaju svi preklopi u segmentiranom govoru. Za uklanjanje preklopa potrebno je ukloniti dio preklopljenog segmenta manje dominantnog sudionika. Uz pretpostavku da je glasniji sudionik dominantniji [34], uklanja se preklopljeni dio segmenta tišeg sudionika. Nakon što su uklonjeni svi preklopi, slijedi analiza podataka. To uključuje provođenje Markovljeve analize, izračun vremena govora i pokretnih prosjeka govora sudionika. Program završava ispisom rezultata analize, čime dobijemo izlazne podatke. Izlazni podaci su .xlsx tablica s rezultatima Markovljeve analize i .png grafovi koji prikazuju pokretne prosjeke i kumulativno vrijeme govora sudionika.

Na primjeru jednog pregleda konstrukcije četveročlanog tima u VS-u prikazan je rad programa. Program značajno ubrzava analizu komunikacijske strukture, pogotovo ako se koristi automatska segmentacija za pripremu ulaznih podataka. Vrijeme potrebno za analizu sesije iz primjera pomoću razvijene računalne podrške je 5,2 minute. U primjeru je uočeno da su konstruktori pričali puno više od recenzenata, ali da razlika u količini segmenata između sudionika nije velika. Također, uočeno je da veliki udio interakcija između sudionika odgovara interakcijama između sudionika različitog tipa, to jest recenzenata i konstruktora, u odnosu na interakcije između sudionika istog tipa.

Računalna podrška omogućava lakšu i bržu analizu komunikacijskih struktura kolaborativnih aktivnosti. Rezultati dobiveni primjenom računalne podrške daju uvid u dinamiku između sudionika. Moguće je odrediti koji sudionici sudjeluju previše ili premalo u pregledu konstrukcije, prema čemu valja poticati sve sudionike da jednako doprinose pregledu. Računalnom podrškom može se analizirati više kolaborativnih aktivnosti pomoću jednakih kriterija. Time je omogućeno lakše analiziranje i uspoređivanje setova podataka.

Iako je računalna podrška namijenjena za analizu pregleda konstrukcija, moguće ju je primijeniti i u druge svrhe s malo prilagodbe. Prepoznata su tri potencijalna smjera daljnjeg razvoja računalne podrške. Jedan smjer je prilagodba za podršku u stvarnom vremenu, drugi je razvijanje računalne podrške koja će uzimati u obzir i kontekst i zadnji cilj je omogućavanje lakšeg korištenja.

LITERATURA

- [1] P. J. Cash, B. J. Hicks, and S. J. Culley, "A comparison of designer activity using core design situations in the laboratory and practice," *Design Studies*, vol. 34, no. 5, 2013, doi: 10.1016/j.destud.2013.03.002.
- [2] G. Huet, S. J. Culley, C. A. McMahon, and C. Fortin, "Making sense of engineering design review activities," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, vol. 21, no. 3. 2007. doi: 10.1017/S0890060407000261.
- [3] C. A. Lauff, D. Knight, D. Kotys-Schwartz, and M. E. Rentschler, "The role of prototypes in communication between stakeholders," *Design Studies*, vol. 66, 2020, doi: 10.1016/j.destud.2019.11.007.
- [4] S. K. Chandrasegaran *et al.*, "The evolution, challenges, and future of knowledge representation in product design systems," in *CAD Computer Aided Design*, 2013, vol. 45, no. 2. doi: 10.1016/j.cad.2012.08.006.
- [5] M. Bassanino, T. Fernando, and K. C. Wu, "Can virtual workspaces enhance team communication and collaboration in design review meetings?," *Architectural Engineering and Design Management*, vol. 10, no. 3–4, 2014, doi: 10.1080/17452007.2013.775102.
- [6] Y. Liu, F. Castronovo, J. Messner, and R. Leicht, "Evaluating the Impact of Virtual Reality on Design Review Meetings," *Journal of Computing in Civil Engineering*, vol. 34, no. 1, 2020, doi: 10.1061/(asce)cp.1943-5487.0000856.
- [7] L. P. Berg and J. M. Vance, "Industry use of virtual reality in product design and manufacturing: a survey," *Virtual Reality*, vol. 21, no. 1, 2017, doi: 10.1007/s10055-016-0293-9.
- [8] O. Eris, N. Martelaro, and P. Badke-Schaub, "A comparative analysis of multimodal communication during design sketching in co-located and distributed environments," *Design Studies*, vol. 35, no. 6, 2014, doi: 10.1016/j.destud.2014.04.002.
- [9] G. Q. Huang, "Web-based support for collaborative product design review," in *Computers in Industry*, 2002, vol. 48, no. 1. doi: 10.1016/S0166-3615(02)00011-8.
- [10] B. S. Dhillon, *Engineering Design: A Modern Approach*. Irwin, 1996.

-
- [11] Y. Liu, J. I. Messner, and R. M. Leicht, "A process model for usability and maintainability design reviews," *Architectural Engineering and Design Management*, vol. 14, no. 6, 2018, doi: 10.1080/17452007.2018.1512042.
- [12] D. Unger and S. Eppinger, "Improving product development process design: A method for managing information flows, risks, and iterations," *Journal of Engineering Design*, vol. 22, no. 10, 2011, doi: 10.1080/09544828.2010.524886.
- [13] C. Cruz-Neira, D. J. Sandin, T. A. DeFanti, R. v. Kenyon, and J. C. Hart, "The CAVE: Audio Visual Experience Automatic Virtual Environment," *Communications of the ACM*, vol. 35, no. 6, 1992, doi: 10.1145/129888.129892.
- [14] A. Berni and Y. Borgianni, "Applications of virtual reality in engineering and product design: Why, what, how, when and where," *Electronics (Switzerland)*, vol. 9, no. 7, 2020. doi: 10.3390/electronics9071064.
- [15] J. Q. Coburn, I. Freeman, and J. L. Salmon, "A Review of the Capabilities of Current Low-Cost Virtual Reality Technology and Its Potential to Enhance the Design Process," *Journal of Computing and Information Science in Engineering*, vol. 17, no. 3, 2017. doi: 10.1115/1.4036921.
- [16] M. Bassanino, K. C. Wu, J. Yao, F. Khosrowshahi, T. Fernando, and J. Skjærbæk, "The impact of immersive virtual reality on visualisation for a design review in construction," 2010. doi: 10.1109/IV.2010.85.
- [17] J. Wolfartsberger, "Analyzing the potential of Virtual Reality for engineering design review," *Automation in Construction*, vol. 104, 2019, doi: 10.1016/j.autcon.2019.03.018.
- [18] L. de Casenave and J. E. Lugo, "Design review using virtual reality enabled cad," in *Proceedings of the ASME Design Engineering Technical Conference*, 2017, vol. 1. doi: 10.1115/DETC2017-67878.
- [19] N. Horvat, S. Škec, T. Martinec, F. Lukacevic, and M. M. Perišić, "Comparing virtual reality and desktop interface for reviewing 3D CAD models," in *Proceedings of the International Conference on Engineering Design, ICED*, 2019, vol. 2019-August. doi: 10.1017/dsi.2019.198.
- [20] I. Freeman, J. Salmon, and J. Coburn, "A bi-directional interface for improved interaction with engineering models in virtual reality design reviews," *International*

- Journal on Interactive Design and Manufacturing*, vol. 12, no. 2. 2018. doi: 10.1007/s12008-017-0413-0.
- [21] N. Becattini and L. Alberio, "Semi-automatic Analysis of Spoken Interaction Dynamics in Collaborative Design Sessions," 2022. doi: 10.1007/978-3-030-91234-5_19.
- [22] J. S. Gero and J. Milovanovic, "A framework for studying design thinking through measuring designers' minds, bodies and brains," *Design Science*, 2020, doi: 10.1017/dsj.2020.15.
- [23] R. T. A. J. Leenders, J. M. L. van Engelen, and J. Kratzer, "Virtuality, communication, and new product team creativity: A social network perspective," *Journal of Engineering and Technology Management - JET-M*, vol. 20, no. 1-2 SPEC., 2003, doi: 10.1016/S0923-4748(03)00005-5.
- [24] H. Jiang and J. S. Gero, "Comparing Two Approaches to Studying Communications in Team Design," in *Design Computing and Cognition '16*, 2017. doi: 10.1007/978-3-319-44989-0_17.
- [25] L. T. M. Blessing and A. Chakrabarti, *DRM, a design research methodology*. 2009. doi: 10.1007/978-1-84882-587-1.
- [26] M. W. van Someren, Y. F. Barnard, and J. A. C. Sandberg, *The think aloud method: a practical guide to modelling cognitive processes*. London: Academic Press, 1994.
- [27] J. S. Gero and U. Kannengiesser, "The Function-Behaviour-Structure Ontology of Design," in *An Anthology of Theories and Models of Design*, 2014. doi: 10.1007/978-1-4471-6338-1_13.
- [28] J. W. T. Kan and J. S. Gero, *Quantitative Methods for Studying Design Protocols*. 2017. doi: 10.1007/978-94-024-0984-0.
- [29] K. A. Ericsson and H. A. Simon, *Protocol analysis: Verbal reports as data (rev. ed.)*. 1993.
- [30] H. Sacks, E. A. Schegloff, and G. Jefferson, "A Simplest Systematics for the Organization of Turn-Taking for Conversation," *Language*, vol. 50, no. 4, p. 696, Dec. 1974, doi: 10.2307/412243.

- [31] M. Barthel, A. S. Meyer, and S. C. Levinson, "Next speakers plan their turn early and speak after turn-final 'go-signals,'" *Frontiers in Psychology*, vol. 8, no. MAR, 2017, doi: 10.3389/fpsyg.2017.00393.
- [32] C. Fredouille and N. Evans, "The influence of speech activity detection and overlap on speaker diarization for meeting room recordings," in *International Speech Communication Association - 8th Annual Conference of the International Speech Communication Association, Interspeech 2007*, 2007, vol. 4. doi: 10.21437/interspeech.2007-736.
- [33] H. Hung, Y. Huang, G. Friedland, and D. Gatica-Perez, "Estimating the dominant person in multi-party conversations using speaker diarization strategies," 2008. doi: 10.1109/ICASSP.2008.4518080.
- [34] I. Volfin and I. Cohen, "Dominant speaker identification for multipoint videoconferencing," *Computer Speech and Language*, vol. 27, no. 4, 2013, doi: 10.1016/j.csl.2012.03.002.
- [35] J. Ramírez, J. C. Segura, C. Benítez, L. García, and A. Rubio, "Statistical voice activity detection using a multiple observation likelihood ratio test," *IEEE Signal Processing Letters*, vol. 12, no. 10, 2005, doi: 10.1109/LSP.2005.855551.
- [36] H. K. Maganti, P. Motlicek, and D. Gatica-Perez, "Unsupervised speech/non-speech detection for automatic speech recognition in meeting rooms," in *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings*, 2007, vol. 4. doi: 10.1109/ICASSP.2007.367250.
- [37] Y. Liu, N. v. Chawla, M. P. Harper, E. Shriberg, and A. Stolcke, "A study in machine learning from imbalanced data for sentence boundary detection in speech," *Computer Speech and Language*, vol. 20, no. 4, 2006, doi: 10.1016/j.csl.2005.06.002.
- [38] F. Galbusera, G. Casaroli, and T. Bassani, "Artificial intelligence and machine learning in spine research," *JOR Spine*, vol. 2, no. 1. 2019. doi: 10.1002/jsp2.1044.
- [39] A. Addlesee, Y. Yu, and A. Eshghi, "A Comprehensive Evaluation of Incremental Speech Recognition and Diarization for Conversational AI," 2021. doi: 10.18653/v1/2020.coling-main.312.

- [40] H. Jiang and C.-C. Yen, "Protocol Analysis in Design Research: a review," in *International Association of Societies of Design Research Conference*, 2010, vol. 78, no. 24.
- [41] "MATLAB," <https://www.mathworks.com/products/matlab.html>, Feb. 19, 2022.
- [42] International Organization for Standardization, *Acoustics – Method for calculating loudness – Part 1: Zwicker method*, vol. 7. 2017.
- [43] J. Tolins and J. E. Fox Tree, "Addressee backchannels steer narrative development," *Journal of Pragmatics*, vol. 70, 2014, doi: 10.1016/j.pragma.2014.06.006.
- [44] J. Rodd, C. Decuyper, H. R. Bosker, and L. ten Bosch, "A tool for efficient and accurate segmentation of speech data: announcing POnSS," *Behavior Research Methods*, vol. 53, no. 2, 2021, doi: 10.3758/s13428-020-01449-6.
- [45] J. S. Gero, J. W. Kan, and M. Pourmohamadi, "Analysing Design Protocols: Development of Methods and Tools," *Research into Design*, 2011.
- [46] J. W. T. Kan and J. S. Gero, "Characterizing innovative processes in design spaces through measuring the information entropy of empirical data from protocol studies," *Artificial Intelligence for Engineering Design, Analysis and Manufacturing: AIEDAM*, vol. 32, no. 1, 2018, doi: 10.1017/S0890060416000548.

PRILOZI

I. MATLAB računalni program

Prilog I. MATLAB računalni program

```
clear
```

```
%% Load data
```

```
environment = 'VR';
```

```
session = 1;
```

```
path = [environment, '/', num2str(session)];
```

```
data_separate = cell(1,8); % 1 = 1 D1, 2 = 1 D2, 3 = 1 R1, 4 = 1 R2, 5 = 2 D1, 6 = 2 D2, 7  
= 2 R1, 8 = 2 R2
```

```
raw_audio_separate = cell(1,8);
```

```
data_separate{1} = importdata([path,  
'/data.mat']).Labels.SpeechDetected{1}.ROILimits;
```

```
[raw_audio_separate{1},sr] = audioread([path, '/1_D1.mp3']);
```

```
data_separate{2} = importdata([path,  
'/data.mat']).Labels.SpeechDetected{2}.ROILimits;
```

```
[raw_audio_separate{2},sr] = audioread([path, '/1_D2.mp3']);
```

```
data_separate{3} = importdata([path,  
'/data.mat']).Labels.SpeechDetected{3}.ROILimits;
```

```
[raw_audio_separate{3},sr] = audioread([path, '/1_R1.mp3']);
```

```
data_separate{4} = importdata([path,  
'/data.mat']).Labels.SpeechDetected{4}.ROILimits;
```

```
[raw_audio_separate{4},sr] = audioread([path, '/1_R2.mp3']);
```

```
data_separate{5} = importdata([path,  
'/data.mat']).Labels.SpeechDetected{5}.ROILimits;
```

```
[raw_audio_separate{5},sr] = audioread([path, '/2_D1.mp3']);
```

```
data_separate{6} = importdata([path,  
'/data.mat']).Labels.SpeechDetected{6}.ROILimits;
```

```

[raw_audio_separate{6},sr] = audioread([path, '/2_D2.mp3']);
data_separate{7}          = importdata([path,
    '/data.mat']).Labels.SpeechDetected{7}.ROILimits;
[raw_audio_separate{7},sr] = audioread([path, '/2_R1.mp3']);
data_separate{8}          = importdata([path,
    '/data.mat']).Labels.SpeechDetected{8}.ROILimits;
[raw_audio_separate{8},sr] = audioread([path, '/2_R2.mp3']);

data      = cell(1,4); % 1 = D1, 2 = D2, 3 = R1, 4 = R2
raw_audio = cell(1,4);
for n = 1:4
    data{n}      =
        cat(1,data_separate{n},data_separate{n+4}+data_separate{n}(length(data_separate{n}
            ),2)); %Holds all data, session types joined, participants separated
    data{n}(:,3) = data{n}(:,2)-data{n}(:,1); %Adds turn duration vector
    raw_audio{n} =
        cat(1,raw_audio_separate{n},raw_audio_separate{n+4}+raw_audio_separate{n}(length
            (raw_audio_separate{n}),2));
end
data_titles = ["D1" "D2" "R1" "R2"];

%Create participant and position vectors
data_table_participants =
    string(1:(length(data{1}(:,1))+length(data{2}(:,1))+length(data{3}(:,1))+length(data{
        4}(:,1))));
data_table_position =
    int16(1:(length(data{1}(:,1))+length(data{2}(:,1))+length(data{3}(:,1))+length(data{4}
        }(:,1))));

```

```

for n = 1:(length(data{1}(:,1)))
    data_table_participants(n) = "D1";
    data_table_position(n) = n;
end
for n = length(data{1}(:,1))+1:length(data{1}(:,1))+length(data{2}(:,1))
    data_table_participants(n) = "D2";
    data_table_position(n) = n - length(data{1}(:,1));
end
for n =
    length(data{1}(:,1))+length(data{2}(:,1))+1:length(data{1}(:,1))+length(data{2}(:,1))
    +length(data{3}(:,1))
    data_table_participants(n) = "R1";
    data_table_position(n) = n - length(data{1}(:,1)) - length(data{2}(:,1));
end
for n =
    length(data{1}(:,1))+length(data{2}(:,1))+length(data{3}(:,1))+1:length(data{1}(:,1))
    +length(data{2}(:,1))+length(data{3}(:,1))+length(data{4}(:,1))
    data_table_participants(n) = "R2";
    data_table_position(n) = n - length(data{1}(:,1)) - length(data{2}(:,1)) -
        length(data{3}(:,1));
end

%Continuous data in a table - sorted by start time
%includes Start, End, Duration, Participant
data_table_unsorted =
    table(cat(1,data{1}(:,1),data{2}(:,1),data{3}(:,1),data{4}(:,1)),cat(1,data{1}(:,2),data{
2}(:,2),data{3}(:,2),data{4}(:,2)),cat(1,data{1}(:,3),data{2}(:,3),data{3}(:,3),data{4}(:,
3)),transpose(data_table_participants),transpose(data_table_position),'VariableNames'
,{'Start';'End';'Duration';'Participant';'Index'});

```

```
data_table = sortrows(data_table_unsorted);
disp('Data loading DONE')

%% Prepare data
n_overlaps = 1; %Remove overlaps
while n_overlaps > 0
    n_overlaps = 0;
    for n = 1:length(data_table.Start)
        if n > length(data_table.Start)
            break
        end
        if n == 1
            i_range = (n+1:n+20);
        elseif n == length(data_table.Start)
            i_range = (n-20:n-1);
        elseif n <= 20
            i_range = [1:n-1 n+1:n+20];
        elseif length(data_table.Start) - n <= 20
            i_range = [n-20:n-1 n+1:length(data_table.Start)];
        else
            i_range = [n-20:n-1 n+1:n+20];
        end
        if data_table.Participant(n) == "D1"
            participantn = 1;
        elseif data_table.Participant(n) == "D2"
            participantn = 2;
        elseif data_table.Participant(n) == "R1"
            participantn = 3;
```

```
elseif data_table.Participant(n) == "R2"
    participantn = 4;
end
for i = i_range
    if data_table.Participant(i) == "D1"
        participanti = 1;
    elseif data_table.Participant(i) == "D2"
        participanti = 2;
    elseif data_table.Participant(i) == "R1"
        participanti = 3;
    elseif data_table.Participant(i) == "R2"
        participanti = 4;
    end
    if and(data_table.Start(i) == data_table.Start(n), data_table.End(i) ==
data_table.End(n)) % case 1
        n_overlaps = n_overlaps + 1;
        sonen =
acousticLoudness(raw_audio{participantn}(round(data_table.Start(n)*sr:data_table.En
d(n)*sr),:),sr);
        sonei =
acousticLoudness(raw_audio{participanti}(round(data_table.Start(i)*sr:data_table.En
d(i)*sr),:),sr);
        if mean(sonen) > mean(sonei)
            data_table(i,:) = [];
        elseif mean(sonei) > mean(sonen)
            data_table(n,:) = [];
        else
            disp(['Equal loudness @ n=',num2str(n), ' and i=', num2str(i)])
        end
    end
end
```

```
break

elseif and(data_table.Start(i) == data_table.Start(n), data_table.End(i) <
data_table.End(n)) %case 2

    n_overlaps = n_overlaps + 1;

    sonen      =
acousticLoudness(raw_audio{participantn}(round(data_table.Start(n)*sr:data_table.En
d(i)*sr),:),sr);

    sonei      =
acousticLoudness(raw_audio{participanti}(round(data_table.Start(n)*sr:data_table.En
d(i)*sr),:),sr);

    if mean(sonen) > mean(sonei)
        data_table(i,:) = [];
    elseif mean(sonei) > mean(sonen)
        data_table.Start(n) = data_table.End(i);
        temp                = data_table(n,:);
        data_table(n,:)     = data_table(i,:);
        data_table(i,:)     = temp;
    else
        disp(['Equal loudness @ n=',num2str(n), ' and i=', num2str(i)])
    end

break

elseif and(data_table.Start(i) > data_table.Start(n), data_table.End(i) ==
data_table.End(n)) %case 3

    n_overlaps = n_overlaps + 1;

    sonen      =
acousticLoudness(raw_audio{participantn}(round(data_table.Start(i)*sr:data_table.En
d(n)*sr),:),sr);
```

```

    sonei    =
acousticLoudness(raw_audio{participanti}(round(data_table.Start(i)*sr:data_table.En
d(n)*sr),:),sr);
    if mean(sonen) > mean(sonei)
        data_table(i,:) = [];
    elseif mean(sonei) > mean(sonen)
        data_table.End(n) = data_table.Start(i);
    else
        disp(['Equal loudness @ n=',num2str(n), ' and i=', num2str(i)])
    end
    break
elseif and(data_table.Start(i) > data_table.Start(n), data_table.End(i) <
data_table.End(n)) %case 4
    n_overlaps = n_overlaps + 1;
    sonen    =
acousticLoudness(raw_audio{participantn}(round(data_table.Start(i)*sr:data_table.En
d(i)*sr),:),sr);
    sonei    =
acousticLoudness(raw_audio{participanti}(round(data_table.Start(i)*sr:data_table.En
d(i)*sr),:),sr);
    if mean(sonen) > mean(sonei)
        data_table(i,:) = [];
    elseif mean(sonei) > mean(sonen)
        temp      = {data_table.End(i), data_table.End(n), 1,
data_table.Participant(n), data_table.Index(n)};
        data_table.End(n) = data_table.Start(i);
        data_table      = [data_table;temp];
        data_table      = sortrows(data_table);
    else

```

```
        disp(['Equal loudness @ n=',num2str(n), ' and i=', num2str(i)])
    end
    break

elseif and(data_table.Start(i) > data_table.Start(n), data_table.Start(i) <
data_table.End(n)) %case 5

    n_overlaps = n_overlaps + 1;

    sonen =
acousticLoudness(raw_audio{participantn}(round(data_table.Start(i)*sr:data_table.En
d(n)*sr),:),sr);

    sonei =
acousticLoudness(raw_audio{participanti}(round(data_table.Start(i)*sr:data_table.En
d(n)*sr),:),sr);

    if mean(sonen) > mean(sonei)
        data_table.Start(i) = data_table.End(n);
    elseif mean(sonei) > mean(sonen)
        data_table.End(n) = data_table.Start(i);
    else
        disp(['Equal loudness @ n=',num2str(n), ' and i=', num2str(i)])
    end
    break

elseif and(data_table.End(i) > data_table.Start(n), data_table.End(i) <
data_table.End(n)) %case 6

    n_overlaps = n_overlaps + 1;

    sonen =
acousticLoudness(raw_audio{participantn}(round(data_table.Start(n)*sr:data_table.En
d(i)*sr),:),sr);

    sonei =
acousticLoudness(raw_audio{participanti}(round(data_table.Start(n)*sr:data_table.En
d(i)*sr),:),sr);
```

```
        if mean(sonen) > mean(sonei)
            data_table.End(i) = data_table.Start(n);
        elseif mean(sonei) > mean(sonen)
            data_table.Start(n) = data_table.End(i);
        else
            disp(['Equal loudness @ n=', num2str(n), ' and i=', num2str(i)])
        end
        break
    end
end
end
disp('Overlap pass DONE')
end

data_table.Duration(:) = data_table.End(:) - data_table.Start(:); % Append durations

% Appending indexes
iD1 = 0;
iD2 = 0;
iR1 = 0;
iR2 = 0;
for n = 1:length(data_table.Participant(:))
    if data_table.Participant(n) == "D1"
        iD1 = iD1 + 1;
        data_table.Index(n) = iD1;
    elseif data_table.Participant(n) == "D2"
        iD2 = iD2 + 1;
        data_table.Index(n) = iD2;
    end
end
```

```
elseif data_table.Participant(n) == "R1"
    iR1      = iR1 + 1;
    data_table.Index(n) = iR1;
elseif data_table.Participant(n) == "R2"
    iR2      = iR2 + 1;
    data_table.Index(n) = iR2;
end
end
end
% Append data (cell) variable
data_table = sortrows(data_table,[4 1]);
data{1}    = table2array(data_table(1:iD1,1:3));
data{2}    = table2array(data_table(iD1+1:iD1+iD2,1:3));
data{3}    = table2array(data_table(iD1+iD2+1:iD1+iD2+iR1,1:3));
data{4}    = table2array(data_table(iD1+iD2+iR1+1:iD1+iD2+iR1+iR2,1:3));
data_table = sortrows(data_table, 1);

disp('Data prep DONE')

%% Data analysis

% Cumulative duration
cumulative_duration = cell(1,4);
for n = 1:4
    for i = 1:length(data{n}(:,1))
        cumulative_duration{n}(i) = sum(data{n}(1:i,3)); % Cumulative talking time vectors, per
            participant
    end
end
end
```

```
%Moving average
data_seconds = zeros(4000,4); %Amount of time spent talking in a given second, per
    participant (1-4)
moving_window = 120; %seconds
for participant = 1:4
    for i = 2:length(data{participant}(:,1))
        if floor(data{participant}(i,1)) == floor(data{participant}(i,2))
            data_seconds(floor(data{participant}(i,1)),participant) =
                data_seconds(floor(data{participant}(i,1)),participant) + data{participant}(i,3);
        else
            data_seconds(floor(data{participant}(i,1)),participant) =
                data_seconds(floor(data{participant}(i,1)),participant) + (ceil(data{participant}(i,1)) -
                data{participant}(i,1));
            n = ceil(data{participant}(i,1));
            while n < floor(data{participant}(i,2))
                data_seconds(n,participant) = data_seconds(n,participant) + 1 ;
                n = n+1;
            end
            data_seconds(n,participant) = data_seconds(n,participant) + (data{participant}(i,2) -
            floor(data{participant}(i,2)));
        end
    end
end

%Normalized moving average
data_normalized = zeros(1000,4); %0.1% steps
data_duration = [find(data_seconds(:,1),1,'last') find(data_seconds(:,2),1,'last')
    find(data_seconds(:,3),1,'last') find(data_seconds(:,4),1,'last')];
```

```

movmean_norm_span = data_duration/1000;
for participant = 1:4
    data_normalized(1,participant) =
        sum(data_seconds(1:round(movmean_norm_span(participant)),participant))/data_dura
        tion(participant);
    for n = 1:999
        data_normalized(n+1,participant) =
            sum(data_seconds(round(n*movmean_norm_span(participant)):round((n+1)*movmea
            n_norm_span(participant)),participant))/data_duration(participant);
    end
end
end

```

%Participant dynamics

```

participant_order = ["D1";"D2";"R1";"R2";"D1-D1";"D1-D2";"D1-R1";"D1-R2";"D2-
    D1";"D2-D2";"D2-R1";"D2-R2";"R1-D1";"R1-D2";"R1-R1";"R1-R2";"R2-D1";"R2-
    D2";"R2-R1";"R2-R2";"D1-D1-D1";"D1-D1-D2";"D1-D1-R1";"D1-D1-R2";"D1-D2-
    D1";"D1-D2-D2";"D1-D2-R1";"D1-D2-R2";"D1-R1-D1";"D1-R1-D2";"D1-R1-
    R1";"D1-R1-R2";"D1-R2-D1";"D1-R2-D2";"D1-R2-R1";"D1-R2-R2";"D2-D1-
    D1";"D2-D1-D2";"D2-D1-R1";"D2-D1-R2";"D2-D2-D1";"D2-D2-D2";"D2-D2-
    R1";"D2-D2-R2";"D2-R1-D1";"D2-R1-D2";"D2-R1-R1";"D2-R1-R2";"D2-R2-
    D1";"D2-R2-D2";"D2-R2-R1";"D2-R2-R2";"R1-D1-D1";"R1-D1-D2";"R1-D1-
    R1";"R1-D1-R2";"R1-D2-D1";"R1-D2-D2";"R1-D2-R1";"R1-D2-R2";"R1-R1-
    D1";"R1-R1-D2";"R1-R1-R1";"R1-R1-R2";"R1-R2-D1";"R1-R2-D2";"R1-R2-
    R1";"R1-R2-R2";"R2-D1-D1";"R2-D1-D2";"R2-D1-R1";"R2-D1-R2";"R2-D2-
    D1";"R2-D2-D2";"R2-D2-R1";"R2-D2-R2";"R2-R1-D1";"R2-R1-D2";"R2-R1-
    R1";"R2-R1-R2";"R2-R2-D1";"R2-R2-D2";"R2-R2-R1";"R2-R2-R2"];

```

```

participant_type_order =
    ["D";"D";"R";"R";"DD";"DD";"DR";"DR";"DD";"DD";"DR";"DR";"RD";"RD";"RR"
    ;"RR";"RD";"RD";"RR";"RR";"DDD";"DDD";"DDR";"DDR";"DDD";"DDD";"DDR
    ";"DDR";"DRD";"DRD";"DRR";"DRR";"DRD";"DRD";"DRR";"DRR";"DDD";"DD

```



```
    turns_table.Values(25)      = turns_table.Values(25) + 1;
elseif data_table.Participant(n+2) == "D2"
    turns_table.Values(26)      = turns_table.Values(26) + 1;
elseif data_table.Participant(n+2) == "R1"
    turns_table.Values(27)      = turns_table.Values(27) + 1;
elseif data_table.Participant(n+2) == "R2"
    turns_table.Values(28)      = turns_table.Values(28) + 1;
end
elseif data_table.Participant(n+1) == "R1"
    turns_table.Values(7)       = turns_table.Values(7) + 1;
if data_table.Participant(n+2) == "D1"
    turns_table.Values(29)      = turns_table.Values(29) + 1;
elseif data_table.Participant(n+2) == "D2"
    turns_table.Values(30)      = turns_table.Values(30) + 1;
elseif data_table.Participant(n+2) == "R1"
    turns_table.Values(31)      = turns_table.Values(31) + 1;
elseif data_table.Participant(n+2) == "R2"
    turns_table.Values(32)      = turns_table.Values(32) + 1;
end
elseif data_table.Participant(n+1) == "R2"
    turns_table.Values(8)       = turns_table.Values(8) + 1;
if data_table.Participant(n+2) == "D1"
    turns_table.Values(33)      = turns_table.Values(33) + 1;
elseif data_table.Participant(n+2) == "D2"
    turns_table.Values(34)      = turns_table.Values(34) + 1;
elseif data_table.Participant(n+2) == "R1"
    turns_table.Values(35)      = turns_table.Values(35) + 1;
elseif data_table.Participant(n+2) == "R2"
```

```
        turns_table.Values(36)    = turns_table.Values(36) + 1;
    end
end
elseif data_table.Participant(n) == "D2"
    if data_table.Participant(n+1) == "D1"
        turns_table.Values(9)     = turns_table.Values(9) + 1;
        if data_table.Participant(n+2) == "D1"
            turns_table.Values(37)  = turns_table.Values(37) + 1;
        elseif data_table.Participant(n+2) == "D2"
            turns_table.Values(38)  = turns_table.Values(38) + 1;
        elseif data_table.Participant(n+2) == "R1"
            turns_table.Values(39)  = turns_table.Values(39) + 1;
        elseif data_table.Participant(n+2) == "R2"
            turns_table.Values(40)  = turns_table.Values(40) + 1;
        end
    elseif data_table.Participant(n+1) == "D2"
        turns_table.Values(10)     = turns_table.Values(10) + 1;
        if data_table.Participant(n+2) == "D1"
            turns_table.Values(41)  = turns_table.Values(41) + 1;
        elseif data_table.Participant(n+2) == "D2"
            turns_table.Values(42)  = turns_table.Values(42) + 1;
        elseif data_table.Participant(n+2) == "R1"
            turns_table.Values(43)  = turns_table.Values(43) + 1;
        elseif data_table.Participant(n+2) == "R2"
            turns_table.Values(44)  = turns_table.Values(44) + 1;
        end
    elseif data_table.Participant(n+1) == "R1"
        turns_table.Values(11)     = turns_table.Values(11) + 1;
```

```
if data_table.Participant(n+2) == "D1"
    turns_table.Values(45) = turns_table.Values(45) + 1;
elseif data_table.Participant(n+2) == "D2"
    turns_table.Values(46) = turns_table.Values(46) + 1;
elseif data_table.Participant(n+2) == "R1"
    turns_table.Values(47) = turns_table.Values(47) + 1;
elseif data_table.Participant(n+2) == "R2"
    turns_table.Values(48) = turns_table.Values(48) + 1;
end

elseif data_table.Participant(n+1) == "R2"
    turns_table.Values(12) = turns_table.Values(12) + 1;
if data_table.Participant(n+2) == "D1"
    turns_table.Values(49) = turns_table.Values(49) + 1;
elseif data_table.Participant(n+2) == "D2"
    turns_table.Values(50) = turns_table.Values(50) + 1;
elseif data_table.Participant(n+2) == "R1"
    turns_table.Values(51) = turns_table.Values(51) + 1;
elseif data_table.Participant(n+2) == "R2"
    turns_table.Values(52) = turns_table.Values(52) + 1;
end

end

elseif data_table.Participant(n) == "R1"
if data_table.Participant(n+1) == "D1"
    turns_table.Values(13) = turns_table.Values(13) + 1;
if data_table.Participant(n+2) == "D1"
    turns_table.Values(53) = turns_table.Values(53) + 1;
elseif data_table.Participant(n+2) == "D2"
    turns_table.Values(54) = turns_table.Values(54) + 1;
```

```
elseif data_table.Participant(n+2) == "R1"
    turns_table.Values(55) = turns_table.Values(55) + 1;
elseif data_table.Participant(n+2) == "R2"
    turns_table.Values(56) = turns_table.Values(56) + 1;
end
elseif data_table.Participant(n+1) == "D2"
    turns_table.Values(14) = turns_table.Values(14) + 1;
if data_table.Participant(n+2) == "D1"
    turns_table.Values(57) = turns_table.Values(57) + 1;
elseif data_table.Participant(n+2) == "D2"
    turns_table.Values(58) = turns_table.Values(58) + 1;
elseif data_table.Participant(n+2) == "R1"
    turns_table.Values(59) = turns_table.Values(59) + 1;
elseif data_table.Participant(n+2) == "R2"
    turns_table.Values(60) = turns_table.Values(60) + 1;
end
elseif data_table.Participant(n+1) == "R1"
    turns_table.Values(15) = turns_table.Values(15) + 1;
if data_table.Participant(n+2) == "D1"
    turns_table.Values(61) = turns_table.Values(61) + 1;
elseif data_table.Participant(n+2) == "D2"
    turns_table.Values(62) = turns_table.Values(62) + 1;
elseif data_table.Participant(n+2) == "R1"
    turns_table.Values(63) = turns_table.Values(63) + 1;
elseif data_table.Participant(n+2) == "R2"
    turns_table.Values(64) = turns_table.Values(64) + 1;
end
elseif data_table.Participant(n+1) == "R2"
```

```
turns_table.Values(16)      = turns_table.Values(16) + 1;
if data_table.Participant(n+2) == "D1"
    turns_table.Values(65)    = turns_table.Values(65) + 1;
elseif data_table.Participant(n+2) == "D2"
    turns_table.Values(66)    = turns_table.Values(66) + 1;
elseif data_table.Participant(n+2) == "R1"
    turns_table.Values(67)    = turns_table.Values(67) + 1;
elseif data_table.Participant(n+2) == "R2"
    turns_table.Values(68)    = turns_table.Values(68) + 1;
end
end
elseif data_table.Participant(n) == "R2"
if data_table.Participant(n+1) == "D1"
    turns_table.Values(17)    = turns_table.Values(17) + 1;
if data_table.Participant(n+2) == "D1"
    turns_table.Values(69)    = turns_table.Values(69) + 1;
elseif data_table.Participant(n+2) == "D2"
    turns_table.Values(70)    = turns_table.Values(70) + 1;
elseif data_table.Participant(n+2) == "R1"
    turns_table.Values(71)    = turns_table.Values(71) + 1;
elseif data_table.Participant(n+2) == "R2"
    turns_table.Values(72)    = turns_table.Values(72) + 1;
end
elseif data_table.Participant(n+1) == "D2"
    turns_table.Values(18)    = turns_table.Values(18) + 1;
if data_table.Participant(n+2) == "D1"
    turns_table.Values(73)    = turns_table.Values(73) + 1;
elseif data_table.Participant(n+2) == "D2"
```

```
        turns_table.Values(74)      = turns_table.Values(74) + 1;
elseif data_table.Participant(n+2) == "R1"
        turns_table.Values(75)      = turns_table.Values(75) + 1;
elseif data_table.Participant(n+2) == "R2"
        turns_table.Values(76)      = turns_table.Values(76) + 1;
end
elseif data_table.Participant(n+1) == "R1"
        turns_table.Values(19)      = turns_table.Values(19) + 1;
if data_table.Participant(n+2)    == "D1"
        turns_table.Values(77)      = turns_table.Values(77) + 1;
elseif data_table.Participant(n+2) == "D2"
        turns_table.Values(78)      = turns_table.Values(78) + 1;
elseif data_table.Participant(n+2) == "R1"
        turns_table.Values(79)      = turns_table.Values(79) + 1;
elseif data_table.Participant(n+2) == "R2"
        turns_table.Values(80)      = turns_table.Values(80) + 1;
end
elseif data_table.Participant(n+1) == "R2"
        turns_table.Values(20)      = turns_table.Values(20) + 1;
if data_table.Participant(n+2)    == "D1"
        turns_table.Values(81)      = turns_table.Values(81) + 1;
elseif data_table.Participant(n+2) == "D2"
        turns_table.Values(82)      = turns_table.Values(82) + 1;
elseif data_table.Participant(n+2) == "R1"
        turns_table.Values(83)      = turns_table.Values(83) + 1;
elseif data_table.Participant(n+2) == "R2"
        turns_table.Values(84)      = turns_table.Values(84) + 1;
end
```

```
    end
  end
end

if data_table.Participant(length(data_table.Participant)-1) == "D1"
  if data_table.Participant(length(data_table.Participant)) == "D1"
    turns_table.Values(5) = turns_table.Values(5) + 1;
  elseif data_table.Participant(length(data_table.Participant)) == "D2"
    turns_table.Values(6) = turns_table.Values(6) + 1;
  elseif data_table.Participant(length(data_table.Participant)) == "R1"
    turns_table.Values(7) = turns_table.Values(7) + 1;
  elseif data_table.Participant(length(data_table.Participant)) == "R2"
    turns_table.Values(8) = turns_table.Values(8) + 1;
  end
elseif data_table.Participant(length(data_table.Participant)-1) == "D2"
  if data_table.Participant(length(data_table.Participant)) == "D1"
    turns_table.Values(9) = turns_table.Values(9) + 1;
  elseif data_table.Participant(length(data_table.Participant)) == "D2"
    turns_table.Values(10) = turns_table.Values(10) + 1;
  elseif data_table.Participant(length(data_table.Participant)) == "R1"
    turns_table.Values(11) = turns_table.Values(11) + 1;
  elseif data_table.Participant(length(data_table.Participant)) == "R2"
    turns_table.Values(12) = turns_table.Values(12) + 1;
  end
elseif data_table.Participant(length(data_table.Participant)-1) == "R1"
  if data_table.Participant(length(data_table.Participant)) == "D1"
    turns_table.Values(12) = turns_table.Values(13) + 1;
  elseif data_table.Participant(length(data_table.Participant)) == "D2"
```

```
    turns_table.Values(14) = turns_table.Values(14) + 1;
elseif data_table.Participant(length(data_table.Participant)) == "R1"
    turns_table.Values(15) = turns_table.Values(15) + 1;
elseif data_table.Participant(length(data_table.Participant)) == "R2"
    turns_table.Values(16) = turns_table.Values(16) + 1;
end
elseif data_table.Participant(length(data_table.Participant)-1) == "R2"
if data_table.Participant(length(data_table.Participant)) == "D1"
    turns_table.Values(17) = turns_table.Values(17) + 1;
elseif data_table.Participant(length(data_table.Participant)) == "D2"
    turns_table.Values(18) = turns_table.Values(18) + 1;
elseif data_table.Participant(length(data_table.Participant)) == "R1"
    turns_table.Values(19) = turns_table.Values(19) + 1;
elseif data_table.Participant(length(data_table.Participant)) == "R2"
    turns_table.Values(20) = turns_table.Values(20) + 1;
end
end

for n = 1:length(turns_table.Markov)
    if n <= 4
        turns_table.Markov(n) = turns_table.Values(n)/sum(turns_table.Values(1:4));
    elseif n <= 20
        turns_table.Markov(n) = turns_table.Values(n)/sum(turns_table.Values(5:20));
    else
        turns_table.Markov(n) = turns_table.Values(n)/sum(turns_table.Values(21:84));
    end
end
```

```

disp('Data analysis DONE')
%% Export
figure %cumulative duration, all participants
hold on
for n = 1:4
    plot(data{n}(:,1),cumulative_duration{n},'DisplayName',data_titles(n))
end
plot([data_separate{1}(length(data_separate{1}(:,2)),1)
      data_separate{1}(length(data_separate{1}(:,2)),1)],[0
      max(cell2mat(cumulative_duration))], 'DisplayName','Pause')
legend('location','NorthWest')
title([environment, ' ', num2str(session), ' Cumulative talking time'])
print(gcf,[environment, num2str(session), '_cumtt.png'],'-dpng')
hold off

figure %cumulative duration, all participants, normalized
hold on
for n = 1:4

    plot(data{n}(:,1)/max(data_table.End),cumulative_duration{n}/max(data_table.End),'
          DisplayName',data_titles(n))
end
plot([data_separate{1}(length(data_separate{1}(:,2)),1)
      data_separate{1}(length(data_separate{1}(:,2)),1)]/max(data_table.End),[0
      max(cell2mat(cumulative_duration)]/max(data_table.End), 'DisplayName','Pause')
legend('location','NorthWest')
title([environment, ' ', num2str(session), ' Cumulative talking time, normalized'])
print(gcf,[environment, num2str(session), '_cumttN.png'],'-dpng')
hold off

```

```
figure %cumulative duration, all participants, self-normalized
hold on
for n = 1:4

    plot(data{n}(:,1)/max(data{n}(:,2)),cumulative_duration{n}/cumulative_duration{n}(
        length(cumulative_duration{n})), 'DisplayName',data_titles(n))
end
plot([data_separate{1}(length(data_separate{1}(:,2)),1)
    data_separate{1}(length(data_separate{1}(:,2)),1)]/max(data_table.End),[0 1],
    'DisplayName','Pause')
legend('location','NorthWest')
title([environment, ' ', num2str(session), ' Cumulative talking time, self-normalized'])
print(gcf,[environment, num2str(session), '_cumttSN.png'],'-dpng')
hold off

figure %moving average, all participants
hold on
for n = 1:4

    plot(movmean(data_seconds(1:max(data_duration(n)),n),moving_window),'DisplayN
        ame',data_titles(n))
end
plot([data_separate{1}(length(data_separate{1}(:,2)),1)
    data_separate{1}(length(data_separate{1}(:,2)),1)],[0 1], 'DisplayName','Pause')
legend('location','NorthWest')
title([environment, ' ', num2str(session), ' Moving average'])
print(gcf,[environment, num2str(session), '_movavrg.png'],'-dpng')
hold off
```

```
figure %moving average, all participants, normalized
hold on
for n = 1:4

    plot(movmean(data_normalized(:,n),round(1000*moving_window/data_duration(n)))
        *1000,'DisplayName',data_titles(n))
end
plot([data_separate{1}(length(data_separate{1}(:,2)),1)
     data_separate{1}(length(data_separate{1}(:,2)),1)]/max(data_table.End)*1000,[0 1],
     'DisplayName','Pause')
legend('location','NorthWest')
title([environment, ' ', num2str(session), ' Moving average, normalized'])
print(gcf,[environment, num2str(session), '_movavrgN.png'],'-dpng')
hold off

close all
```