

TARTU ÜLIKOOL
Loodus- ja täppisteaduste valdkond
Arvutiteaduse instituut
Informaatika õppekava

Rauno Jaaska

Juturoboti loomine Delta õppehoonesse

Bakalaureusetöö (9 EAP)

Juhendaja: Mark Fišel, PhD

Tartu 2021

Juturoboti loomine Tartu Ülikooli Delta õppehoonesse

Lühikokkuvõte: Töö eesmärk on Tartu Ülikooli Delta õppehoone külastajaid abistava ingliskeelse juturoboti loomine raamistiku Rasa Open Source abil. Töös on toodud välja valdkonna hetkeseis olemasolevate tehnoloogiate ja lahenduste võtmes, antud ülevaade loodud juturobotist ning põhjendatud tehtud valikuid. Töö väljundiks on juturoboti lähtekood, mis sisaldab juturoboti treenimisandmeid, juturoboti lisakomponente, varundatud andmebaasi, andmebaasi uuendamise skripte, juturoboti tööks vajalike süsteemiteenuste konfiguratsioonifaile ja juturoboti käivitusjuhust.

Võtmesõnad: juturobot, keeletehnoloogia, Rasa Open Source

CERCS: P176 Tehisintellekt

Chatbot for the Delta Centre of the University of Tartu

Abstract: The aim of this thesis is the creation of a virtual assistant designed for visitors of the Delta Centre of the University of Tartu using the Rasa Open Source framework. The thesis contains a description of state-of-the-art technologies and existing virtual assistants, the overview of the final product, and the justifications of decisions made during development. The result of the thesis is the source code of the final product, which consists of training data and custom components of the chatbot, a backup of the database, scripts for updating the database, configuration files for the system services required to run the chatbot, and a manual for setting up the chatbot.

Keywords: chatbot, natural language understanding, Rasa Open Source

CERCS: P176 Artificial Intelligence

Sisukord

1	Sissejuhatus	4
2	Mõisted ja terminid	6
3	Taust	7
4	Metoodika	8
4.1	Rasa Open Source	8
4.2	Rasa Action Server	9
4.3	PostgreSQL	9
4.4	Nginx	10
4.5	Fomantic-UI	10
4.6	Väljajäänud tehnoloogiad	10
4.6.1	Duckling	10
4.6.2	Rasa X	11
4.6.3	Mozilla DeepSpeech	12
4.6.4	TartuNLP Neurotölge	13
5	Arhitektuur	14
6	Juturoboti tuum	16
6.1	Rasa tööpõhimõtted	16
6.2	Loodud funktsionaalsus	16
6.2.1	Koostööruumide broneerimise demo	16
6.2.2	Delta töötajate kabinetide numbrid	17
6.2.3	Kursuste sündmuste toimumisajad ja -kohad	18
6.2.4	Ootamatute sisendite käitlemine	19
7	API kasutamine	20
8	Kokkuvõte	22
Lisad		26
	Litsents	26

1 Sissejuhatus

Käesoleval sajandil on üheks arvutiteaduse olulisemaks arendusvaldkonnaks kujunenud automatiseerimine. Igas tööstusvaldkonnas tehakse kulude vähendamiseks püüdlusi inime tööjõu robotitega asendamise suunas. Juturobotite peamise kasutusala võib näha klienditeeninduse automatiseerimist teenindusvaldkonnas, kus Maailmapanga 2019. aasta andmetel töötab maailmas 50 ja Eestis 68 protsenti töötavatest inimestest [1]. Eesti ettevõtete seas kasutavad juturoboteid klienditeeninduse osana näiteks telekommunikatsioonifirmad Telia ja Elisa [2, 3]. Ühtlasi on ülemaailmse robotikavõistluse Robotex raames Tallinna linnaku Ülemiste City eestvedamisel toimunud külastajaid teenindava roboti Kratt, mille üheks komponendiks on ka külastaja vestluse mõistmine, loomise võistlus [4]. Et kasutajaid teenindavad juturobotid on niivõrd levinud, on loomulik, et ka Tartu Ülikoolil kui eesti rahvusülikoolil on oma juturobot.

2020. aasta alguses valminud Tartu Ülikooli Delta õppehoonesse on alaliselt välja pandud erinevad demod. Nende abil on nii keskuse külastajatele kui ka töötajatele võimalik tutvustada Tartu Ülikooli kompetentse arvutiteaduse erinevates alamvaldkondades ja üleüldiseid arvutiteaduse võimalusi. Arendatud juturobot on mõeldud olema Delta õppehoone demopargi loomulik osa, mis sobib nii ühekordseteks demonstratsioonideks kui ka igapäevaseks Delta keskusega ja selles toimuva õppetööga seotud informatsiooni otsinguks.

Kuna juturoboti arendus on mahukas tegevus, piirdub see töö inglise keeles vestleva juturoboti loomisega. Töö avaldamise hetkel toimub eestikeelsete keelemudelite loomine, mille põhjal on võimalik luua juturoboti eestikeelne analoog. Kaks juturobotit on seejärel võimalik ühendada ning seada näiteks Delta õppehoonesse paigaldatud statsionaarse terminali kaudu kasutatavaks keskuse külastajatele ja töötajatele nii eesti kui ka inglise keeles. Kombineeritud juturobot, edaspidi Deltabot, saab kasutajaid aidata erinevate Delta keskusega seonduvate küsimustega. Võimalikud päringud võivad olla nii üldharivad, näiteks Delta keskuse ajaloo või selles tegutsevate institutsioonide tutvustamine, kui ka ülimalt spetsiifilised, näiteks küsimused mõne õppejõu kabineti numbri või mõne loengu toimumisaja kohta.

Töö koosneb viiest peatükist. Esimeses peatükis on toodud välja juturobotite valdkonna hetkeseisu kirjeldus nii Eestis kui ka ülemaailmselt. Teine peatükk koosneb kasutatud tehnoloogiate kirjeldustest ja tehtud valikute põhjendustest; peatüki teine pool sisaldab tehnoloogiaid, mis olid arenduse käigus kasutusel, kuid jäid lõpptulemusest peatükis kirjeldatud põhjustel välja. Kolmandas peatükis on antud juturobotit toetavate komponentide ülevaade ning eraldiseisvate komponentide omavahelise suhtluse skeem. Neljas peatükk kirjeldab loodud juturoboti tööpõhimõtteid, arendatud funktsionaalsust ja spetsiaalselt Deltaboti tarbeks loodud lisakomponente. Viimane peatükk kirjeldab juturobotile päringute tegemise üksikasju ning turvalisust.

Töö kirjalikku osa saadab arendatud juturoboti lähtekood. Lähtekood koosneb töös loodud tarkvaralahendustest, milleks on juturoboti treenimisandmed, juturoboti lisakom-

ponendid keeletuvastuseks, varundatud andmebaas, andmebaasi uuendamise skriptid ja juturoboti tööks vajalike süsteemiteenuste konfiguratsioonifailid. Nendele lisaks on lähtekoodiga kaasas juturoboti käivitusjuhised. Lähtekood on avalikult kättesaadav aadressil <https://github.com/Aeryan/Delta>.

2 Mõisted ja terminid

API (ingl Application Programming Interface) - reeglistik, mis määrab programmi ja muu tarkvara vahelise suhtluse korra [5].

ATI sülearvuti Tartu Ülikooli arvutiteaduse instituudi poolt tudengitele laenutatavad Intel i5-8650U protsessoriga sülearvutid [6].

Avatud lähtekood (ingl open source) - tarkvara, mille lähtekoodi on autor lubanud kasutada ja muuta [7].

Kavatsus (ingl intent) - "kasutaja sõnumi taga peituv mõte või eesmärk"¹ [8].

Küsimustik (ingl form) - "toiming, mis pärib kasutajalt ettenähtud olemite väärtusi"² [8].

Levenshteini kaugus - ühe sõne teiseks sõneks muundamiseks vajalik minimaalne tähtede kustutamiste, lisamiste ja asenduste arv [9].

Moodul (ingl module) - programmeerimiskeele Python kontekstis fail, mis sisaldab samas keeles kirjutatud definitsioone ja käske, ning mida saab kasutada teiste keeles Python kirjutatud skriptide osana [10].

Olem (ingl entity) - "muutuja, mis hoiustab vestluse käigus kasutaja sõnumitest eraldatavate võtmesõnu"³ [8].

Pöördproksiserver (ingl reverse proxy) - välisvõrgus paikneva kliendi ja sisevõrgus paikneva serveri suhtlust vahendav server [11].

Raamistik (ingl framework) - "loodavat platvormi, tarkvara, riistvara, protokollid vms toetav struktuur" [7].

Systemd - "süsteemi ja selle teenuste haldur Linuxile" [12].

Toiming (ingl action) - "vestluse käigus üks juturoboti poolt sooritatav tegevus"⁴ [8].

Väärtusväli (ingl slot) - "muutuja, millega hoiustatakse vestluse käigus teavet"⁵ [8].

¹Originaalis "In a given user message, the thing that a user is trying to convey or accomplish"

²Originaalis "A type of custom action that asks the user for multiple pieces of information"

³Originaalis "Keywords that can be extracted from a user message"

⁴Originaalis "A single step that a bot takes in a conversation"

⁵Originaalis "A key-value store that Rasa uses to track information over the course of a conversation"

3 Taust

Juturobotid jaotatakse eesmärgipõhiselt kasutajatega üldist vestlust pidavaiks ning ülesandepõhisteks ehk kindlaid ülesandeid täitvateks [13]. Et hetkeseisuga pole arvutid võimelised inimeste loomulikku keelt täies ulatuses omandama, keskendutakse neist kahest kategooriast juturobotite arendamisel viimasele. Seeläbi leidub hulgaliselt ülesandepõhiste juturobotite loomise tarkvara, näiteks Dialogflow, Wit.ai, Bot Framework ja Watson, mille arendajaiks on vastavalt tarkvarahiid Google, Facebook, Microsoft ja IBM. Laialdast kasutust leidvad personaalsed assistendid nagu Siri ja Alexa kuuluvad definitsiooni järgi samuti ülesandepõhiste juturobotite kategooriasse - kuigi nende võimekuse piirid on laiad, täidavad nad olemuslikult siiski kindlaid ülesandeid, näiteks äratuse määramine või internetiallikatest Emumäe kõrguse leidmine.

Ka Eestis on viimastel aastatel ülesandepõhiseid juturoboteid kasutusele võtnud mitmed organisatsioonid. Eraettevõtetest kasutavad juturoboteid klienditeeninduse osana näiteks Telia ja LHV, mis kasutavad ettevõtte Alphablues arendatud juturoboteid, ning Elisa ja Swedbank, mis kasutavad ettevõtte Mindtitan loodud juturoboteid [14, 15]. Riigiasutuste seas on hetkel avalikkusele ligipääsetavad Statistikaameti juturobot Iti, mis on leitav ameti kodulehel, ja Päästeameti juturobot, mida on võimalik kasutada sotsiaalvõrgustiku Facebook suhtluskeskkonnas Messenger [16, 17]. Ühtlasi arendati Politsei- ja Piirivalveameti ning Riigi Infosüsteemi Ameti koostööl välja riigiasutuste ülese juturoboti prototüüp, mis suudab vastata kasutajate asutusespetsiifilistele küsimustele kasutatavast keskkonnast sõltumata, juhtides vajadusel kasutaja päringule vastust andva ametini [18, 19].

4 Metoodika

Selles peatükis toon välja kasutatud tehnoloogiad ning põhjendan nende valikut.

4.1 Rasa Open Source

Juturoboti loomise esimene ja tõenäoliselt olulisim samm on õige raamistiku valimine. Erinevaid raamistikke saab võrrelda mitmest aspektist.

Esmase filtreerimise tegin eeldatava kasutuskeskkonna põhjal. Kuna loodud juturobot peab olema kättesaadav füüsilise terminali, veebilehe ja eraldiseisva rakenduse kaudu, pole mõistlik valida kindla rakenduse, näiteks Facebook Messenger, tarbeks mõeldud juturobotite loomise raamistikke.

Teise tegurina kaalusin, kuidas on mõistlik juturobotit majutada. Kuna arvutiteaduse instituudil on võimekus vajalikke teenuseid ise majutada, on otstarbekas vaadata juturobotite poole, mis on kasutatavad lokaalselt. Kui kasutada pilveteenusena pakutavat juturobotit, puudub instituudil ülevaade, mida saadetud teabega tegelikult tehakse. Majutades juturobotit oma serverites, toimub kogu sõnumivahetus ja andmetöötlus instituudi järelevalve all ning kasutajate andmetega ei saa ülikooli teadmata midagi juhtuda.

Kolmandaks on oluline lähtekoodi avatus. Avatud lähtekoodiga tarkvara tegevus on läbipaistev. Kui tegu pole avatud lähtekoodiga tarkvaraga, võib loodav rakendus töö käigus sooritada soovimatuid dokumenteerimata tegevusi, näiteks süsteemi ressursse kasutades krüptoraha kaevandamine või kättesaadavate andmeid kolmandatele isikutele edastamine. Avatud lähtekoodiga tarkvara korral on aga kõik komponendid läbivaatusele avatud, mistõttu väärtegevus tarkvarast puudub või on ülimalt hästi peidetud. Kuna viimase saavutamine on laialdase kasutuse juures ülimalt keeruline, pärsib lähtekoodi avatus mingil määral laialt kasutatud tarkvara korral soovimatute tegevuste olemasolu.

Avatud lähtekoodi peamine kasu tuleneb aga võimalusest tarkvara muuta. Kasutades avatud lähtekoodiga raamistikke, on võimalik vajaduse tekkides komponente ümber kirjutada, mis läbi saab juturobotile luua ametlikust versioonist puuduvat funktsionaalsust.

Neljandaks mängis raamistiku valikul rolli tarkvara mitmekeelsus. Kuna Delta õppehoone kuulub eesti rahvusülikooli alla, ei piisa vaid inglise keelt kõneleva juturoboti loomisest. Sobiv raamistik peab võimaldama ka eesti- ja venekeelse juturoboti arendust. Kuna eesti keele näol on tegu võrdlemisi vähekasutatava keelega ning seetõttu on juturobotite loomise raamistike arendajatepoolne huvi eesti keele toe vastu väike, on see tingimus samaväärne sihtkeelest sõltumata juturoboti loomise võimaldamisega. Seega tuleb kasutada raamistikku, mis ei ole keelespetsiifiline.

Viimaseks võtsin arvesse juhendaja arvamust ja olemasolevate süsteemide loomisel tehtud valikuid. Kuigi minu juhendajal puudus juturobotite loomise kogemus, soovitas ta uurida Rasa sobivust.

Varasemalt loodud juturobotite uurimisel keskendusin Eestis avaliku sektori kasutuses olevate juturobotite tausta uurimisele. Päästeametiga suheldes sain teada, et nende

juturobot on loodud raamistikku ManyChat kasutades. ManyChat on mõeldud keskkonnas Facebook Messenger kasutatavate juturobotite loomiseks [20]. Kindla rakendusega piiratud juturoboti loomise välistasin ma juba varasemalt, seega välistasin raamistikku ManyChat kasutamise.

Suhtlusest Statistikaametiga selgus, et ameti juturoboti Iti on loonud ettevõtte AlphaBlues. AlphaBlues on oma kodulehe sõnul loonud juturobotid ka Eestis tegutsevatele ettevõtetele Telia, Tallink ja LHV Pank, kasutades ettevõttesiseselt välja arendatud tarkvara [14]. Et see tarkvara pole avalikkusele kättesaadav, pidin sobivat raamistikku otsima mujalt.

Politsei- ja Piirivalveameti ning Riigi Infosüsteemide Ameti koostöös loodava juturoboti lähtekoodi uurides ilmneb, et juturoboti arendamiseks on samuti kasutatud raamistikku Rasa [21]. Et Rasa Open Source vastas kõigile minu poolt seatud kriteeriumitele, seda soovitas minu juhendaja ning see on sobiv ka Eesti riigiasutustele, otsustasin Deltaboti raamistikku valikul selle kasuks.

4.2 Rasa Action Server

Raamistikku Rasa Open Source üheks komponendiks on juturobotist eraldi käitav server Rasa Action Server. See võimaldab Rasa Open Source juturobotitel piiranguteta kasutada programmeerimiskeeles Python kirjutatud käske juturoboti töö loomuliku osana [22]. Deltabot kasutab seda serverit kasutajale vajaliku informatsiooni andmebaasist pärimiseks.

4.3 PostgreSQL

Deltabot vajab oma ülesannete täitmiseks teavet, mis on Tartu Ülikooli hallatavates veebikeskkondades avalikult kättesaadav. Kasutaja päringu järgselt algallikast vajaliku informatsiooni pärimine on sobilik, kui teave jõuab kasutajani talle vastuvõetava aja jooksul. Vajalik teave pole päringute kaudu otseselt kättesaadav - teabe leidmine vajab enam kui üht veebipäringut ning nendega saadud andmete töötlust. See võtab kokku enam kui paar-kolm sekundit, mida võib pidada keskmise kasutaja juures teabe hankimise ülempiiriks. Seetõttu peab vajalik teave olema juturobotile kättesaadavamal kujul.

Andmehanke kiirendamiseks toetub juturobot andmebaasile, milles sisaldub vajaliku teabe koopia, ja skriptidele, mis ammutavad avalikest allikatest vajaliku teabe ja salvestavad selle andmebaasi. Deltabot ja sellele vajalik teave ei määra samas ainuõiget andmebaasilahendust ning valitud süsteem ei mõjuta suures pildis juturobotit ennast. Kuna mul on andmebaasihaldussüsteemiga PostgreSQL varasemalt kokku puutudes tekkinud selle süsteemi kasutamisel mõningane vilumus, otsustasin juturoboti arendamisel kasutada just seda.

4.4 Nginx

Juturoboti laiemale kasutajaskonnale avamiseks on vajalik mugav kasutajaliides. Selleks eraldi rakenduse loomine on võrdlemisi mahukas, seega otsustasin luua veebilehe, mis vahendab vestlust kasutaja ja juturoboti vahel. Selle veebilehe kasutajatele serverimiseks on vaja veebiserverit. Teisalt on veebiserverit vaja ka pöördproksiserveriks juturobotit majutavast serverist väljaspool olevate rakenduste ja juturoboti vahele, millest on pikemalt juttu seitsmendas peatükis.

Veebiserverite seas on enim kasutuses Apache ja Nginx [23]. Varasemalt on mul kogemusi mõlema kasutamiseega. Töö Nginx veebiserveritega on olnud ladus, kuid Apache kasutamine on olnud konarlik ning alati pole soovitud tulemuseni jõudmine õnnestunud. Kuna veebiserveri valik ei mõjuta suures pildis juturobotit ennast, otsustasin kasutada Nginx veebiserverit.

4.5 Fomantic-UI

Kasutajatele serveritava veebilehe loomisel otsustasin arenduse kiiruse huvides kasutada raamistikku Fomantic-UI, millega toimus veebilehe loomine laialdase varasema kasutuskogemuse tõttu palju kiiremini ja ladusamalt kui ainult programmeerimiskeelt JavaScript kasutades. Fomantic-UI on "raamistik, mis lihtsustab atraktiivsete ja adaptiivsete veebilahenduste loomist" [24].

4.6 Väljajäänud tehnoloogiad

Juturoboti arengust täieliku ülevaate andmise eesmärgil toon siinkohal välja tarkvarakomponendid, mis ei leidnud lõpplahenduses kasutust, kuid olid arenduse käigus kasutuses või uurimise all.

4.6.1 Duckling

Raamistiku Rasa Open Source baasil loodavad juturobotid vajavad kasutaja sisendist teabe ammutamiseks lisakomponente. Deltaboti arenduse algfaasis loodud Delta koostööruumide broneerimise demo vajas üht sellist komponenti, et lugeda kasutaja sisestatud sõnumist välja inimeste arvu, mida kasutajale vajalik koostööruum peab mahutama. Tekstis numbrite tuvastamist toetab komponent nimega Duckling [25]. Duckling toimis numbrite tuvastusel ootuspäraselt, tuvastades nii araabia numbritega kui ka sõnakujul esitatud arvud. Broneerimissüsteem jäi aga juturobotist välja, mistõttu pole Duckling juturoboti lõpliku arhitektuuri osa.

4.6.2 Rasa X

Rasa X on tarkvara, mis lihtsustab pideva kasutajate tagasiside abil raamistikuga Rasa Open Source loodavate juturobotite arendust [26]. Rasa X lihtsustab juturobotite arendust, pakkudes graafilist keskkonda, mille kaudu saavad kasutajad juturobotit katsetada ning arendajad toimunud vestlustes sõnumeid märgendada ja juturobotite valikuid korrigeerida [27].

Kuigi Rasa X lihtsustab arendusprotsessi, on sellel mõned puudujäägid. Esimene mõttekoht on tõsiasi, et erinevalt kõigist teistest kasutatust leidnud tehnoloogiatest pole Rasa X vaba lähtekoodiga tarkvara, vaid tasuta kinnise lähtekoodiga tarkvara [28]. See fakt üksinda pole samas ka täielikult vabavaralise roboti loomisel takistuseks - tegelikkuses muudab Rasa X arendatavat juturobotit inimarendajaga samaväärselt läbi seadistusfailide. Tarkvara eesmärk on protsessi lihtsustamine, mistõttu võib arenduse lõppemisel Rasa X tarkvara eemaldada ning juturoboti olemus sellest ei muutu. Üldpildis ei saanud Rasa X kasutamise otsustamisel lähtekoodi avatus määravaks.

Teise murekohana ilmnes, et arenduseks kasutatavas ATI sülearvutis hangus Rasa X peale mõneminutilist kasutust. Samas esinesid võimekusprobleemid muu tarkvara, näiteks Mozilla kõnetuvastuse, kasutamisel. See probleem on lahendatav arenduseks kasutatava riistvara vahetamisega ning ei ole ületamatu.

Suurim murekoht Rasa X kasutamisel oli juturoboti vastuste erinevus, sõltuvalt sellest, kas juturobotit käitis Rasa X või Rasa Open Source. Täpsemalt ilmnes tarkvaraga Rasa X kasutamisel järgnev olukord. Juturobot sai kasutajalt sõnumi, milles kasutaja päris Delta töötaja kontori numbrit. Juturobot mõistis sisendit korrektselt, tuvastas tekstis õige töötajanime ning edastas kogutud teabe korrektselt Rasa Action Serverile. Rasa Action Server võttis teabe vastu ja sooritas andmebaasipäringu otsitava kontorinumbriga kohta. Probleem tekkis otsitava info väljastamisel - Rasa Action Server sai andmebaasist kontorinumbriga täisarvulise väärtuse ning pidi väärtuse muundama sõnekujule, tulemuse liitma kahele arvu ümbritsevale sõnele ja väljastama saadu kasutajale. Tarkvaraga Rasa X käivitatud juturobotis läks aga arv tagastatavast sõnest teadmata põhjusel kaduma ning kasutajale väljastati ainult arvu ümbritsevatest sõnedest koosnev sõne. Kui juturobot oli käivitatud tarkvara Rasa Open Source abil, kusjuures juturoboti mudelit ja Rasa Action Serverit polnud katsete vahepeal muudetud, väljastas juturobot vaatlusaluse teksti ootuspäraselt. Ka ei olnud viga ühekordne - see avaldus igal katsel ka peale serverite taaskäivitust, arenduseks kasutatava sülearvuti taaskäivitust ning mitme päeva möödumist. Katsed nii omal käel veale jälile saada kui ka Rasa dokumentatsioonist ja kasutajatefoorumist selgitust otsida olid viljatud. Kuna juturoboti korrektne töötamine on ülioluline; antud viga ilmnes siis ja ainult siis, kui juturoboti käitajaks oli Rasa X ning Rasa X pole juturoboti tööks kriitiliselt vajalik komponent, otsustasin selle ja teiste eelpool kirjeldatud probleemsete kohtade koosmõjul jätkata arendusega tarkvara Rasa X kasutamata.

4.6.3 Mozilla DeepSpeech

Et tänapäeval teada-tuntud assistendid nagu Siri ja Alexa kasutavad kõnetuvastust, tegin katsetusi ka Deltaboti komponentide sekka kõnetuvastuse lisamisega. Katsetuste käigus mõistsin, et kõnetuvastuse lisamine vajab juturoboti kasutuskeskkonnaga arvestamist. Kui juturobot on tulevikus ligipääsetav läbi statsionaarse terminali, tuleb arvestada terminali asukoha eripäradega. Näiteks, kui terminal asub raamatukogu lugemissaalis, pole häälkäskluste võimaldamine soovitatav. Kui juturobot on tulevikus kätte saadav läbi veebikeskkonna või (mobiili)rakenduse, peab lisaks kõnetuvastusele olema kindlasti võimalik ka tekstikujul päringute sisestamine, sest kasutaja võib olla keskkonnas, kus häälkäskluste andmine on taunitav, näiteks loengu- või raamatukogu lugemissaal. Teisalt võib olla tegu keskkonnaga, kus häälkäskluste andmine ei ole mõõduka või tugeva taustamüra tõttu efektiivne, näiteks ühistransport või kohvik.

Häältuvastuse võimalikkuse kaardistamiseks katsetasin Mozilla kõnetuvastustarkvara DeepSpeech. Mozilla DeepSpeech sisaldab valmistreenitud ingliskeelseid mudeleid ja on käitav programmeerimiskeele Python mooduli abil [29]. Need on olulised, sest mudelite puudumisel peaksin sobivad mudelid treenima ise, mis pole mahukuse tõttu selle töö sees mõeldav, ning programmeerimiskeele Python moodul võimaldab juturoboti arendamisel piirduda ainult ühe programmeerimiskeele kasutamisega. Tarkvara katsetamisel ATI sülearvutis toimus kõne muundamine tekstiks aeglaselt ning töötlemise hiline mine oli kumulatiivne, s.t mida kauem kõnetuvastusprogramm töötas, seda aeglasemalt väljastas programm teksti. Protsessorit AMD Ryzen 5 3600 kasutaval arvutil väljastas sama programm tulemusi praktiliselt reaalajas ning kümne minuti jooksul aeglustumist ei ilmnenu d. Seega on programmeerimiskeele Python moodulit kasutades reaalajas kõnetuvastus kättesaadav.

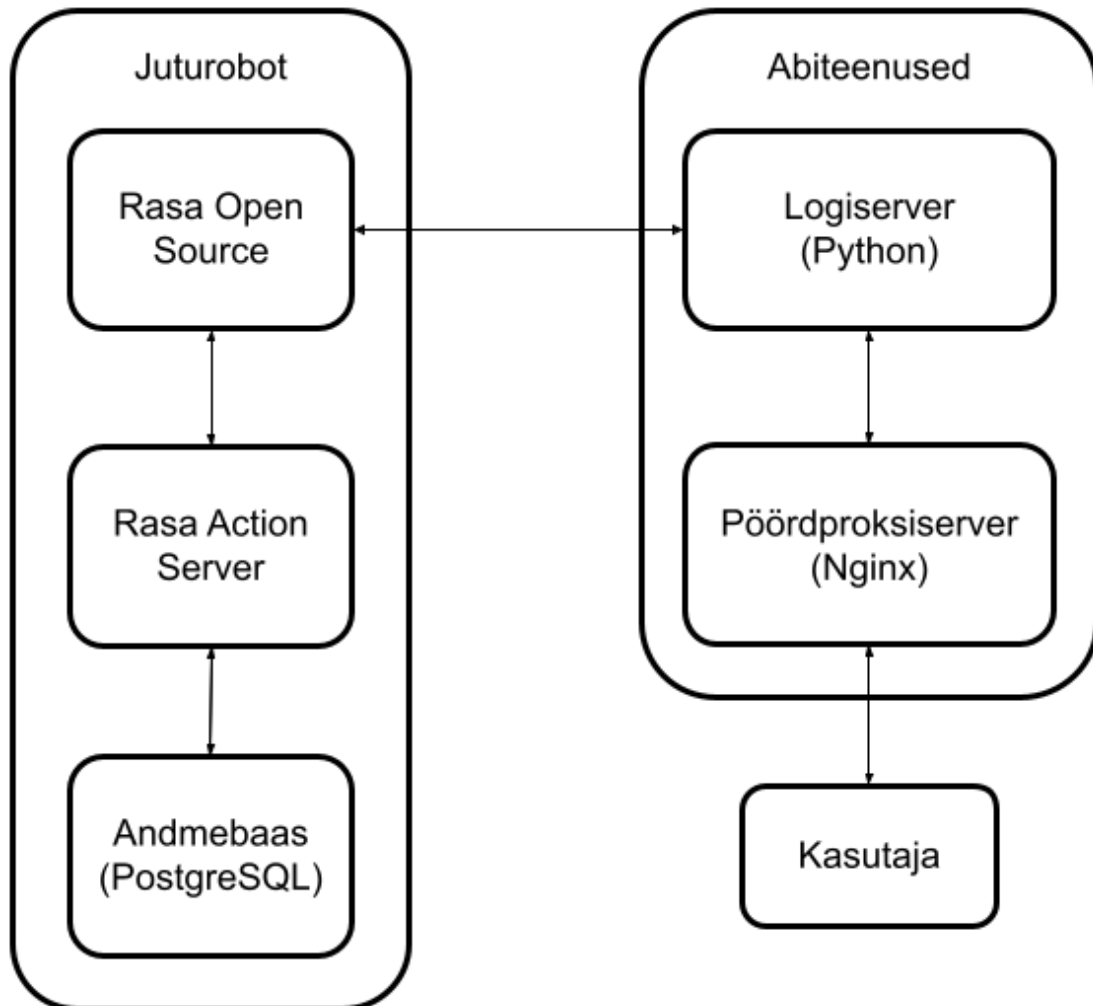
Deltabotile töötajate nimedele vastavate kontorinumbrite tagastamise arendamisel sai selgeks, et kõnetuvastus pole selle töö raames võimalik. Delta hoone õppejõudude nimede taust on ülimalt mitmekesine ning sellest tulenev probleem esineb juba eesti päritolu nimede olemasolul. Nimelt on vähetõenäoline, et väljaspool Eestit loodavad ingliskeelse kõne tuvastamise mudelid on loodud piisaval hulgal eesti päritolu nimesid sisaldavatel andmetel, et võimaldada eesti nimede korrektset tuvastust. Eestis keskendutakse aga eestikeelsete kõnetuvastusmudelite arendamisele, mis ei võimalda ingliskeelse kõne tuvastust. Arvestades tõsiasja, et lisaks eesti päritolu nimedele peab kõnetuvastusmudel suutma tuvastada ka näiteks saksa ja slaavi päritolu nimesid ning nõnda mitmekesiseid ingliskeelseid keeleandmeid tõenäoliselt ei eksisteeri, ei ole selles töös võimalik loodavale juturobotile kõnetuvastust lisada.

4.6.4 TartuNLP Neurotõlge

Juhendaja soovitusel katsetasin juturoboti töö käigus tõlketarkvara kasutamist. Katse eesmärk oli teha kindlaks, kas tõlketarkvara abil on võimalik ingliskeelset juturobotit kasutada teistes keeltes, näiteks eesti keeles, tõlkides muukeelne sisend juturobotile mõistetavaks ingliskeelseks ning juturoboti vastus tagasi muukeelseks. Tõlketarkvarana kasutasin Tartu Ülikooli keeletehnoloogia uurimisrühma arendatud tõlketööriista, millega ühendumine toimus tõlketööriista API kaudu [30, 31].

Juturoboti kasutamine tõlketarkvara vahendusel toimus eestikeelse sisendi korral üllatavalt hästi. Juturobot mõistis tõlgitud sisendeid ning tagastatud sõnade tõlge oli mõistetav, kuid mitte täiuslik. Töötajate kontorite pärimisel tuvastas juturobot korrektse töötajanime ning tagastas vastava ruuminumbri. Kursuste tuvastamisel tekkis probleem: kuna kursuste pealkirjad on spetsiifilised sõned, peab tõlketööriist olema võimeline neid täpselt tõlkima. See pole lihtne ülesanne. Näiteks matemaatika ja statistika instituudi kursuse "Kõrgem matemaatika I" tõlkeks pakub tööriist "Higher mathematics I", kuid kursuse tegelik ingliskeelne pealkiri on "Calculus I". Seetõttu pole üldiste tõlketööriistade kasutamine Deltaboti mitmekeelseks muutmiseks mõeldav. Et Tartu Ülikooli kursustel treenitud tõlketööriista ei eksisteerinud, ei ole Deltaboti tõlketarkvara vahendusel kasutamine töö kirjutamise hetkel võimalik.

5 Arhitektuur



Joonis 1. Deltaboti arhitektuur.

Deltabot koosneb arhitektuuriliselt viiest serverist. Roboti tuum on Rasa Open Source, mille töö on mõista sisendteksti ning tagastada korrektset väljundit. Raamistiku lahutamatu osa on Rasa Action Server, mis võimaldab kasutada programmeerimiskeeles Python kirjutatud programme juturoboti loomuliku osana. Üks näide sellest on tekstist eraldatud töötajanimede kasutamine PostgreSQL andmebaasist nimedele vastavate kontorinumbrite pärimiseks. Need komponendid moodustavad roboti põhiosa. Et Rasa Open Source ja Rasa Action Server töötaksid ilma avatud kasutajaliideseta, on lähtekoodis antud systemd abil käitavate teenuste konfiguratsioonide mallid failide *rasa-core.service* ja

rasa-actions.service kujul. Systemd poolt hallatava teenusena jääb ssh-sessiooni kaudu käivitatud juturobot peale sessiooni sulgemist tööle kuni vastavate teenuste või masina töö lõpetamiseni.

Kirjeldatud arhitektuur on küll piisav roboti kasutamiseks käsureal, kuid tavakasutajale mugavama kasutajaliidese pakkumist see ei võimalda. Selleks kirjutasin Fomantic-UI abil HTML-faili, mida serveerib kasutajatele Nginx veebiserver. Ühtlasi käitub see veebiserver ka ka proksiserverina, vahendades suhtlust kliendirakenduse, milleks on siinkohal eelnevalt mainitud veebileht, ja juturoboti vahel. Nginx veebiserveri kasutamiseks on vajalik vastav konfiguratsioonifail, mis on lisatud juturoboti lähtekoodile nime *rasa.conf* all.

Eelnevale lisaks kirjutasin valikulise komponendina keeles Python skripti, mis salvestab veebiserveri ja juturoboti vahel liikuvad sõnumid tekstifailidesse. Kuigi see pole juturoboti tööks otseselt vajalik, võimaldab vestluste salvestamine nende analüüsi vestluste toimumisajast sõltumata. Vestluste analüüs on väärtuslik kahel põhjusel. Peamiselt võimaldab analüüs leida vigu juturoboti konfiguratsioonis, näidates täpselt, millised vestlusteed viivad ebakorrekse väljundini. Teisalt annab kasutajate päringute uurimine aimu, millist funktsionaalsust kasutajad juturobotilt ootavad. Seega on kasutajate peetud vestluste sisu salvestamine neist info ammutamiseks igati mõistlik. Analoogselt eelpool kirjeldatud teenustele on logiserveri systemd teenuse mall antud lähtekoodi failis *rasa-logger.service*.

6 Juturoboti tuum

6.1 Rasa tööpõhimõtted

Deltaboti kui raamistikul Rasa Open Source põhineva juturoboti tegevus on kirjeldatud viie sätetefailiga.

- *config.yml* määrab komponendid, mis töötlevad sissetulevaid sõnumeid või valivad sisendile õiged vastused; sõnumeid töötlevad komponendid on näiteks klassifikaatorid ehk kavatsuse määrajad ja sõnumitest olemitele vastavate väärtuste eraldajad [32].
- *nlu.yml* toob välja kõikvõimalikud kavatsused ja kirjeldab nendele vastavaid sõnumeid [33].
- *domain.yml* sisaldab vestlussessioonide seadistust ning kirjeldab, millistest kavatsustest, olemitest, väärtusväljadest, küsimustikest ja toimingutest on juturobot teadlik [34].
- *rules.yml* ja *stories.yml* kirjeldavad juturobotiga peetavate vestluste kulgu läbi reeglite ja lugude, kusjuures reeglid on mõeldud lühikeste ja üheselt läbitavate vestluste tarbeks ning lood on mõeldud pikemate vestluste paindlikuks läbimiseks [35, 36].

6.2 Loodud funktsionaalsus

Töö algstaadiumis kaardistasin juhendajaga Deltaboti potentsiaalsed kompetentsid, milleks olid koostööruumide broneerimine, majajuhiste jagamine, õppejõudude kabinetinumbrite väljastamine ning loengute, praktikumide jms. toimumisaegade ja -paikade loetlemine. Lõpliku lahenduse said neist neljast viimased kaks. Järgnevalt kirjeldan loodud funktsionaalsusi nende arendamise järjekorras.

6.2.1 Koostööruumide broneerimise demo

Delta õppehoone teisel korrusel asuvad tudengite koostöö soodustamiseks mõeldud koosoleku- ja koostööruumid, kuid töö kirjutamise hetkel puudub nende broneerimiseks automaatne süsteem [37]. 2021 jaanuaris toimunud DeltaX võistluse raames sai DeltaQR meeskond valmis broneerimissüsteemi prototüübi, kuid reaalsesse kasutusse pole loodud lahendus jõudnud. Et pole kindel, kas ja mis kujul oleks juturobotisse mõistlik Delta ruumide broneerimist lisada, jäi loodud demo lõpuni välja arendamata, kuid demo loomisel saadud teadmised raamistiku Rasa Open Source kasutamise osas lihtsustasid ja kiirendasid juturoboti edasist arendust.

6.2.2 Delta töötajate kabinettide numbrid

Delta töötajate kabinetinumbrite jagamisel kerkis esimese probleemina esile ajakohaste andmete hankimine. Juhendaja soovitusel otsustasin allikana kasutada arvutiteaduse instituudi ning matemaatika ja statistika instituudi veebilehti, kus on välja toodud nii instituudi töötajad kui ka nende kontorite numbrid. Selleks on juturoboti lisakomponentide hulgas veebilehtede töötlemise skript *ut_employee_parser.py*, mis salvestab Delta töötajate nimed ning olemasolevad kontorinumbrid juturoboti andmebaasi. Juturoboti ülesseadmisel saab selle skripti määrata perioodiliselt (näiteks kord nädalas) käivituvaks, mis tagab andmete värskuse.

Kui kasutaja pärib Delta töötaja kohta, tuleb kontorinumbri leidmiseks teha andmebaasi päring töötaja nimega. Selleks on vaja kasutajale huvipakkuva töötaja nimi eraldada seda ümbritsevast sõnumist. Rasa dokumentatsioon toob välja erinevaid andmete ammutajaid, mille seas leidub ka isikunimede eraldajaid [32]. Need komponendid ei ole aga efektiivsed näiteks eesti nimede tuvastamisel, sest need toetuvad väärtuste tuvastamisel masinõppe mudelitele ning levinud ingliskeelsete mudelite treenimisel kasutatud treeningandmed ei sisalda piisaval hulgal eesti nimesid.

Väärtuste tuvastamiseks on Rasa dokumentatsioonis toodud välja kolm lahendust: masinõppemudelid, regulaaravaldised ning andmetabelid [33]. Sobiva masinõppe mudeli treenimine on liiga mahukas, et teha seda selle töö raames. Regulaaravaldiste kasutamine nimede eraldamiseks ei ole hea lahendus, sest isikunimed ei erine teistest sõnadest struktuuri poolest piisavalt. Andmetabelid, mis tuleb ettevalmistavalt koostada kõigist võimalikest väärtustest, on samas hea lahendus, sest ainult Delta töötajate nimede tuvastamine on antud ülesande juures piisav. Seetõttu toimub sisendtekstist Delta töötajate nimede tuvastamine andmetabeli abil, mille loob skript *ut_employee_parser.py*.

Andmetabeli ja komponendi `RegexEntityExtractor` abil teksti isikunimeks sobivuse kontroll viib vastavusse ainult teksti, mis on vastega võrdne. Ainsa erinevusena lubab komponent suurtähtede asendust väiketähtedega ja vastupidi [33]. Reaalses olukorras võib kasutaja aga teha trükivigu. Teisalt võib nimede puhul juhtuda, et kasutaja ei tea teda huvitava nime korrektset kirja pilti. Erisuse põhjustajateks võivad olla näiteks nime vöörus ja inimeste komme piirduda pikemate nimede korral kahe nime kasutamisega, mis puhul võib kasutajal puududa kokkupuude korrektse kirja pildiga. Adekvaatne juturobot peaks olema võimeline tuvastama nimesid ka juhul, kus kirja pilt ei ole täiesti korrektne.

Kirjeldatud probleemide ületamiseks kirjutasin komponendi `RegexEntityExtractor` põhjal komponendi `EmployeeExtractor`. See leiab programmeerimiskeele Python mooduli `FuzzyWuzzy` abil sisendi ja iga andmetabelis kirjeldatud väärtuse täpsuse hinnangu, mis põhineb Levenshteini kaugusel. Kui vähima kaugusega sisendi ja eeldatava väärtuse paar püsib seadistatud lävendi piires, tagastab komponent tekstis tuvastatud andmetabelis kirjeldatud väärtuse. Lävendi väärtuse 80% valisin empiirilisel katsetuste tulemuste põhjal.

Et töötaja kontorinumbri leidmiseks on vaja ainult töötaja nime, koosneb kontori-

numbri pärimise vestlustee kasutaja sisendist ja juturoboti vastusest ning vestlustee on juturoboti seadistustes kirjeldatud reeglite abil. Kavatsus *request_employee_office* algatab reegli *office_search_initialisation* läbimise, mis lõpeb toiminguga *office_search* väljakutsega. Toiming *office_search* kutsub töötajale määratud kontori olemasolule vastavalt välja kas vastuse *utter_office_result* või vastuse *utter_office_no_result* väljastamise. Need kutsuvad omakorda välja kas reegli *office_result_print_followup* või reegli *office_no_result_print_followup* täitmise, mis tühjendavad vestluse käigus täidetud olemite väärtused.

6.2.3 Kursuste sündmuste toimumisajad ja -kohad

Kursuste sündmuste allikaks on Tartu Ülikooli õppeinfosüsteemi API. API abil leiab skript *ut_course_parser.py* kursuste sündmuste teabe ning salvestab selle andmebaasi. Et õppeinfosüsteemis on sündmuste toimumisnädalad kirjeldatud Tartu Ülikooli õppenädalate põhjal, täidab skript *ut_week_generator.py* õppenädalate tabeli; õppenädalate tabel võimaldab vestluse toimumise kuupäeva ja sündmuste toimumise nädalapäeva võrdlemist ja seeläbi korrektse nädala plaani kuvamist. Analoogselt eelmises alapeatükis toodud skriptile *ut_employee_parser.py* on ka siin toodud skriptid mõeldud perioodiliseks käivitamiseks. Kursuste toimumiste värskendust võiks sooritada ideaalis kord päevas, kuid mitte vähem kui kord nädalas - juturobot ei tohi olla teadmatutes kursuste toimumiste õppeinfosüsteemi kantud erakorralistest muudatustest. Ülikooli õppenädalate genereerimist pole vaja käivitada enam kui kord õppeaastas, sest õppeaasta õppenädalateks jaotumine pole muutuva iseloomuga.

Et kursuse sündmuse päring ei tagastaks liigselt informatsiooni, piirduvad päringud käesoleva nädala sündmustega ning kasutaja täpsustatud sündmuste liigiga. Seega peab kasutaja täitma kahe olemi sisu: kursuse nimi ja sündmuse liik. Kui need sisalduvad kasutaja alguses sisendis ja juturobot suudab need eraldada, väljastab juturobot leitud info. Kui ühe või mõlema olemi sisu jääb eraldamata, peab juturobot paluma kasutajalt nende täpsustamist. Puuduoleva teabe üleküsimiseks kasutan selleks ettenähtud küsimustikku, mis omakorda kehtestab loo, mitte reegli, kasutamise.

Analoogselt töötajate isikunimede sisendist eraldamisega tuleb kursuste teabe saamiseks eraldada sisendist kursuse nimi ja sündmuse liik. Tartu Ülikooli kursuste nimede ja sündmuste liikide eraldamiseks ei eksisteeri masinõppe mudelit ega selleks vajalikku andmehulka ning regulaaravaldistega on selle ülesande lahendamine asjatult keerukas, seega otsustasin alampeatükis 6.2.2 kirjeldatud protsessile analoogselt kasutada andmetabeleid, mille loob sel puhul skript *ut_course_parser.py*. Et arvestada võimalike trükivigadega, kirjutasin alampeatükis 6.2.2 kirjeldatud isikunimede häguse eraldaja eeskujul kursuse nimede eraldaja *CourseTitleExtractor* ja sündmuse liigi eraldaja *CourseEventExtractor*; mõlema lävendiks on taaskord 80-protsendiline korrektsus.

6.2.4 Ootamatute sisendite käitlemine

Kui juturobot ei mõista saadud sisendit ehk juturoboti enesekindlus sisendile määratud kavatsuse osas on liialt väike, on mõistlik paluda kasutajalt sõnumi teisiti sõnastamist. Rasa dokumentatsioon toob selle saavutamiseks välja komponendi FallbackClassifier, mis väljastab võõra sisendi korral üldjuhul sobiva sõnumi. Dokumentatsioon toonitab, et komponendi seadistuses on kohustusliku parameetrina juturoboti enesekindluse lävend, millest madalama enesekindluse korral komponent aktiveerub [38]. Juturoboti katsetamise käigus leidsin, et sobiv lävend on 0.4 - kõrgema lävendi korral pakkus juturobot teatud juhtudel olukorrale järgnevat tegevust korrektselt, kuid see jäi madala enesekindluse tõttu sooritamata.

7 API kasutamine

Et juturoboti kasutus ei oleks piiratud töös loodud kasutajaliidesega, on mõistlik avada juturobot päringutele kõikvõimalikest allikatest. Selle saavutamise eesmärgil on juturobot läbi Nginx proksiserveri avatud API -päringutele. Rasa dokumentatsiooni [39] kohaselt ootab API-päringutele avatud juturobot POST-päringut, mille sisu on saatjat identifitseeriv sõne ja töötlemiseks mõeldud sõne. Peale sõnumi kättesaamist ja töötlemist saadab juturobot saadud tulemuse automaatselt tagasi sõnumi saatjale, nagu dokumentatsioonis kirjeldatud.

Saatjat identifitseeriv sõne ei pea sisaldama sõnumi allika teavet. Identifitseeriva sõne eesmärk on märgistada erinevaid kasutajaid nii, et igal kasutajal oleks unikaalne identifikaator, mida saab kasutada sõnumi saatmise allikast sõltumata. See identifikaator võimaldab juturobotil pidada meeles erinevate kasutajatega peetavate vestluste seisu ning seeläbi pidada vestlusi, mis vajavad järjestikuste sisendite töötlust ja mälus hoidmist. Selline on näiteks vestlus, kus esmalt küsib kasutaja mõne aine toimumisaegade kohta, mille peale küsib juturobot omakorda, millistest sündmustest on kasutaja huvitatud, ning saanud vastuse, et kasutaja tahab teada loengutega seonduvat teavet, tagastab juturobot vastava informatsiooni.

API-päringuid kasutab ka varasemates peatükkides mainitud veebileht. See edastab kasutajalt saadud päringu muutmata kujul POST-päringuga veebiserverile, mis edastab selle omakorda juturobotile, ning kuvab veebiserverilt saadud vastuse kasutajale.

Identifitseeriva sõnena kasutab veebileht UNIX-süsteemide ajastandardi algusest möödunud millisekundite arvu, mis fikseeritakse lehe avamise hetkel. See tähendab, et kasutaja vestlussessioon püsib senikaua, kui kasutaja lehte ei sule ega värskenda. Lehe värskendamisel ja uuesti avamisel määrab avatud leht endale uue identifikaatori. Sellist süsteemi kasutades on võimalik, et kaks kasutajat saavad tahtlikult või juhuslikult sama identifikaatori ning saavad seeläbi segada üksteise vestlusi. Arendusjärgus rakenduse kasutajaskond on aga väike ning juturoboti funktsionaalsus ei sisalda destruktiivseid päringuid, näiteks kasutaja õppeainelt maha registreerumist, mistõttu on kasutajate identifitseerimine lehe laadimise aja järgi piisav lahendus.

Kui juturoboti pädevusse lisanduvad destruktiivsed päringud, tuleb nende sooritamiseks paratamatult kasutaja tuvastada näiteks ülikooli kontole sisse logimise teel. Sellisel juhul on võimalik kasutajat identifitseeriv sõne luua kasutajale omase unikaalse teabe, näiteks isikukoodi, räsamise teel, mis tagab kasutajaid identifitseerivate sõnede unikaalsuse.

Turvalisuse seisukohast on oluline API avatuse määr. Rasa OpenAPI dokumentatsioonist on leitavad mitmed API päringud, millele peab puuduma üldine ligipääs, näiteks kasutuses oleva juturoboti mudeli mahalaadimine ja vestluste kulu manipuleerimine [40]. Dokumentatsioonis on selle probleemi lahendustena toodud välja sisseehitatud autentimise kasutamine ja API välisvõrgule mitteavamine [41]. Loodud robotis on rakendatud nendest viimast. Et juturobot on välisvõrgule kättesaadav ainult läbi Nginx proksiser-

veri, mis lubab läbi ainult REST API päringuid, on kõik OpenAPI dokumentatsioonis kirjeldatud päringud teostatavad ainult serverisiseselt saadetuna. Selleks vajab ründaja omakorda otsest ligipääsu serverile, kas läbi füüsilise terminali või ssh-sessiooni. Kuna taolise ligipääsuga on võimalik juturobotit kahjustada ka API päringuid kasutamata, on juturobot API avamisega võimalikuks muutunud rünnakute eest kaitsstud.

8 Kokkuvõte

Töös on antud ülevaade Tartu ülikooli Delta õppehoone külastajate abistamiseks mõeldud ingliskeelsest juturobotist. Bakalaureusetöö väljundiks olev juturobot suudab vastata kasutajate päringutele õppejõudude kabinetinumbrate ning kursuste toimumisaegade ja -paikade osas.

Juturoboti lähtekood on avalikustatud aadressil <https://github.com/Aeryan/Delta>. Lisaks käivitusjuhisele on lähtekoodist leitavad kõik töö käigus loodud tarkvaralahendused: juturoboti treenimisandmed, juturoboti lisakomponendid, varundatud andmebaas, andmebaasi uuendamise skriptid ja juturoboti tööks vajalike süsteemiteenuste konfiguratsioonifailid.

Lõplikus lahenduses kasutuses olevad tehnoloogiad täidavad neile seatud ülesandeid ning nende dokumentatsioon on kergesti kättesaadav. Töös enimkasutatud raamistiku Rasa Open Source dokumentatsioon on kohati puudulik või ebapädev ning sageli tuleb pöörduda avalike foorumite poole; mõlema kasutamisel lahendusid lõviosa tekkinud probleemidest.

Edaspidi on Delta majaplaani digitaliseerimise ja koostööruumide broneerimise tarkvarade loomisel võimalik juturobot nendega liidestada ning neid kasutades lisada juturobotile nii koostööruumide broneerimise kui ka majajuhina toimimise võimekus. Eestikeelsete keelemudelite valmimisel on loodud juturoboti eeskujul võimalik luua samaväärne eestikeelne juturobot ning nende ühildamisel on võimalik toetada nii eesti kui ka välistudengeid.

Viidatud kirjandus

- [1] Employment in Services (% of total employment) (modeled International Labour Organization estimate). <https://data.worldbank.org/indicator/SL.SRV.EMPL.ZS>. (29.04.2021).
- [2] Juturobot Mia - Telia. <https://www.telia.ee/era/chat/>. (29.04.2021).
- [3] Kontaktid | Elisa Eesti. <https://www.elisa.ee/et/elisast/kontaktid>. (29.04.2021).
- [4] Robotex International 2020 Ülemiste City Kratt rules. https://robotex.ee/wp-content/uploads/2020/10/6115_U%CC%88lemiste-Kratt_V004_Digital-1.pdf. (29.04.2021).
- [5] AKIT - Andmekaise ja infoturbe leksikon. <https://akit.cyber.ee/term/3088-api>. (29.04.2021).
- [6] Sülearvutid | Tartu Ülikooli arvutiteaduse instituut. <https://www.cs.ut.ee/et/oppimine/sulearvutid>. (19.04.2021).
- [7] e-Teatmik: IT ja sidetehnika seletav sõnaraamat. <http://www.vallaste.ee/>. (29.04.2021).
- [8] Rasa Glossary. <https://rasa.com/docs/rasa/glossary>. (29.04.2021).
- [9] Levenshtein, Vladimir I. Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 1966, 10(8):707–710.
- [10] Modules - Python 3.7.10 documentation. <https://docs.python.org/3.7/tutorial/modules.html>. (29.04.2021).
- [11] AKIT - Andmekaise ja infoturbe leksikon. <https://akit.cyber.ee/term/4886-reverse-proxy>. (29.04.2021).
- [12] Systemd - Kuutõrvaja. <https://kuutorvaja.eenet.ee/wiki/Systemd>. (04.05.2021).
- [13] Jurafsky, Dan, Martin, James H. *Speech and Language Processing (3rd ed. draft)*. Ch. 24, pp. 1-2. <https://web.stanford.edu/~jurafsky/slp3/> (03.05.2021).
- [14] Alphablues | We Build Virtual Customer Assistants. <https://alphablues.com>. (13.04.2021).
- [15] MindTitan - Build AI-driven organization with machine learning solutions. <https://mindtitan.com/>. (03.05.2021).

- [16] Statistikaameti asejuht: uus töötaja teeb virtuaalmaailmas ära reaalmaailma 10 inimese töö | DigiPRO. 2020. <https://digipro.geenius.ee/rubriik/uudis/statistikaameti-asejuht-uus-tootaja-teeb-virtuaalmaailmas-ara-reaalmaailma-10-inimese-too/>. (13.04.2021).
- [17] Sibold, Gregor. Päästeameti Facebooki juturobot õpetab sulle, kuidas enda maja mitte põlema panna | Digigeenius. 2017. <https://digi.geenius.ee/blogi/turvaline-tehnika/paasteameti-facebooki-juturobot-opetab-sulle-kuidas-mitte-enda-maja-polema-panna/>. (13.04.2021).
- [18] Krattijupid | Krattide veebileht. <https://www.kratid.ee/krattijupid>. (13.04.2021).
- [19] Bürokratt | Krattide veebileht. <https://www.kratid.ee/burokratt>. (13.04.2021).
- [20] Messenger Bot Marketing Made Easy with ManyChat. <https://manychat.com/>. (13.04.2021).
- [21] Juturoboti Bürokratt prototüübi lähtekood Eesti koodivaramus. <https://koodivaramu.eesti.ee/burokratt/juturobot-prototyypp/bot-interaction-service/-/tree/master/manager>. (13.04.2021).
- [22] Introduction to Rasa Action Servers. <https://rasa.com/docs/action-server/>. (18.04.2021).
- [23] Usage Statistics and Market Share of Web Servers, April 2021. https://w3techs.com/technologies/overview/web_server. (18.04.2021).
- [24] Fomantic-UI. <https://fomantic-ui.com/>. (18.04.2021).
- [25] Components. <https://rasa.com/docs/rasa/components/#ducklinghttpextractor>. (18.04.2021).
- [26] Introduction to Rasa X. <https://rasa.com/docs/rasa-x/>. (19.04.2021).
- [27] Rasa X | Rasa. <https://rasa.com/rasa-x/>. (19.04.2021).
- [28] How we make money - Rasa. <https://rasa.com/how-we-make-money/>. (19.04.2021).
- [29] Welcome to DeepSpeech's documentation! - DeepSpeech 0.9.3 documentation. <https://deepspeech.readthedocs.io/en/v0.9.3/>. (19.04.2021).

- [30] Neurotõlge. <https://neurotolge.ee/#info>. (19.04.2021).
- [31] API kirjeldus, Neurotõlge. <https://neurotolge.ee/api>. (19.04.2021).
- [32] Components. <https://rasa.com/docs/rasa/components/>. (22.04.2021).
- [33] NLU Training Data. <https://rasa.com/docs/rasa/nlu-training-data>. (22.04.2021).
- [34] Domain. <https://rasa.com/docs/rasa/domain>. (22.04.2021).
- [35] Rules. <https://rasa.com/docs/rasa/rules>. (22.04.2021).
- [36] Stories. <https://rasa.com/docs/rasa/stories>. (22.04.2021).
- [37] Õpiruumid ja Tehnika | Tartu Ülikooli arvutiteaduse instituut. <https://www.cs.ut.ee/et/instituudist/opiruumid-tehnika>. (22.04.2021).
- [38] Fallback and Human Handoff. <https://rasa.com/docs/rasa/fallback-handoff#fallbacks>. (23.04.2021).
- [39] Your Own Website. <https://rasa.com/docs/rasa/connectors/your-own-website/>. (13.04.2021).
- [40] Rasa Open Source Documentation. <https://rasa.com/docs/rasa/pages/http-api>. (13.04.2021).
- [41] Rasa Open Source HTTP API. <https://rasa.com/docs/rasa/http-api>. (13.04.2021).

Lisad

I. Litsents

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, **Rauno Jaaska**,
(autori nimi)

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose
Juturoboti loomine Delta õppehoonesse,
(lõputöö pealkiri)
mille juhendaja(d) on Mark Fišel,
(juhendaja nimi)
reprodutseerimiseks eesmärgiga seda säilitada, sealhulgas lisada digitaalarhiivi DSpace kuni autoriõiguse kehtivuse lõppemiseni.
2. Annan Tartu Ülikoolile loa teha punktis 1 nimetatud teos üldsusele kättesaadavaks Tartu Ülikooli veebikeskkonna, sealhulgas digitaalarhiivi DSpace kaudu Creative Commons'i litsentsiga CC BY NC ND 3.0, mis lubab autorile viidates teost reprodutseerida, levitada ja üldsusele suunata ning keelab luua tuletatud teost ja kasutada teost ärieesmärgil, kuni autoriõiguse kehtivuse lõppemiseni.
3. Olen teadlik, et punktides 1 ja 2 nimetatud õigused jäävad alles ka autorile.
4. Kinnitan, et lihtlitsentsi andmisega ei riku ma teiste isikute intellektuaalomandi ega isikuandmete kaitse õigusaktidest tulenevaid õigusi.

Rauno Jaaska
07.05.2021