

Tartu Ülikool
Loodus- ja täppisteaduste valdkond
Tehnoloogiainstituut

Karina Sein

**EESTIKEELSE KÕNESÜNTEESI VÕIMALDAMINE
ROBOOTIKA ARENDUSPLATVORMIL ROS**

Bakalaureusetöö (12 EAP)
Arvutitehnika eriala

Juhendaja:
robotika dotsent Karl Kruusamäe

Tartu 2020

Sisukord

Resümee	3
Abstract	3
1. Sissejuhatus	5
2. Kirjanduse ülevaade	7
2.1 Kõnelevad robotid	7
2.2 Eestikeelsed kõnetehnoloogiad	10
2.3 Kuidas töötavad kõnelevad robotid?	12
2.4 Kõnesüntesaatorid	13
2.5 Eesti keele kõnesüntesaatorid	13
2.6 ROS	15
3. Eesmärk ja ülesande püstitus	17
3.1. Funktsionaalsed nõuded	17
3.2. Mittefunktsionaalsed nõuded	17
4. Disain	18
4.1 Lahenduse arhitektuur	18
4.2 synthts_et	19
4.3 sound_play	20
4.4 synthts_et_ros	20
4.4.1 Serveri ülesseadmine	20
4.4.2 Serveri töö	21
4.4.3 Serveri sulgumine	25
4.4.3 Serveri kiirkäivitamine	25
5. Tulemus	26
5.1. Tulemi kirjeldus	26
5.2 Paigaldamine	27
5.3. Teadaolevad piirangud	27
6. Kokkuvõte	28
Kirjandus	29
Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks	33

Resüme

Eestikeelse kõnesünteesi võimaldamine robotika arendusplatvormil ROS

Eestis tegeletakse aktiivselt eestikeelsete kõnesüntesaatorite arendamisega. Sellest hoolimata on nende rakendamine robotitele olnud minimaalne. Tõenäoliseks põhjuseks on tehnoloogia ebaküpsus ning robotikasse rakendamise tehniline keerukus. Lõputöö eesmärk on luua modulaarne lahendus, mis võimaldaks võimalikult paljudele robotitele eestikeelne kõnesüntees kättesaadavaks teha. Selleks luuakse maailmas laialdaselt levinud robotite arendusplatvormi ROS tugi eestikeelsele kõnesüntesaatorile. Lõputöö tulemusena valmis ROSi kimp `synthts_et_ros`, mis tagab teksti heliks konverteerimise ja kõne ettemängimise funktsionaalsuse. Seega on loodud modulaarne lahendus, mis võimaldab arendajatel lisada robotile eestikeelse kõne võimekuse.

CERCS: T120 Süsteemitehnoloogia, arvutitehnoloogia; T125 Automatiseerimine, robotika, juhtimistehnika

Märksõnad: ROS, robotika, eestikeelne kõnesüntees

Abstract

Enabling Estonian speech synthesis on the Robot Operating System (ROS)

Despite Estonian speech synthesizers being actively developed in Estonia, their implementation on robots has been minimal. The probable cause for this might be the immaturity of the available technologies and technical complexity of implementation to robots. This bachelor's thesis proposes to create a modular solution to make Estonian speech synthesizers available to as many robots as possible. For this purpose, ROS support (ROS being a widespread development platform for robots) was created to an existing Estonian speech synthesiser software. As a result of the thesis, a ROS package named `synthts_et_ros` has been created. This provides the functionalities of text-to-speech synthesis and playing

back the speech to the user. Therefore, a modular solution was created which enables developers to add Estonian speech synthesis to ROS-enabled robots.

CERCS: T120 Systems engineering, computer technology; T125 Automation, robotics, control engineering

Keywords: ROS, robotics, speech synthesis in Estonian

1. Sissejuhatus

Seoses pideva tehnoloogilise arenguga näeme järjest rohkem roboteid igapäevaelus ning ka abilistena tööl. Neid kasutatakse tootmises [1], meditsiiniliste protseduuride läbiviimisel [2], koduabilistena [3] ning paljudel muudel aladel. Tulevikus võib olla võimalik robotitega inimkeeles suhelda. Näiteks võib olla võimalik koduabilisest robotile öelda, mida õhtusöögiks valmistada ning robot saab sellest aru ning oskab vastata, kaua sellega aega läheb. Robotiga sujuva ja loomuliku vestluse saavutamiseni pole veel jõutud, aga selle suunas on tehtud edusamme. Näiteks on Shanghai raamatukogus kasutusel robot, mis oskab vastata kuuekümmele erinevale küsimusele [4], Münheni lennujaamas robot, mis vestleb küllastajatega [5] ning Lääne Austraalia kunstmuuseumis robot, mis tutvustab teoseid [6].

Inimese ja roboti vahelise dialoogi õnnestumine eeldab mitut tehnoloogiliselt keerukat sammu. Robot võtab inimese kõne mikrofoni kaudu sisendiks. Seejärel tehakse helist tekst, mida on võimalik töödelda. Saadud teksti analüüsimise eesmärk on aru saada öeldu sisust, et valida sobiv vastus. Vastus genereeritakse seejärel heliks ning mängitakse kuulajale kõlarite kaudu ette. Protsessi keerulisemad osad on heli põhjal teksti loomine sõnade tuvastamise keerukuse tõttu ning selle analüüsimine, mille üks väljakutseid esitavaid asjaolusid on, et ühte mõtet saab mitmel erineval viisil väljendada.

Eestikeelsete kõnetehnoloogiatega tegeletakse Eestis aktiivselt. Näiteks on võimalik lasta Postimehe kodulehelt uudiseid ette lugeda. Samas tegeleb Eesti Keele Instituut intensiivselt eestikeelse kõne sünteesimise arendamisega [7]. Sellest hoolimata on eestikeelsete kõnesüntesaatorite robotitele rakendamine olnud minimaalne - põhjus ilmselt tehnoloogia ebaküpsus ning robotikasse rakendamise tehniline keerukus.

Käesoleva lõputöö eesmärk on luua modulaarne lahendus, mille abil võimalikult paljudele robotitele eestikeelne kõnesüntees kättesaadavaks teha ning luua seni puuduolev ühendus eesti keele süntesaatorite ja robotite vahele. See võimaldaks arendajatel kiirelt ja lihtsalt lisada kõne sünteesimise funktsioonalsus oma robotitele. Selle saavutamiseks luuakse maailmas laialt levinud robotika arendusplatvormi ROS (*Robot Operating System*) tugi eesti keele süntesaatorile.

Lõputöö koosneb kokku kuuest peatükist. Teises peatükis uuritakse erinevaid kõnelevaid roboteid ning kõne sünteesimise meetodeid. Lisaks vaadatakse lähemalt ROSi. Kolmandas

peatükis täpsustatakse lõputöö eesmärki ning nõudeid. Peatükis neli tutvustatakse lõpptulemuse erinevaid komponente ja kuidas need omavahel suhtlevad. Seejärel analüüsitakse viiendas peatükis tulemi funktsionaalsust ja piiranguid ning arutatakse edasisi võimalikke arendusi. Kuuendas peatükis esitatakse lõputöö kokkuvõte.

2. Kirjanduse ülevaade

2.1 Kõnelevad robotid

Inimese kombel kõnelevaid seadmeid nähakse tihti kui kõige kasutajasõbralikumaid mooduseid tehnoloogia igapäevasele kättesaadavaks muutmisel. Näiteks on inimesed harjunud autodes navigatsioonisüsteemidega, mis inimhääles juhile suuniseid jagavad. Jõudsalt on arenenud ka nutiabiliste nagu Amazoni Alexa [8], Google'i Assistant [9], Apple'i Siri [10] ja Microsofti Cortana [11] kasutamine. Kuid lisaks kasutusmugavusele võib teksti ettelugev tehnoloogia aidata vaegnägijaid, kellel muidu oleks piiratum juurdepääs päevakajalisele informatsioonile. Seetõttu on igati mõistetav ka kõnetehnoloogiate rakendamine robotika eri valdkondades.

Interaktiivsete mänguasjade kujul võivad isegi lapsed kokku puutuda kõnelevate robotitega. Üks selline mänguasi on *Fancy Nancy Shall We Be Fancy* nukk [12]. Nukk oskab lapsi inglise keeles juhendada, kuidas temaga mängida ning reageerib, kui tema riiete värvi muudetakse [12]. Interaktiivsus saavutatakse helifailide ettemängimisel, mis on nuku peale salvestatud [3]. Veel üks näide kõnelevatest mänguasjadest on *Talking Hamster Plush Toy*, mis oskab lapse poolt ette öeldud fraase korrata [13]. Lisaks on olemas *Wenjuan Remote Control Intelligent Robot Toy*, mis saadab puldilt sisendiks võetud kõne robotile ning mängib selle koheselt ette [14].

Kõne funktsionaalsust võib leida ka teenindusrobotites. Näiteks katsetati 2018. aastal Müncheni lennujaamas, kuidas reisijad suhtuksid robotitesse, mis meenutavad inimesi [5]. Josie Pepperi nimeline robot oskab inglise keeles tervitada ja vastata lendude, restoranide ja poodidega seotud küsimustele (joonis 1a) [5]. Teine sarnane robot on Troika, mis 2017. aastal lisati Incheoni rahvusvahelisse lennujaama (joonis 1b) [15]. Troika oskab inglise, korea, hiina ja jaapani keeles edastada informatsiooni lendude ajakava kohta ning vajadusel saata reisijad õige väravani [15].



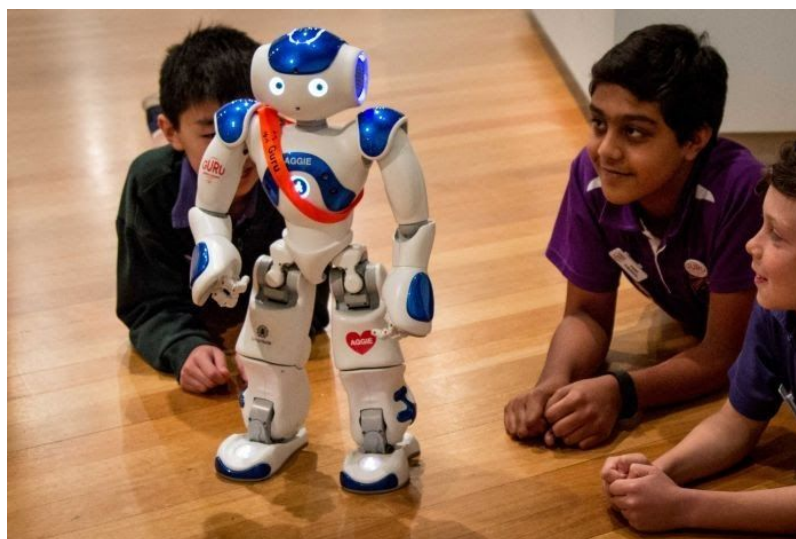
(a)



(b)

Joonis 1. (a) *Josie Pepper* [5], (b) *Troika* [16].

Kõnelevaid teenindusroboteid kasutatakse ka muuseumides. Näiteks lisati 2016. aastal Lääne Austraalia Kunstimuuseumisse robot Aggie (joonis 2), mille eesmärk on muuseumi külastajatele pakkuda omapärast ekskursiooni kogemust [6]. Robot oskab kõndida, tutvustada kunstiteoseid ning neid kommenteerida [6]. Teine koht, kus on kasutatud selliseid roboteid, on *The Smithsonian*'i muuseumid [17]. Seal kasutati robotit Pepper, et meelitada inimesi muuseumide vähem külastatud ruumidesse ning osadesse [17]. Pepper oskab külastajatele interaktiivselt tutvustada muuseumi ruume ning erinevaid teoseid [17].



Joonis 2. *Robot Aggie* [6].

Tehnoloogia areneb pidevalt ning kõne sünteesimist hakatakse järjest rohkemates kohtades kasutama sealhulgas ka kodudes. Näiteks Guofeng Zheng jt disainisid ja

implementeerisid oma uurimistöös mooduse, kuidas inimene saaks hiina keeles robotiga suhelda [18]. Selles projektis kasutati muude vahendite seas ka kõne sünteesimist ning ROSi [18]. Projekt oli suunatud teenindusrobotite arendamisele ning tarkadele kodudele [18]. Veel üks näide on Rudy, mis on suunatud vanemate inimeste hooldamisele [19]. Rudy oskab kasutajaga suhelda, meelde tuletada, millal ravimeid võtta ning videokõnesid pidada [19]. Ka Eestis võib leida kodudes kasutatavaid roboteid nagu näiteks robottolmuimeja Roomba [20] ning põrandapesumasin Braava [21]. Mõlemad robotid kasutavad kõne funktsiooni, et kasutajale edastada inglise keeles informatsiooni töö jooksul esinenud probleemide kohta nagu kinnijäämine või aku tühjenemine.

Kaugkõhalolu on tehnoloogiate kogum, mille eesmärk on robotite abil võimaldada inimestel olla kohtades, kus nad füüsiliselt ei asu [22]. Inimene kasutab oma asukohas asuvaid sensoreid, mille andmed kantakse robotile väljundiks [22]. Siinkohal kantakse inimkõne robotite abil ühest kohast teise võimalikult väheste viivitustega [22]. Üks näide kaugkõhalolu robotitest on iRobot Ava 500 (joonis 3a) [23]. iRobot Ava 500 võimaldab tahvelarvuti abiga planeerida kaugkõhalolu [23]. Planeeritud ajal liigub robot eelnevalt määratud ruumi [23]. Mõlemal osapoolel on kõlarid, kaamera ning ekraan, et oleks võimalik nii inimeste kõne kui ka visuaalset pilti üle kanda [23]. Teine sarnane kaugkõhalolu robot on Double 3 (joonis 3b) [24]. Ka sellel robotil on liikumise funktsionaalsus, kaamerad, mikrofonid ja kõlarid [24].



(a)



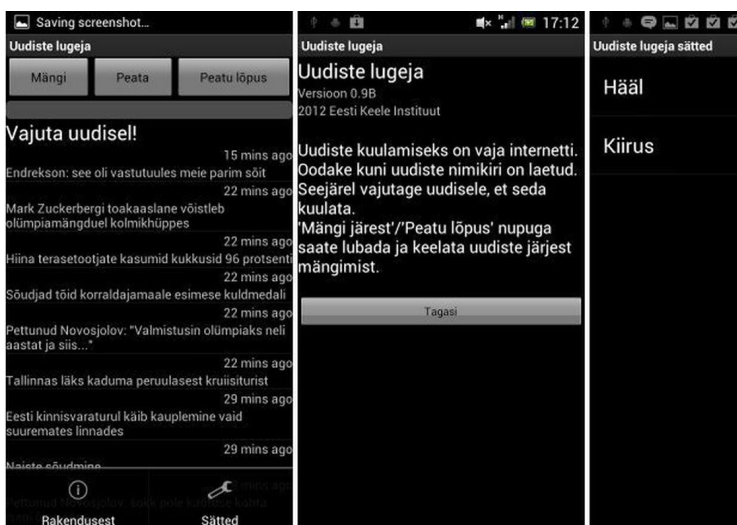
(b)

Joonis 3. (a) *iRobot Ava 500* [23], *Double 3* [25].

2.2 Eestikeelsed kõnetehnoloogiad

Kõnetehnoloogiate rakendusi leidub ka Eestis. Näiteks lisati 2018. aastal Elisa Iseteeniduskeskkonda kliente abistav *chatbot* ehk juturobot Annika [26]. 2019. aastast tegutseb Annika mitte ainult teksti teel klientidega vaid ka kõneldes [26]. Hetkel on Annika võimeline mõistma eestikeelset kõnet ning selle alusel kliente edasi suunama [26]. Tuleviku plaanide seas on lihtsamatele küsimustele vastamine [26].

Eesti Keele Instituut kasutab kõne sünteesimist oma nutirakenduses, mis loeb ette uudiseid (joonis 4a) [27]. Rakenduse kasutaja saab sirvida uudiseid, valida, kas uudiseid loetakse ette järjest või ühe kaupa ning saab määrata kõne häält ning kiirust [27]. Ka Postimees kasutab oma kodulehel kõne funktsionaalsust. Postimehe digipaketi tellijatel on võimalik lasta artikleid ette lugeda (joonis 4b).



(a)



(b)

Joonis 4. (a) *Uudiste lugeja* [27], (b) *Kuulatav artikkel Postimehe lehel* [28].

Ka meditsiini valdkonnas võib leida kõnetehnoloogiate rakendusi. Näiteks on välja töötatud moodus, mis võimaldab radioloogide kõne tekstiks konverteerida [29]. See võimaldab mugavalt kirja panna selle, mida radioloogid piltidelt välja loevad [29]. Lääneriikides kasutatakse sellise töö jaoks spetsialiseerunud sekretäre, Eestis aga pole vastavate kvalifikatsioonidega inimesi leida [29]. Seega oli vastav tarkvara vajalik, et vabastada radiolooge rutiinsest tööst [29].

Eestikeelsete kõnetehnoloogiate kasutust leidub ka robotitel. Näiteks on olemas mänguasjad, mis suhtlevad lastega eesti keeles nagu robot Robert, mis on võimeline lausuma 43e erinevat ettesalvestatud lauset eesti keeles (joonis 5) [30].

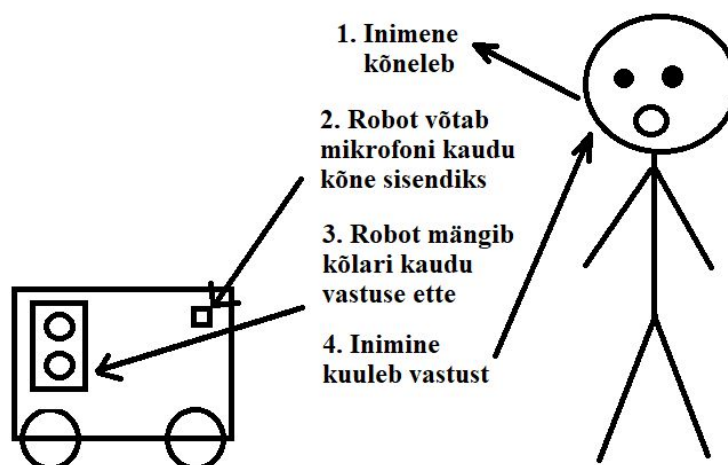


Joonis 5. Robot Robert [30].

Ülikoolid Eestis võtavad samuti osa eesti keelsete kõnetehnoloogiate arendamisest. 2018. aastal lisandus Tartu Ülikooli üheks semestriks robot Pepper [31]. Semestri jooksul õppis Pepper eesti keeles kõnelema ning jälgendama inimeste liigutusi [31]. Tallinna Tehnikaülikoolis loodi tarkvaralahendus NAO robotile, mis võimaldab inimesel suhelda robotiga eesti keeles [32]. Lahenduse puhul on nii salvestatud kõne kui ka sünteesitud kõne failid madala heli kvaliteediga [32]. Seega on NAO roboti kõnetuvastusprogrammi töö raskendatud ning osa sõnu võivad olla valesti tuvastatud [32].

2.3 Kuidas töötavad kõnelevad robotid?

Robotiga suhtlemisel on mitu tehniliselt nõudlikku etappi. Robot võtab inimese kõne mikrofoni kaudu sisendiks ning loob selle alusel teksti. Saadud teksti põhjal määratakse sobiv vastus. Vastus genereeritakse heliks ning mängitakse kõlari kaudu inimesele ette. Protsess on kujutatud joonisel 6.



Joonis 6. Etapid inimese ja roboti vahelisel suhtlemisel.

Selleks, et roboteid kõnelema panna on erinevaid variante:

- 1) eelsalvestatud helifailide mahamängimine [12] [30],
- 2) robotit juhtiva inimese kõne ülekandmine [22] ja
- 3) kõne sünteesimine [18] [32].

Esimese variandi puhul salvestatakse inimese kõne seadme peale ning mängitakse sisseprogrammeeritud tingimustel kasutajale ette. [12] [30]

Inimkõne ülekandmise puhul ei toimu salvestamist. Inimese kõne võetakse mikrofoni kaudu sisendiks ning saadetakse teises asukohas asuvale robotile ette kandmiseks. Väljundiks on robotile saadetud kõne. [22]

Kõne sünteesimise puhul on roboti peal tarkvara, mis tagab keele sünteesimise funktsionaalsuse. Programm saab sisendiks teksti, mis konverteeritakse heliks ning mängitakse kõlarite vahendusel kasutajale ette. [18] [32]

Mitte ühegi variandi puhul ei saa robot aru sünteesitud kõne tähendusest.

2.4 Kõnesüntesaatorid

Kõnesüntesaatorite eesmärk on imiteerida inimkõnet sisendiks etteantud teksti alusel. Kõnesüntees koosneb kolmest sammust:

- 1) protsessimine,
- 2) häälduse kokkupanemine,
- 3) tekstile hääle andmine [33].

Kõnesüntesaator alustab tööd protsessimisest, mille käigus leitakse sobiv viis, kuidas sõna hääldada [33]. Raskusi valmistavad sõnad, millel on mitu erinevat hääldust nagu näiteks eesti keeles palk ja kann või sõnad, mille hääldus ja kirja pilt erinevad [33]. Erinevate häälduste vahelt valitakse välja see, millel on kõige suurem tõenäosus olla õige [33]. Häälduse välja valimise järel tuleb tekst teha tükkideks ja foneemidest ehk väikseimast keeleüksustest [34] uuesti kokku panna [33]. Viimase sammuna tuleb tekstile hääle anda. Hääle sünteesimiseks on kolm levinumat varianti:

- 1) juhendav,
- 2) formantsüntees,
- 3) artikulaarne [33].

Juhendav meetod kasutab eelnevalt salvestatud inimeste kõnet [33]. Selle meetodi eeliseks on loomulik kõla [33]. Formantsüntees kasutab erineva helikõrgusega helisid, et imiteerida inimhäält [33]. Erinevalt juhendavast meetodist, on formantsünteesiga võimalik kõiki sõnu hääldada sealhulgas ka võõrsõnu ning väljamõeldud sõnu, aga puudusi on heli sünteetilises kõlas [33]. Artikulaarse meetodi tööpõhimõte seisneb inimhääle järgi modelleerimises [33]. Selle sünteesi mooduse puuduseks on keerulisus, mille tõttu on seda meetodit vähem katsetatud [33].

2.5 Eesti keele kõnesüntesaatorid

Eestikeelse kõne sünteesimiseks vajalike ressursside leidmiseks on kaks põhilist allikat: Eesti Keele Instituudi kõnesünteesi veebileht [7] ja tarkvaralabori veebileht [35]. Eesti Keele Instituudi kõnesünteesi veebilehel [7] saab testida erinevaid eesti keele sünteesimise

meetodeid ja hääli ning leida juhised tarkvara installimiseks. Kõnesünteesi lahenduste seas on esindatud üksuste valikul põhinev süntees, formantsüntees, difoonsüntees ning peidetud Markovi mudelil põhinev kõnesüntees ehk HTS-kõnesüntees¹ nii eraldi seisval kujul kui ka Festivali tarkvarale ehitatult [7]. Festival on vabavara, mis tagab kasutajale raamistiku, millele ehitada kõnesünteesi süsteeme, ning sisaldab erinevaid näidismoduleid [36].

Eesti Keele Instituut on valimistanud tabeli, mis võrdleb nende poolt arendatavaid kõne sünteesimise meetodeid [7] (Tabel 1).

Moodulid	Kompaktsus	Segmentaalne kvaliteet	Kõne ladusus	Tekstitöötlus	Sünteesi-juhtimise tase	Kõne variatiivsus
UnitSelection-et (üksuste valikul põhinev süntees)	madal	kõrge	mõõdukas	kõrge	madal	kõrge
HTS-et (HTS-süntees)	mõõdukas	mõõdukas	kõrge	kõrge	kõrge	mõõdukas
eSpeak-et (formantsüntees)	kõrge	madal	mõõdukas	madal	kõrge	madal
Mbrola-et (diftoonsüntees)	mõõdukas	mõõdukas	mõõdukas	kõrge	mõõdukas	mõõdukas

Tabel 1. Eesti keele süntesaatorite võrdlus [7].

HTS-kõne sünteesimise tarkvara `synthts_et` on välja töötanud Indrek Kiisler Eesti Keele Instituudist [37]. `Synthts_et` on eesti keelt sünteesiv programm, mille baasiks on rakendusliides “`hts_engine API`”, mis kasutab peidetud Markovi mudeli põhimõtteid ning mille arendajaks on HTS Working Group [37]. Eeliste seas on mitu erinevat häält, mis genereeritud kõnele on võimalik anda [7]. Samas on saadud heli kvaliteet ja arusaadavus kõrged ning saadud kõne on ladus [7]. Kõnet annab erinevate parameetritega reguleerida [7]. Tarkvara kasutamiseks tuleb `synthts_et` alla laadida ning kompileerida [37]. Seejärel käib kasutamine käsurea vahendusel [37]. GitHubi lehel on toodud kasutamise kohta järgnev näide [37]:

```
$ bin/synthts_et -lex dct/et.dct -lexd dct/et3.dct -o out_tnu.wav  
-f in.txt -m htsvoices/eki_et_tnu.htsvoice -r 1.1
```

`bin/synthts_et` on fail, kus asub `main` funktsioon, `-lex` määrab analüüsi sõnastiku, `-lexd` määrab ühestaja sõnastiku, `-o` määrab väljundiks saadud WAV-faili asukoha ja nime, `-f`

¹ HMM/DNN-based Speech Synthesis System (HTS), <http://hts.sp.nitech.ac.jp/>

määrab sisendiks võetava tekstifaili asukoha, -m määrab kasutatava hääle ning -r määrab kõne kiiruse [37].

HTS-kõnesüntees on saadaval ka Festivali põhisenä [7].

Üksuste valikul põhineva sünteesi puhul on tegemist korpuspõhise sünteesiga [7]. Kõnekorpused koosnevad diktorite salvestatud kõnest ning on sünteesi helilise poole aluseks [7]. Eelisteks on loomuliku kõlaga kõne ning segmentide kõrge kvaliteet [7]. Erinevalt HTS-kõnesünteesist pole genereeritava kõne parameetreid võimalik reaajas muuta [7]. Lisaks on kõnekorpused mahukad [7].

Formantsünteesi variant on eSpeak tarkvarapõhine [7]. eSpeak on vabavaraline kõnesüntesaator, mis toetab mitmeid keeli sealhulgas ka eesti keelt [38]. Formantsüntees ei põhine kõnekorpustel vaid allikas-filter mudelil [7]. Selles mudelis modelleerib allikas häälekurdude tööd ning filter modelleerib formante ehk osahääli [39] [7]. Formantsünteesiga saadud kõne on aga raskesti arusaadav segmentide madala kvaliteedi tõttu [7]. Vastukaaluks ei ole nõuded arvutusvõimekusele ja mälumahule kõrged [7].

Difoonsüntees kasutab kõneüksustena segmente, mis koosnevad kahest järjestikusest häälikust, ehk difoone [7].

Lisaks on veel olemas tarkvaralabori veebilehel närvivõrkudel põhinev kõne sünteesimise meetod [35]. Tarkvara on treenitud kasutades Deep Voice 3 Pytorch tarkvara ning aluseks on kasutatud seitsme erineva raadio diktori kõnet [35]. Mitmele häälele baseerumise eeliseks on arusaadav kõne, aga selle eest on müra saadud helis rohkem [35].

2.6 ROS

ROS (*Robot Operating System*) on raamistik, mis on mõeldud robotite tarkvara arendamiseks [40]. ROSi peamine eelis uurimistöö raames on selle laialdane kasutus robotitel [41]. Seega tagab ROSi valik, et välja töötatud lahendus saab olema kättesaadav võimalikult paljudele robotitele.

ROSi peamine eesmärk on lihtsustada koodi taaskasutamist robotite arendamisel [42]. Lisaks on koodi lihtne testida töövahenditega, mis ROSiga kaasa tulevad [42]. ROSi raamistik toetab Pythonit, C++i ja Lispi [42]. Katselised teegid on olemas ka Javale ning Luale [42].

ROSi organiseeritakse tarkvara kimpudesse [43]. Kimp moodustab kasuliku mooduli ning võib sisaldada:

1. ROSi sõlmesid,
2. teeke, mis ei kasuta ROSi,
3. andmestikke,
4. konfiguratsiooni faile
5. ning muid komponente [43].

Kimpude eesmärk on lihtsustada koodi taaskasutamist [43].

ROSi pakendi (ingl *wrapper*) kasutamine on meetod, mis illustreerib, kuidas ROSiga saab koodi taaskasutada [44]. ROSi pakendi implementeerimisel alustatakse eraldiseisvast lähtekoodist, mis ei kasuta ühtegi ROSi osa oma koodis [44]. Seejärel luuakse ROSi sõlm koos vastava teenusega, mis teeb vajaliku initsialiseerimise ning kasutab ära algse koodi poolt tagatud funktsionaalsused [44]. Seejuures on vajalik CMakeLists.txt fail õigesti vormistada, et oleks selge, kuidas lähtekoodi kompuleerida ning kuhu installida [45]. Sellise kasutuse eeliseks on, et koodi on võimalik käitada nii ROSi peal kui ka eraldi, ilma et algset koodi oleks muudetud [44].

ROS tagab ka mooduse, kuidas erinevate protsesside ja teemade vahel sõnumeid vahetada [42]. Lisaks on ROS andmetekeskse ning modulaarse olemusega, mis teeb sõlmede lisamise kergeks [46].

Lisaks kasutatakse lõputöös ROSi toiminguserverit (ingl *action server*) ning -klienti (ingl *action client*). Nendevaheline suhtlus toimub üle toimigute. Toimingus on täpsustatud saadetavate andmete tüübid eesmärgile, tagasisidele ning kinnitusele. Klient saadab serverile eesmärgi ning server saadab vastu töö protsessi kohta tagasisidet ning töö lõpetades kinnituse edukuse kohta. [47] [48]

3. Eesmärk ja ülesande püstitus

Lõputöö eesmärk on luua modulaarne moodus, kuidas võimalikult paljudele robotitele kõne sünteesimine kättesaadavaks teha. Selle saavutamiseks luuakse eestikeelse kõne süntesaatorile synthts_et ROSi pakend.

3.1. Funktsionaalsed nõuded

- 1) Konverteerib eestikeelse teksti heliks, mis mängitakse kõlarite kaudu ette.
- 2) Kui programm töötab, peab olema võimalik kõne sünteesimist korduvalt välja kutsuda (st teha eestikeelse kõnesünteesi päringuid).
- 3) Kui kõne süntees ebaõnnestub, antakse sellest päringu teinud sõlmele teada.
- 4) Päringu teinud sõlm saab tagasisidet, kui kaugel ollakse helifaili ette mängimisega.
- 5) Päringu teinud sõlme programmi teavitatakse helifaili ette mängimise edukusest.
- 6) Kogu süsteem on käivitav ühe käsu ehk kiirkäivitusfailiga.
- 7) Programmi toimimiseks vajalikud ajutised failid eemaldatakse sulgemisel.

3.2. Mittefunktsionaalsed nõuded

- 1) Kasutatakse Linuxi versiooni Ubuntu 16.04
- 2) Kasutatakse ROS Kinetic'ut.
- 3) Kõne sünteesimiseks kasutatakse eelnevalt olemasoleva kõnesüntesaatori loogikat, mis on ROSile kättesaadavaks tehtud.
- 4) Kõne häält, kiirust ning kõla seadistatakse parameetri serveri vahendusel.
- 5) Implementatsioon kasutab ROSi toiminguserverit.

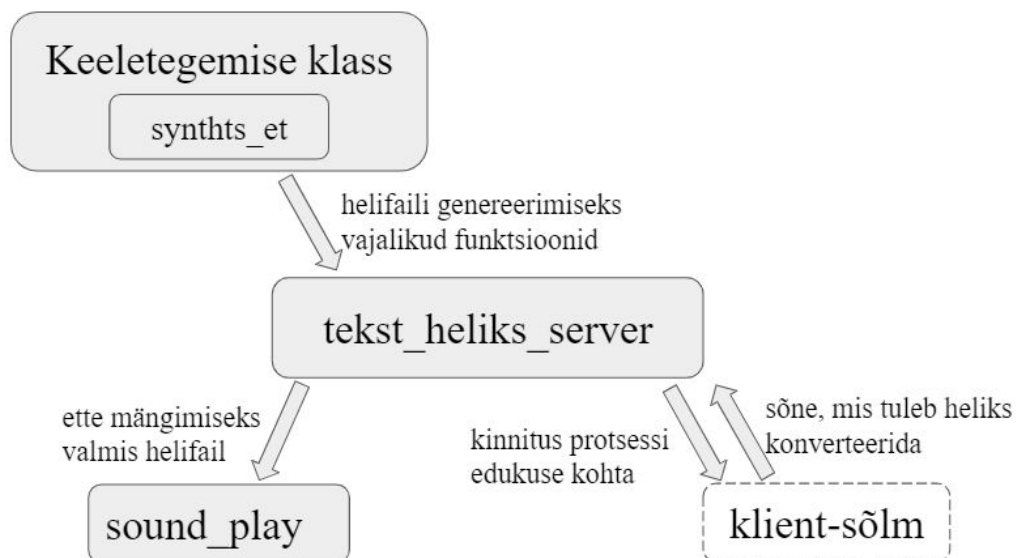
4. Disain

4.1 Lahenduse arhitektuur

Käesoleva lõputöö raames loodi `synthts_et` ROSi kimp, mis võimaldab robotite tarkvara arendajal integreerida ROSi-põhisesse robotsüsteemi eesti keele sünteesi võimekuse. Valminud lahendus ja paigaldusjuhend on leitav GitHubi koodihoidlas: github.com/ut-ims-robotics/synthts_et_ros.

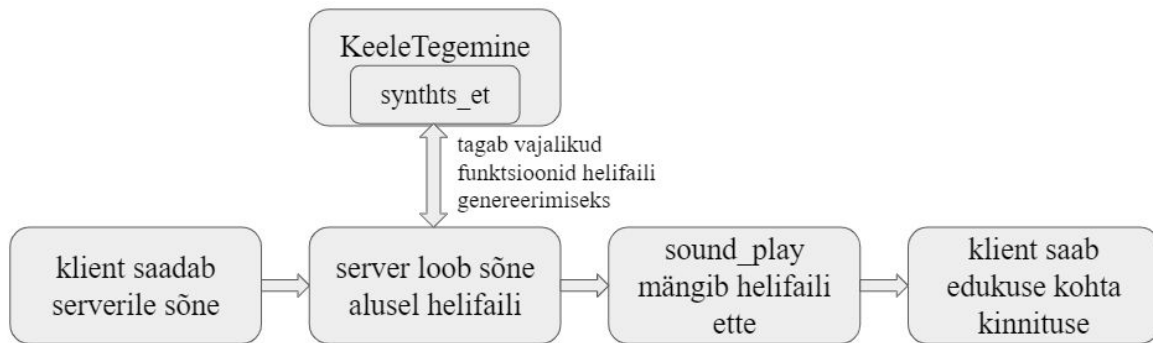
Joonisel 7 on esitatud loodud lahenduse arhitektuur. Valminud lahendus koosneb neljast komponendist. Need komponendid on:

1. KeeleTegemise klass - kapseldab `synthts_et` poolt pakutava keelesünteesi võimekuse
2. `sound_play` - mängib ette helifaile
3. `tekst_heliks_toiminguserver` - genereerib sõnest helifaili ja saadab ette mängimisele
4. klient-sõlm - robotsüsteemi arendaja poolt implemneteeritav programm, et saata toiminguserverile päringuid, mis sisaldavad sünteesitavat eestikeelset teksti



Joonis 7. `synthts_et` ros kimbu arhitektuur.

Kogu süsteemi protsessidiagramm on illustreeritud joonisel 8.



Joonis 8. *synthts_et_ros* kimbu funktsionaalne kirjeldus.

Selleks, et sünteesida eestikeelset kõnet ja seda roboti kõlarite vahendusel ette mängida, tuleb robotisüsteemi arendajal implementeerida nn klient-sõlm. Klient-sõlme abil saadetakse toiminguserverile sõne ehk tekst, mida heliks konverteerida. Server kasutab `synthts_et` funktsioone, et genereerida nende põhjal eestikeelset kõnet sisaldav WAV-tüüpi helifaili. Saadud helifail edastatakse `sound_play` kimbule, mis mängib heli kõlarite kaudu ette. Server saadab kliendile tagasisidena genereeritud helifaili pikkuse sekundites ning seejärel iga sekund, kui kaugel ollakse helifaili ette mängimisega. Kui helifaili ettemängimine on lõpetanud, saadetakse kliendile kinnitus, kas töö oli edukas. Kui sünteesi käigus esines tõrge, siis tagastab server kliendile veakoodi.

4.2 `synthts_et`

`Synthts_et` on eestikeelse kõne sünteesimise tarkvara [37]. Tarkvara kasutamise tulemusena saadud kõne on kvaliteetne, arusaadav ning ladus. Sünteesitavat kõne annab erinevate parameetritega reguleerida. Lisaks on tarkvara allalaadimiseks olemas selged juhised. Nende eeliste tõttu valiti `synthts_et` lõputöös kasutatavaks kõnesüntesaatoriks.

`Synthts_et` tagab oma `dct`-kaustas olevad analüüsi ja ühestaja sõnastikud ning `hstvoices`-kaustas olevad hääled, mis on vajalikud kõne sünteesimiseks.

Teiseks on kirjeldatud `synthts_et` bin-kaustas asuvas `synthts_et.cpp` failis ära loogika, kuidas saada sõnest helifail. Kuna olemasolev `synthts_et.cpp` sisaldas endas juba `main()` funktsiooni, siis ei saanud ROSi funktsionaalsust selle ümber otse pakendada.

Lõputöö raames loodi uus `KeeleTegemine` klass (joonised 7 ja 8), mis sisaldab kogu `synthts_et.cpp` funktsionaalset osa. `KeeleTegemine` klass kapseldab `synthts_et` poolt pakutava keelesünteesi võimekuse.

4.3 `sound_play`

ROSi `audio_common` teek sisaldab muuhulgas ka `sound_play` kimpu, mis võimaldab mängida ette nii sisseehitatud helisid kui ka OGG- ja WAV-tüüpi audiofaile [49]. Antud lahenduses kasutatakse `sound_play` kimbu `play.py` sõlme `synthts_et` poolt genereeritud eestikeelset kõnet sisaldava WAV-tüüpi audiofaili esitamiseks robotil.

4.4 `synthts_et_ros`

Valminud lahenduse keskne komponent on ROSi toiminguserverina käitatav sõlm nimega `tekst_heliks_server`. Kogu `tekst_heliks_server` sõlme töö võib jaotada kolme faasi: serveri ülesseadmine, serveri töö ja serveri sulgumine.

4.4.1 Serveri ülesseadmine

Serveri ülesseadmisel luuakse kasutaja kodukausta peidetud `.tekstHeliks`-kaust, mis on määratud asukohaks, kuhu hakatakse helifaile genereerima. Seejärel käivitatakse ROSi toiminguserver, mis jääb päringute ootele. Toiminguserveri loomisel kasutatakse parameetri serverist saadud väärtusi, et luua uus `KeeleTegemise` objekt. `Synthts_et` tagab `KeeleTegemise` objektile kõik vajalikud funktsioonid helifaili genereerimiseks teksti alusel.

Parameetrite fail `parameetrid.yaml` asub `synthts_et_ros/config`-kaustas. Seal on kaksteist muudetavat parameetrit:

1. `lex_file_name` - analüüsi sõnastiku asukoht ja failinimi sõnena. Fail leidub `synthts_et` `dct` kaustas. Sõnastik on vajalik helifaili genereerimisel.

2. `lexd_file_name` - ühestaja sõnastiku asukoht ja failinimi sõnena Fail leidub `synthts_et` `dct` kaustas. Sõnastik on vajalik helifaili genereerimisel.
3. `fn_voices` - kasutatava hääle asukoht ja failinimi sõnena. Kasutatavate häälte asukoht on `synthts_et` `htsvoices` kaust. Parameeter määrab genereeritava kõne hääle.
4. `dur_fname` - kestusfaili asukoht ja failinimi sõnena. Kui `write_durlabel` on tõene, siis kirjutatakse antud faili kõik kliendilt saadud sõne tähed, millele on juurde lisatud pausid ning märged häälduse kohta.
5. `speed` - kõnetempo komakohalise arvuna. Parameeter määrab sünteesitava kõne kõnetempo.
6. `half_tone` - sünteesitava kõne kõla muutev parameeter komakohalise arvuna.
7. `gv_weight1` - sünteesitava kõne kõla muutev parameeter komakohalise arvuna.
8. `gv_weight2` - sünteesitava kõne kõla muutev parameeter komakohalise arvuna.
9. `print_label` - kui antud tõeväärtus on tõene, siis prinditakse iga sõnes oleva silbi kohta välja informatsiooni, mis on seotud helifaili genereerimisega. Täpsemalt prinditakse välja keele_tegemine.cpp failis oleva `genereeri_lause` funktsiooni `do_all` väljund.
10. `print_utt` - kui antud tõeväärtus on tõene, siis prinditakse välja laused, millest kliendilt saadud sõne koosneb.
11. `write_raw` - kui antud tõeväärtus on väär, siis kirjutatakse väljundfaili päis, mis sisaldab informatsiooni genereeritud WAV-tüüpi faili kohta.
12. `write_durlabel` - tõeväärtus, mis määrab, kas `dur_fname` parameetris määratud faili kirjutatakse midagi. Kui antud väärtus on tõene, siis peab olema määratud kestusfail.

4.4.2 Serveri töö

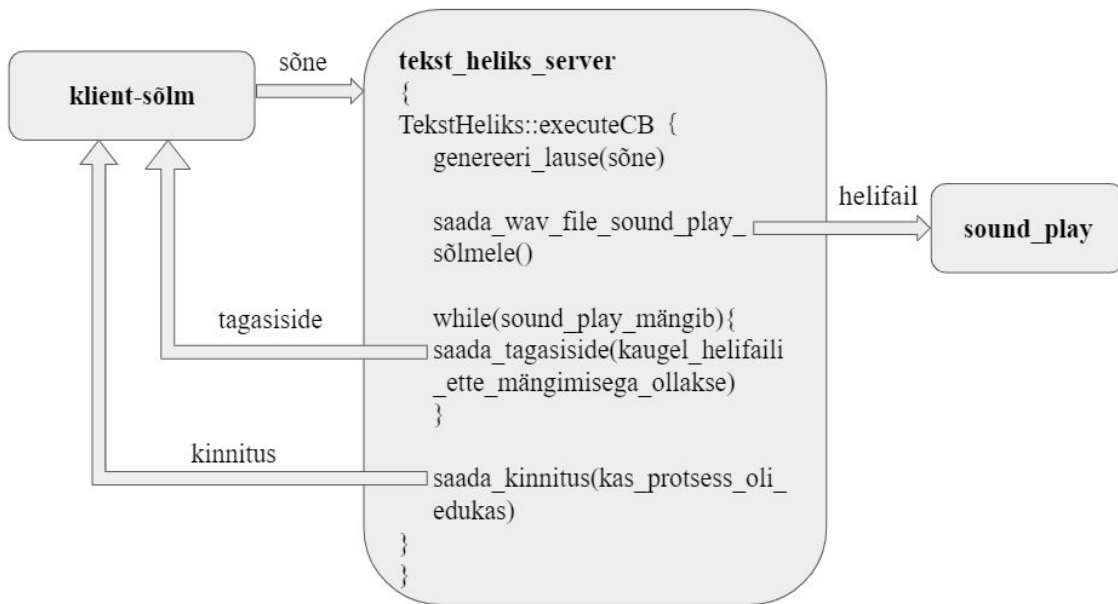
Toiminguserveri ning -kliendi vaheline suhtlus on määratletud `action`-kaustas olevas `lausu_fraas.action` failis (joonis 9). Toimingu eesmärgiks on `string`-tüüpi muutuja nimega fraas. Tulemus on defineeritud tõeväärusena nimega `edukas`. Tagasiside on `float32`-tüüpi muutuja nimega `progress`. Tagasiside on täisarv, kui saadetakse helifaili pikkust sekundites. Kui saadetakse tagasisidet selle kohta, kui kaugel ollakse helifaili ettemängimisega, siis on saadetavad väärtused vahemikus null kuni üks. Serveri töö algab, kui saadakse uus fraas.

```
1 #goal definition
2 string fraas
3 ---
4 #result definition
5 bool edukas
6 ---
7 # feedback
8 float32 progress
```

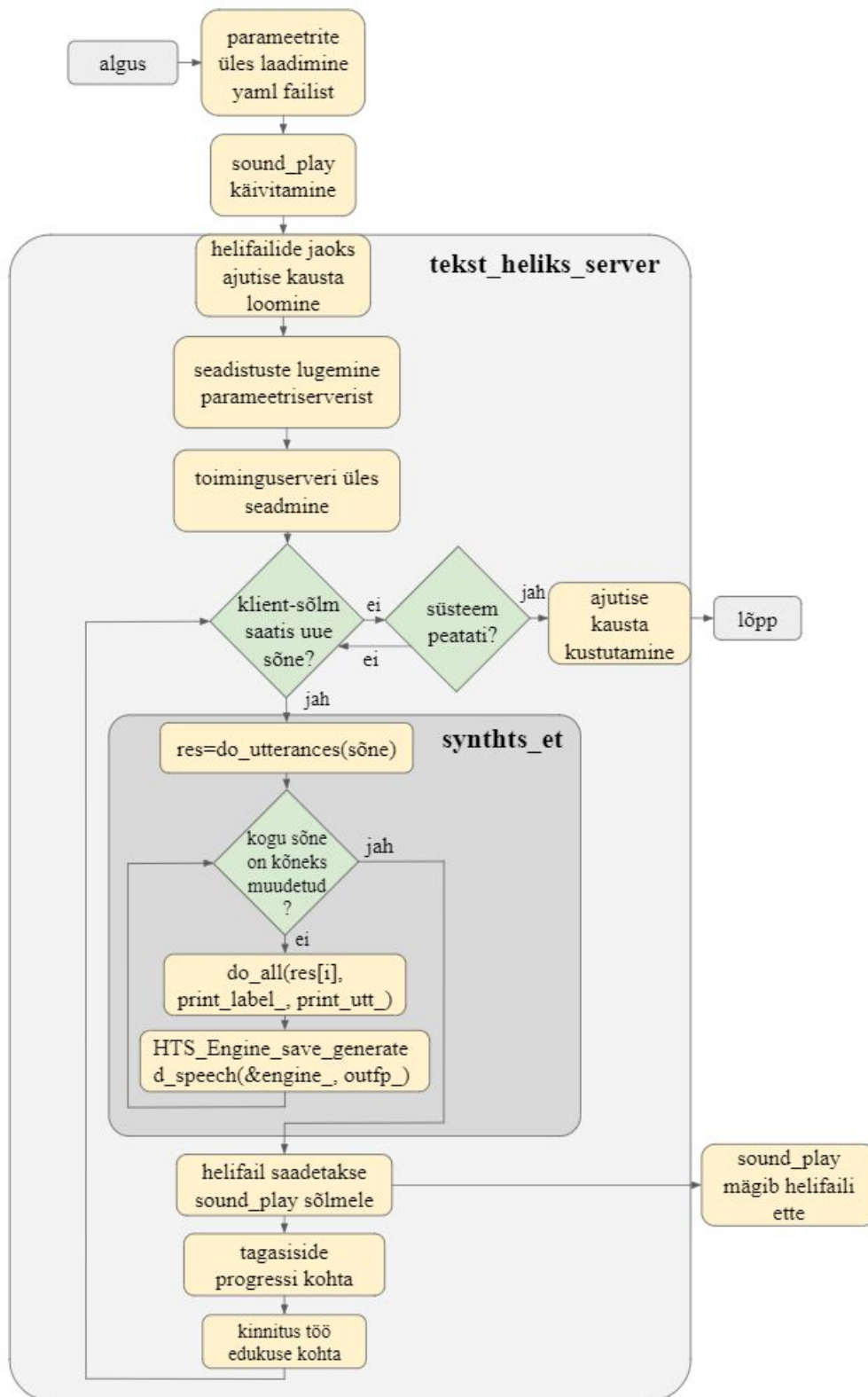
Joonis 9. *lausu_fraas.action*.

Kui toiminguserver saab kliendilt uue sõne, kutsutakse välja `TekstHeliks::executeCB` meetod (joonis 10). Selles meetodis kutsutakse välja serveri ülesseadmisel loodud `KeeleTegemise` objekti `genereeri_lause` meetod, mis võtab sisendiks kliendilt saadud sõne ning kasutades `synthts_et` programmi funktsioone ning muutujaid, genereerib eestikeelset kõnet sisaldava WAV-tüüpi helifaili. Helifail tekitatakse parameetri serveris ette määratud asukohta. Meetod `genereeri_lause` tagastab tõeväärtuse, millega saab kontrollida, kas helifail sai edukalt loodud.

Seejärel arvutatakse välja helifaili pikkus sekundites ning saadetakse tagasisidena kliendile. Helifail edastatakse `sound_play` kimbule, mis hakkab eestikeelset kõnet ette mängima. Samaaegselt, kui `sound_play` töötab, hakkab server iga sekund saatma kliendile tagasisidet, kui kaugel ollakse helifaili mängimisega. Helifaili lõppu jõudes saadetakse kliendile kinnitus, kas ettemängimine oli edukas ning server jääb uut sõne ootama. Kogu `synthts_et_ros` loogika üldskeem (joonis 10) ning `synthts_et_ros` protsessidiagramm (joonis 11) on illustreeritud joonistel 10 ja 11.



Joonis 10. *synthts_et_rosi* loogika üldine skeem.



Joonis 11. *synthts_et_rosi* protsessidiagramm.

4.4.3 Serveri sulgumine

Serveri sulgemisel kutsutakse välja *shutdown* meetod, mis lõpetab kõne sünteesimisega seotud protsesside töö. Lisaks kustutatakse tekstHeliks-kaust, mis loodi serveri ülesseadmisel.

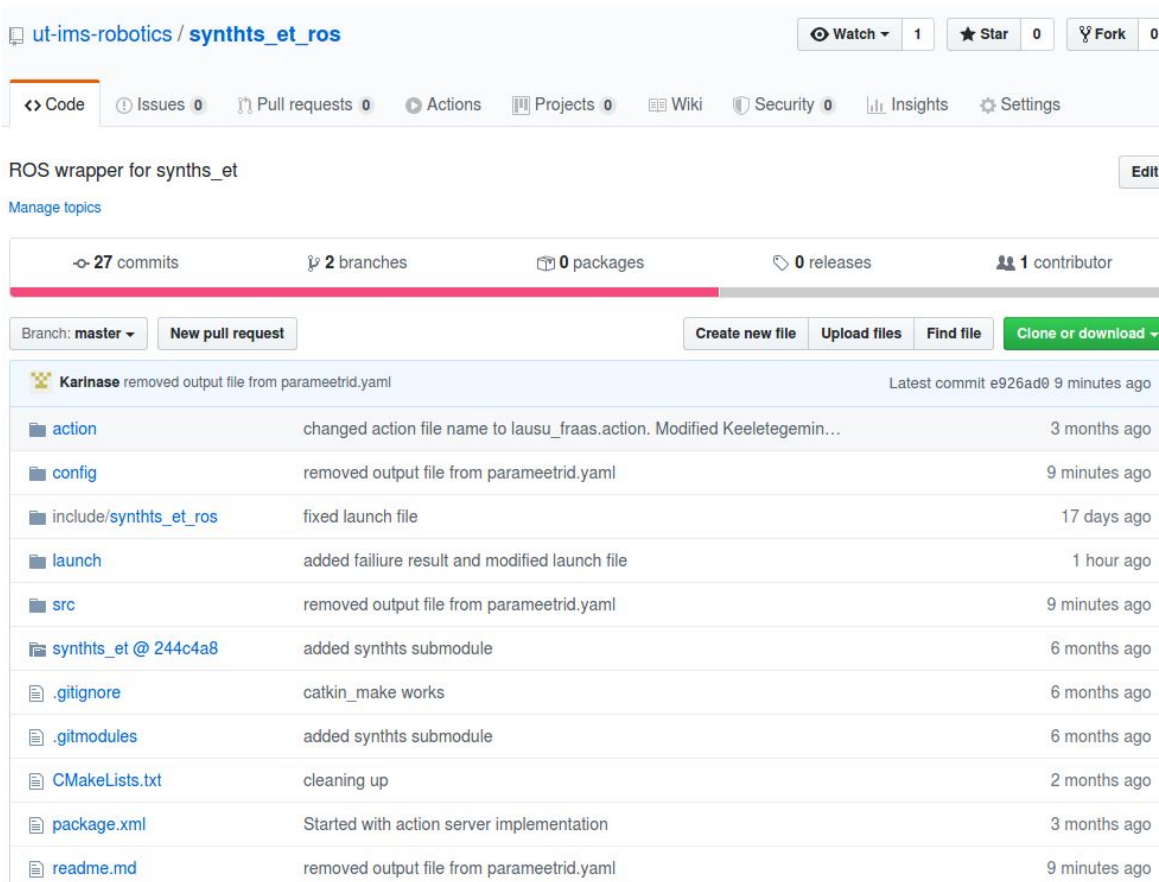
4.4.3 Serveri kiirkäivitamine

Toiminguserveri tekst_heliks_serveri kiirkäivitamiseks loodi ROSi käivitusfail (*launch file*), mis võimaldab kogu süsteemi käsurealt hõlpsasti käivitada. Kiirkäivitusfaili abil käitatakse nii sound_play kimbu play.py sõlm kui ka tekst_heliks_server ning laaditakse konfiguratsioonifail parameetriserverisse. Ühtlasi on synthts_et_ros kimpu loodud ka näidiseks klient-sõlm tekst_heliks_klient, mille järgi saab iga robotisüsteemi arendaja modelleerida oma nõuetele vastava kliendi.

5. Tulemus

5.1. Tulemi kirjeldus

Uurimistöö käigus valmis pakend `synthts_et_ros` (joonis 12), mis võimaldab teksti kõneks konverteerida, kasutades seejuures `synthts_et` programmi loogikat alusena. ROSi abiga on lahendus tehtud kättesaadavaks võimalikult paljudele robotitele. Pakendis on toiminguserver, mis võtab vastu saabuvasid tellimusi ning täidab neid. Helifailid mängitakse automaatselt ette kasutades `sound_play` kimpu. Olemas on ka näidisklient, mis demonstreerib, kuidas saata toiminguserverile päringuid. Seega on loodud modulaarne lahendus, mis võimaldab lisada robotitele eestikeelse kõne sünteesimise funktsionaalsuse.



ut-ims-robotics / `synthts_et_ros` Watch 1 Star 0 Fork 0

<> Code Issues 0 Pull requests 0 Actions Projects 0 Wiki Security 0 Insights Settings

ROS wrapper for `synthts_et` Edit

Manage topics

27 commits 2 branches 0 packages 0 releases 1 contributor

Branch: master New pull request Create new file Upload files Find file Clone or download

Commit	Description	Time
Karinase removed output file from <code>parameetrid.yaml</code>		Latest commit e926ad9 9 minutes ago
action	changed action file name to <code>lausu_fraas.action</code> . Modified Keeletegemin...	3 months ago
config	removed output file from <code>parameetrid.yaml</code>	9 minutes ago
include/synthts_et_ros	fixed launch file	17 days ago
launch	added failiure result and modified launch file	1 hour ago
src	removed output file from <code>parameetrid.yaml</code>	9 minutes ago
synthts_et @ 244c4a8	added <code>synthts</code> submodule	6 months ago
.gitignore	catkin_make works	6 months ago
.gitmodules	added <code>synthts</code> submodule	6 months ago
CMakeLists.txt	cleaning up	2 months ago
package.xml	Started with action server implementation	3 months ago
README.md	removed output file from <code>parameetrid.yaml</code>	9 minutes ago

Joonis 12. `synthts_et_ros` GitHubis.

5.2 Paigaldamine

Lahendus on mõeldud Linuxi distributsioonidele ning on arendatud kasutades ROS Kinetic'ut. Tarkvara paigaldamiseks on tarvis kõigepealt seadistada ROSi catkin töökeskkond ROSi kimpude lähtekoodi kompileerimiseks. Loodud töökeskkonda tuleb alla laadida pakendi `synthts_et_ros` lähtekood, mis sisaldab alammoodulina `synthts_et` tarkvara. Pärast catkin-töökeskkonna kompileerimist saab tarkvara toimimist testida muutes parameetrite faili vastavalt vajadusele ning käivitades `parameetrid.launch` faili ja sõlme `tekst_heliks_klient`. Kogu loodud tarkvara ja täpsem juhend asub GitHubi lehel:

https://github.com/ut-ims-robotics/synthts_et_ros [50].

5.3. Teadaolevad piirangud

Hetkeses lahenduses on tagatud on ainult HTS-kõnesüntees. Teisi eestikeelse kõne sünteesimise mooduseid ei ole lisatud antud lahendusesse.

Lahendus on kirjutamise seisuga võimeline mängima kuni kümne sekundi pikkusega helifaile `sound_play` poolt kasutatavate draiverite tarkvara piirangute tõttu [51]. Programmi poolt genereeritava WAV-faili pikkus võib olla aga pikem.

6. Kokkuvõte

Eestis tegeletakse aktiivselt kõnetehnoloogiate arendamisega. Nende rakendamine robotitel on olnud aga minimaalne. Lõputöö eesmärk oli luua modulaarne moodus, mis võimaldaks eestikeelse kõne sünteesimise funktsionaalsust lisada võimalikult paljudele robotitele.

Lõputöö tulemusena valmis töötav ROSi kimp `synthts_et_ros`, mis võimaldab etteantud teksti helifailiks konverteerida ning ette mängida. ROSi abiga on loodud lahendus tehtud kättesaadavaks võimalikult paljudele robotitele. Lahendus sisaldab arendajale ka näidis klient-sõlme.

Edasiste sammudena on võimalik lisada erinevaid kõne sünteesimise meetodeid peale HTS-kõnesünteesi. See võimaldaks erinevaid meetodeid efektiivsemalt omavahel võrrelda. Lisaks oleks võimalik ka lisada kõnest arusaamise funktsionaalsusi.

Kirjandus

- [1] „Industrial robot“, *Wikipedia*. märts 28, 2020, Vaadatud: mai 03, 2020. [Online]. Available at: https://en.wikipedia.org/w/index.php?title=Industrial_robot&oldid=947783022.
- [2] „Robot-assisted surgery“, *Wikipedia*. apr 17, 2020, Vaadatud: mai 03, 2020. [Online]. Available at: https://en.wikipedia.org/w/index.php?title=Robot-assisted_surgery&oldid=951497696.
- [3] „Domestic robot“, *Wikipedia*. märts 25, 2020, Vaadatud: mai 03, 2020. [Online]. Available at: https://en.wikipedia.org/w/index.php?title=Domestic_robot&oldid=947228722.
- [4] New China TV, *A smarter Shanghai Library*. .
- [5] Munich Airport, „Josie Pepper - A humanoid robot with artificial intelligence“. <https://www.munich-airport.com/hi-i-m-josie-pepper-3613413> (vaadatud apr 09, 2020).
- [6] E. Wynne, „Meet the robotic new guide at the WA Art Gallery“, *ABC News*, mai 18, 2016. <https://www.abc.net.au/news/2016-05-18/art-gallery-of-wa-introduces-robot-tour-guide-aggie/7424760> (vaadatud apr 09, 2020).
- [7] „Kõnesüntees“. https://www.eki.ee/heli/index.php?option=com_content&view=article&id=6&Itemid=465 (vaadatud apr 09, 2020).
- [8] „Amazon Alexa“, *Wikipedia*. apr 30, 2020, Vaadatud: mai 04, 2020. [Online]. Available at: https://en.wikipedia.org/w/index.php?title=Amazon_Alexa&oldid=954056067.
- [9] „Google Assistant, your own personal Google“, *Assistant*. <https://assistant.google.com/> (vaadatud mai 04, 2020).
- [10] „Siri“, *Apple*. <https://www.apple.com/siri/> (vaadatud mai 04, 2020).
- [11] „Cortana - Your personal productivity assistant“, *Cortana - Your personal productivity assistant*. <https://www.microsoft.com/en-us/cortana> (vaadatud mai 04, 2020).
- [12] „Amazon.com: Fancy Nancy Shall We Be Fancy, 15" Talking Doll, 35+ Phrases, Colorful Lights & Music: Toys & Games“. <https://www.amazon.com/Fancy-Nancy-Talking-Phrases-Colorful/dp/B07BW7THT9/re>

- f=sr_1_1?dchild=1&keywords=talking+doll&qid=1586256971&sr=8-1 (vaadatud apr 09, 2020).
- [13] „Amazon.com: Talking Hamster Plush Toy, Repeat What You Say Funny Kids Stuffed Toys, Talking Record Plush Interactive Toys for, Birthday Gift Kids Early Learning: Toys & Games“.
<https://www.amazon.com/Talking-Hamster-Interactive-Birthday-Learning/dp/B074VZXXM1> (vaadatud apr 29, 2020).
- [14] „Amazon.com: Wenjuan Remote Control Intelligent Robot Toy Walking Talking Dancing Toy Robots for Kids, Sings, Reads Stories, Interactive Educational RC Robots Toys for 3 4 5 6 7 8 9 Year Old Boys and up: Toys & Games“.
https://www.amazon.com/Wenjuan-Control-Intelligent-Interactive-Educational/dp/B085W62SDX/ref=sr_1_87?dchild=1&keywords=talking+toys&qid=1588117982&sr=8-87 (vaadatud apr 29, 2020).
- [15] anews, „Troika robot speaks 4 languages“, *anews*.
<http://www.anews.com.tr/webtv/life/troika-robot-speaks-4-languages> (vaadatud apr 09, 2020).
- [16] „Robots to assist tourists at South Korean airport - Taipei Times“, juuli 05, 2017.
<https://www.taipeitimes.com/News/world/archives/2017/07/05/2003673948> (vaadatud apr 22, 2020).
- [17] *The Smithsonian's newest guide is a robot.* .
- [18] G. Zheng, S. Bi, H. Min, K. Yang, ja Y. Zhang, „Design and Implementation of Chinese Speech Robot Control System Based on Android Embedded Platform“, *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*, juuli 2017, lk 498–503, doi: 10.1109/CYBER.2017.8446468.
- [19] *Elder Care Robots.* .
- [20] „Roomba“, *iRobot | Intelligentsed ja innovatiivsed robotid*.
<https://www.irobot.ee/roomba/> (vaadatud mai 04, 2020).
- [21] „Braava“, *iRobot | Intelligentsed ja innovatiivsed robotid*. <https://www.irobot.ee/braava/> (vaadatud mai 04, 2020).
- [22] „Telepresence“, *Wikipedia*. veebr 28, 2020, Vaadatud: apr 16, 2020. [Online]. Available at: <https://en.wikipedia.org/w/index.php?title=Telepresence&oldid=943069603>.

- [23] „iRobot Ava 500 autonomous telepresence robot is designed for chatting“, *New Atlas*, juuni 13, 2013. <https://newatlas.com/irobot-ava-500/27865/> (vaadatud apr 22, 2020).
- [24] „Double Robotics - Telepresence Robot for Telecommuters“. <https://www.doublerobotics.com/double3.html> (vaadatud apr 30, 2020).
- [25] „Double Robotics Double 3 Telepresence Robot“. https://www.bhphotovideo.com/c/product/1508570-REG/double_robotics_1013_double_2_robot.html (vaadatud apr 30, 2020).
- [26] „Tehisintellekt Elisas | Elisa Eesti“. <https://www.elisa.ee/et/kratid> (vaadatud apr 09, 2020).
- [27] „Keeleinstituudi nutirakendus loeb uudiseid ette“, *Tehnika*, sept 10, 2012. <https://tehnika.postimees.ee/967396/keeleinstituudi-nutirakendus-loeb-uudiseid-ette> (vaadatud apr 30, 2020).
- [28] „Kuula: robot loeb nüüd veebilehti ette“, *Tehnika*, aug 15, 2019. <https://tehnika.postimees.ee/6753889/kuula-robot-loeb-nuud-veebilehti-ette> (vaadatud apr 30, 2020).
- [29] „Radioloogid juba räägivad arvutiga ja see mõistab neid, kõigi raskuste kiuste“, *Digigeenius*. <https://digi.geenius.ee/rubriik/uudis/radioloogid-juba-raagivad-arvutiga-ja-see-moistab-neid-koigi-raskuste-kiuste/> (vaadatud apr 30, 2020).
- [30] „Eestikeelne puldiga juhitud robot Robert“. <https://www.karupoegpuhh.ee/puldiga-juhitav-robot-robert> (vaadatud apr 09, 2020).
- [31] E. T. | ERR, „TÜ robottudeng hakkas semestri lõpuks eesti keelt rääkima“, *ERR*, juuni 29, 2018. <https://novaator.err.ee/843162/tu-robottudeng-hakkas-semestri-lopuks-eesti-keelt-raakima> (vaadatud apr 09, 2020).
- [32] M. Nurmsaar ja E. Halling, „Eesti keeles suhtlev sotsiaalse käitumisega robot“, juuni 2017, Vaadatud: apr 09, 2020. [Online]. Available at: <https://digikogu.taltech.ee/testimine/en/Item/d5244dda-5b05-4d09-90b8-c6b7d5b894a0>.
- [33] C. Woodford, „How speech synthesis works“, *Explain that Stuff*. <http://www.explainthatstuff.com/how-speech-synthesis-works.html> (vaadatud apr 09, 2020).
- [34] „Eesti keele käsiraamat“. <https://www.eki.ee/books/ekk09/index.php?p=2&p1=5>

- (vaadatud apr 09, 2020).
- [35] „Eestikeelse närvivõrkudel põhineva kõnesünteesi esialgsed tulemused“.
<http://www.tarkvaralabor.ee/deepvoice3/> (vaadatud apr 09, 2020).
- [36] „Festival“.
<http://www.cstr.ed.ac.uk/projects/festival/> (vaadatud apr 09, 2020).
- [37] I. Kiissel, *ikiissel/synthts_et*. 2019.
- [38] „eSpeak: Speech Synthesizer“.
<http://espeak.sourceforge.net/> (vaadatud apr 23, 2020).
- [39] „Formant“, *Vikipeedia*. juuli 18, 2019, Vaadatud: apr 23, 2020. [Online]. Available at:
<https://et.wikipedia.org/w/index.php?title=Formant&oldid=5396117>.
- [40] „ROS.org | About ROS“.
<https://www.ros.org/about-ros/> (vaadatud mai 04, 2020).
- [41] „ROS.org | Is ROS For Me?“
<https://www.ros.org/is-ros-for-me/> (vaadatud mai 19, 2020).
- [42] „ROS/Introduction - ROS Wiki“.
<http://wiki.ros.org/ROS/Introduction> (vaadatud apr 09, 2020).
- [43] „Packages - ROS Wiki“.
<http://wiki.ros.org/Packages> (vaadatud mai 18, 2020).
- [44] „What is the concept of ROS wrapper? - ROS Answers: Open Source Q&A Forum“.
<https://answers.ros.org/question/225518/what-is-the-concept-of-ros-wrapper/> (vaadatud apr 09, 2020).
- [45] „catkin/CMakeLists.txt - ROS Wiki“.
<http://wiki.ros.org/catkin/CMakeLists.txt> (vaadatud mai 19, 2020).
- [46] M. Quigley *et al.*, „ROS: an open-source Robot Operating System“, lk 6.
- [47] „actionlib_tutorials/Tutorials/SimpleActionServer(ExecuteCallbackMethod) - ROS Wiki“.
http://wiki.ros.org/actionlib_tutorials/Tutorials/SimpleActionServer%28ExecuteCallbackMethod%29 (vaadatud mai 19, 2020).
- [48] „actionlib_tutorials/Tutorials/SimpleActionClient - ROS Wiki“.
http://wiki.ros.org/actionlib_tutorials/Tutorials/SimpleActionClient (vaadatud mai 19, 2020).
- [49] „sound_play - ROS Wiki“.
http://wiki.ros.org/sound_play (vaadatud apr 16, 2020).
- [50] *ut-ims-robotics/synthts_et_ros*. Robotics at IMS Lab, 2020.
- [51] „Is it possible to play whole .wav file · Issue #96 · ros-drivers/audio_common“, *GitHub*.
https://github.com/ros-drivers/audio_common/issues/96 (vaadatud apr 30, 2020).

Lihtlitsents lõputöö reprodutseerimiseks ja üldsusele kättesaadavaks tegemiseks

Mina, Karina Sein

1. annan Tartu Ülikoolile tasuta loa (lihtlitsentsi) minu loodud teose **“Eestikeelse kõnesünteesi võimaldamine robotika arendusplatvormil ROS”**, mille juhendaja on Karl Kruusamäe 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'i 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.

Karina Sein

20.05.2020