

RAPORT TEHNICO-ȘTIINȚIFIC

Dan Tufiș, Verginica Barbu Mititelu, Radu Ion, Elena Irimia, George Cioroiu, Eric Curea
Mihai Dascălu, Stefan-Trășan-Matu, Traian Rebedea

Rezumat

În 2018, Institutul de Ceretari pentru Inteligență Artificială a fost implicat în două proiecte componente ale proiectului complex ROBIN: ROBIN-Social și ROBIN-Dialog (pe care l-a și coordonat).

O primă etapă a cercetărilor noastre s-a referit la analiza literaturii de specialitate în privința sistemelor de dialog și a capabilităților incorporate ale robotului Pepper. De asemenea, s-a precizat noțiunea de micro-lume pentru care proiectul va implementa arhitectura funcțională de prelucrare. Au fost definite specificații funcționale ale sistemului și micro-lumilor și comenzile primitive de pseudo-limbaj necesare pentru interacțiunea cu robotul.

Activitatea tehnico-științifică a echipei noastre în cadrul proiectului s-a concentrat pe crearea și testarea resurselor și a modelelor acustice și de limba necesare dialogului în limba română cu robotul Pepper. Acest lucru a presupus mai întâi definirea universurilor de discurs pentru care Pepper va avea competența dialogului în limba română.

Pentru a construi un lexicon și un model de limbă relevante, au fost imaginate patru scenarii de dialog - experimente wizzard-of-oz - (asistent de vânzări într-un magazin de electronice, companion pentru o persoană necesitând asistență, partener de dialog "promoțional" și ghid pentru vizite la centrul de cercetare PRECIS).

Cuvintele din dialogurile respective au fost prelucrate (lematizate și adnotate morfo-lexical) și au fost stocate într-un lexicon preliminar. Pentru fiecare cuvânt din lexiconul preliminar au fost selectate câte 10 cuvinte cu semantică distribuțională similară. Acest lucru a fost posibil folosind vectorii semantici (word-embeddings) construiți din corpusul de referință al limbii române CoRoLa (Barbu Mititelu et al., 2018), (Păiș, Tufiș, 2018). Etapa următoare a extensiei dicționarului vă lua în considerare includerea sinonimilor, hiponimilor și a hiperonimilor cuvintelor existente în lexicon (în cazul în care aceștia nu sunt deja incluși). Această extensie se bazează pe ontologia lexicală Romanian WordNet (Tufiș, Barbu Mititelu, 2014). În condițiile definirii de noi universuri de discurs, lexiconul și modelul de limbă vor fi actualizate corespunzător, prin automatizarea întregului proces.

Modelul acustic inițial, specific limbii române, se bazează pe o serie de înregistrări vocale (circa 105 ore) și a fost testat în contextul unei aplicații de "keyword-spotting" (Pipa, Boroș, 2016). Acest model acustic (bazat actualmente pe tri-foneme) urmează a fi adaptat (probabil folosind di-foneme, pentru a satisface limitele puterii de calcul a robotului Pepper) și integrat cu modelul de limbă dezvoltat din textele scenariilor create. Lexiconul existent, a trebuit îmbogățit cu informație

specifică aplicațiilor de prelucrare a vorbirii (împărțire în silabe, marcarea accentului și transcrierea fonetică pentru fiecare cuvânt).

Obiectivele proiectului component ROBIN-Dialog au fost:

Activitatea: Act 1.12 - Definirea specificațiilor funcționale și arhitecturale ale modulelor software și micro-lumilor

Categorie activitate: A1 - Cercetare fundamentală Indicatori de realizare: Specificații funcționale ale sistemului și micro-lumilor

Obiectivul a fost integral îndeplinit, s-au definit micro-lumile țintă și structura de principiu a sistemului de dialog în microlumile respective

Activitatea: Act 1.13 - Construirea lexiconului/lexicoanelor de cuvinte și a expresiilor reprezentative pentru micro-lumea/lumile țintă

Categorie activitate: A1 - Cercetare fundamentală Indicatori de realizare: Lexicon în format electronic - descrierea lexiconului

Obiectivul a fost integral îndeplinit, a fost construit lexiconul micro-lumilor țintă. Pe baza scenariilor dezvoltate în micro-lumile vizate de proiect, a fost dezvoltat un lexicon inițial, conținând 587 de forme (corespunzând unor 318 perechi leamnă/parte de vorbire). Formatul lexiconului în această etapă a proiectului este de tip tabular (cu coloane separate prin tab-uri), în care pe prima coloană se regăsește forma ocurență a cuvântului, pe a doua se regăsește lema (forma de dicționar) iar pe a treia observăm descrierea (eticheta) morfo-sintactică a cuvântului (verb, modul indicativ, timpul prezent, persoana a 3-a, numărul plural). Setul inițial de leme/forme a trecut prin mai multe etape de extindere, pentru a crea un lexicon cât mai comprehensiv, care să ajute la adaptarea sistemului de dialog la diverse variante de exprimare a aceluiași conținut semantic. Într-o primă etapă, pornind de la acest set inițial de leme/forme, s-au extras din corpusul COROLA (Mititelu et al., 2018) reprezentări vectoriale învățate automat, cunoscute și ca “word embeddings” (Paiș and Tufiș, 2018). Acestea sunt reprezentări vectoriale dense în spații vectoriale de mici dimensiuni, care pot fi folosite în aplicații de prelucrare a limbajului natural.

Activitatea: Act 1.14 - Extensia automată a lexiconului/lexicoanelor creat(e) manual la 4-1-2. Validarea și corectarea lexiconului/lexicoanelor extins(e)

Categorie activitate: A1 - Cercetare fundamentală Indicatori de realizare: L10 - Descriere lexicoane de cuvinte extinse automat, lexicon extins

Obiectivul a fost integral îndeplinit, lexiconul micro-lumilor a fost extins automat, dar etapele intermediare ale extensiei au fost validate manual. În contextul proiectului nostru, vectorii semantici au fost folosiți pentru a identifica cuvinte similare (ca încărcătură semantică) cu cuvintele din setul inițial de leme. Vectorii de similaritate au fost generați atât pentru cuvinte în forma lor ocurență cât și în forma dicționar și au fost validați manual pentru a îndepărta eventualele cuvinte care se îndepărtează foarte mult de universul micro-lumilor cu care lucrăm. Toate cuvintele rezultate din această primă etapă de extindere, au trecut printr-un proces de identificare în RoWordnet (vezi referință) a hiperonimelor (unitate lexicală cu semnificație mai generală și mai extinsă față de cea de la care se

pornește) și sinonimelor asociate. Cuvintele rezultate au fost validate manual și introduse (eliminând duplicații) la rândul lor în lexicon. Ultima etapă este cea de asociere a informației de leamnă și etichetă morfologică pentru cuvintele colectate și de extindere a lexiconului cu toate cuvintele din familiile paradigmatică ale fiecărei leme deja existente, făcând apel la lexiconul dezvoltat la ICIA, tbl.wordform.ro (o resursă cu peste 1.150.000 de intrări, de forma <formă ocurentă>tab<lemă>tab<etichetă morfo-sintactică>). Această ultimă etapă nu necesită validare manuală, deoarece tbl.wordform.ro este o resursă care a trecut prin etape de validare pe măsură ce a fost îmbogățită. Într-o etapă ulterioară a proiectului (etapa II, livrabilul 10), lexiconul va fi îmbogățit cu informația de transcriere fonetică, silabație și poziționare a accentului.

Activitatea: Act 1.15 - Diseminare

Categorie activitate: D1 - Activități suport - Diseminarea pe scară largă prin comunicarea și publicarea națională sau internațională a rezultatelor Indicatori de realizare: 1 lucrare științifică

Obiectivul a fost integral îndeplinit, fiind prezentată la **Conferința internațională LREC 2018** (conferință ISI) de la Miyazaki, Japonia, lucrarea Dan Tufiș, Dan Cristea A Bird's-eye View of Language Processing Projects at Romanian Academy, Miyazaki, Japan, LREC 2018, pp. 2445-2451, ISBN 979-10-95546-00-9.

De asemenea, la **Conferința internațională "Linguistic Resources and Tools for Processing of the Romanian Language"**, a fost prezentată lucrarea Nenciu, B., Ruseti, S., & Dascalu, M. (2018). Extracting Actions from Romanian Instructions for IoT Devices. In V. Pais, D. Gifu, D. Trandabat, D. Cristea & D. Tufis (Eds.), 13th Int. Conf. on Linguistic Resources and Tools for Processing Romanian Language (ConsILR 2018) (pp. 168–176). Iasi, Romania. Ambele lucrări menționează cu multumiri, finanțarea cercetărilor de către proiectul ROBIN-Dialog

Site proiectului ROBIN-Dialog a fost realizat: <http://www.racai.ro/p/robin/>

Toate obiectivele asumate de proiectul component ROBIN-Dialog au fost îndeplinite complet.

Structura ofertei de servicii de cercetare și tehnologice Institutul de Cercetări pentru Inteligență Artificială (ICIA/RACAI), va oferi pe platforma ERRIS serviciile de cercetare și tehnologice enumerate în tabelul următor (<https://erris.gov.ro/RACAI-ICIA>):

Servicii:

Interogare corpus de referință al limbii române corola.racai.ro; TTL <http://ws.racai.ro/ttlws.wsdl>, Modular Language Processing for Lightweight Applications (MPLA) cu prelucrări pentru mai mult de 40 de limbi <http://slp.racai.ro/index.php/mlplanew/>; Sistemul ROBIN-dialog de prelucrare a dialogurilor în micro-lumile țintă; Lexicon extins pentru aplicații de prelucrare a vorbirii; Sistem de detecție de cuvinte cheie în conversații înregistrate <http://heimdall.racai.ro/>

Romanian Spoken Language Processing; rsp.racai.ro Romanian Anonymous Speech Corpus rasp.racai.ro.

Laboratorul de cercetare din cadrul Facultății de Automatică și Calculatoare (UPB) implicat în cadrul proiectului ROBIN-Dialog se regăsește la <https://erris.gov.ro/PRECIS-UPB> - laboratorul 12 (Computer Based Innovation and Collaborative Knowledge Development).

Locuri susținute de acest proiect:

7 cercetători cu vechime în ICIA plus 2 tineri cercetători angajați pe proiect.

3 cercetători cu vechime în UPB plus 2 tineri cercetători angajați pe proiect.

In cadrul acestui proiect component cecurile nu au fost valorificate iar sumele corespunzatoare au fost realocate cheltuielilor de management.

Introducere

Sistemele de dialog conțin, în general, cinci componente principale, așa cum evidențiază Figura 1: ASR (Automatic Speech Recognition-recunoașterea automată a vorbirii), NLU (Natural Language Understanding-înțelegerea limbajului natural), Dialog Manager, NLG (Natural Language Generation-generarea limbajului natural), and TTS (Text To Speech synthesis). Pe scurt, componenta ASR responsabilă pentru preluarea semnalului vocal și convertirea sa într-o secvență de cuvinte.. Componenta NLU are menirea de a înțelege și a formaliza semnificația secvenței de cuvinte produsă de ASR. Reprezentarea formală generată de NLU este transmisă Managerului de Dialog, care este componenta responsabilă de răspunsul sistemului de dialog la secvența sonoră de intrare. Spre deosebire de celelalte componente reprezentate în Figura 1, Managerul de Dialog are memoria interacțiunilor anterioare, proprietate esențială pentru asigurarea unui comportament cooperant și rațional într-o conversație cu replici alternative (conversational turns). Răspunsul generat de Managerul de Dialog poate fi reprezentarea unui comenzi efectorii/mortice ca reacție la o comandă de acțiune sau a unui răspuns, sau cerere de clarificare la o solicitare de informații. Reprezentarea comenzii efectorii (în cazul unui robot cu posibilități motrice) este transmisă modulului de comandă motrice al robotului (nereprezentat în Figura 1). Reprezentarea semantică a răspunsului textual este transmisă componentei NLG care o transformă într-o secvență de cuvinte inteligibilă de factorul uman. Ultima parte a lanțului de interacțiune este conversia textului produs de NLG într-o rostire vocală. Această prelucrare este responsabilitatea modulului TTS (Text to Speech).

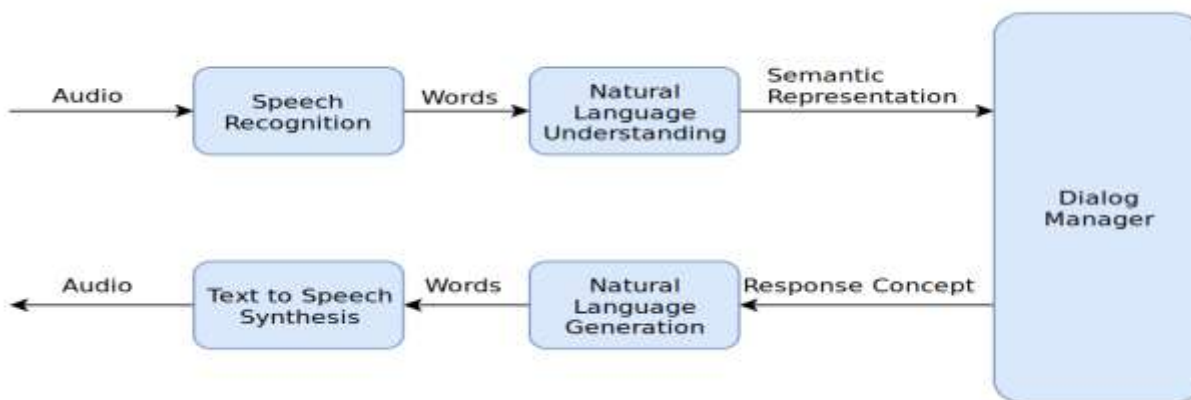


Figura 1: Arhitectura abstractă a unui sistem de dialog

Sisteme de dialog bazate pe scenarii

Sistemele de dialog în limbaj natural sunt programe de calculator care conversează cu utilizatorul într-o limbă dată (de ex. în limba română) cu scopul de a furniza informații. De exemplu, există sisteme de dialog care ajută utilizatorul să înțeleagă cum se rezolvă o problemă de fizică (Rus et

al., 2013), să cumpere un produs (Yan et al., 2017) sau chiar să asiste doctorul la punerea unui diagnostic (Liu et al., 2018), prin conversația directă cu bolnavul.

Aceste sisteme de dialog sunt foarte specializate astfel încât ele funcționează numai în scenarii predefinite¹. Într-un astfel de scenariu, din punctul de vedere al programului, dialogul se află într-o stare bine definită după fiecare replică primită de la utilizator: aceeași dacă programul nu a înțeles ce a spus utilizatorul sau o stare nouă dacă sistemul de dialog a analizat corect spusele utilizatorului și a deslușit calea de urmat în graful care formalizează structura dialogului. În momentul în care sistemul de dialog ajunge la un nod din care nu mai poate avansa (nu mai există arce care să iasă din nod), dialogul s-a încheiat cu succes iar sistemul de dialog poate executa acțiunile asociate cu această stare finală (de exemplu să afișeze diagnosticul sau produsul recomandat).

O micro-lume reprezintă o specificație a mediului și a scenariului în care se desfășoară dialogul. Pentru sistemele de dialog exemplificate mai sus, mediul în care se desfășoară dialogul este calculatorul iar acesta nu execută nicio acțiune în afară de a furniza informațiile căutate de utilizator. Însă, în cazul în care sistemul de dialog se atașează unui robot care poate executa acțiuni (se poate deplasa, poate ridica obiecte, poate identifica persoane, etc.), atunci când programul de dialog a atins o stare finală care are asociate acțiuni specifice, robotul le poate executa, *în mediul pe care îl cunoaște*.

Să luăm de exemplu micro-lumea „căutării persoanei X în sala de clasă” de către un robot care se poate mișca prin clasă și, cu ajutorul camerei sale video, poate identifica o persoană. Această micro-lume trebuie să precizeze/ofere:

- Ce persoane se află în clasă
- În ce bancă stă fiecare persoană
- Coordonatele (x, y) ale fiecărei bănci astfel încât robotul să poată ști unde să se ducă
- O imagine (sau mai multe) în format electronic pentru fiecare persoană astfel încât programul de recunoaștere facială să poată fi antrenat.
- O structură (arborescentă) a dialogului de urmat:
 - o Formula de salut/start dialog: „Salut Pepper!”, „Bună Pepper!”, etc.
 - o Cererea principală: „F a venit?” (numai numele de familie F e precizat)
 - Robot (în cazul în care sunt mai multe persoane cu același nume de familie F): „Vă referiți la X sau Y?”
 - Utilizator: „La X.”
 - o Stare finală: robotul se poate deplasa la banca în care stă F X, îl/o poate recunoaște și se poate întoarce pentru a confirma prezența.
 - Stare finală (în cazul în care există o singură persoană cu numele de familie F): robotul se poate deplasa la banca în care stă F X, îl/o poate recunoaște și se poate întoarce pentru a confirma prezența.

¹ Încă nu există un sistem de dialog universal care să poată conversa cu utilizatorul uman despre un subiect arbitrar, fără niciun scenariu de dialog. Acesta este dezideratul Inteligenței Artificiale, conform unuia dintre teoreticienii celebri ai domeniului, Alan Turing (https://en.wikipedia.org/wiki/Turing_test).

- Cererea principală: „F X a venit?” (numele complet e precizat)
 - Stare finală (în cazul în care există o singură persoană cu numele de familie F): robotul se poate deplasa la banca în care stă F X, îl/o poate recunoaște și se poate întoarce pentru a confirma prezența.

O micro-lume poate exista în realitate (ca în exemplul de mai sus) sau poate fi simulată pe calculator, caz în care devine virtuală (vezi, de exemplu, jocurile 3D interactive în care personajul controlat de utilizator interacționează verbal cu personaje controlate de calculator). Oricare ar fi natura unei micro-lumi, inginerul de cunoștințe trebuie să se asigure că *a enumerat toate modurile în care se poate interacționa cu micro-lumea*. De aici rezultă că, cu creșterea complexității micro-lumii, această enumerare devine din ce în ce mai dificilă, până în momentul în care devine imposibilă pentru „micro-lumea” care este egală cu realitatea, cel puțin cu mijloacele de descriere care există acum.

Exemplificare a interacțiunilor simulate în micro-lumile alese

1. Folosirea robotului ca un asistent la vizite în centrul de cercetare PRECIS:

- Robotul întâlnește oameni pe hol care:

* se prezintă pentru ca robotul să realizeze asocierea între față și nume:

Robot: Salut! / Bună! Eu sunt Pepper. / Mă cheamă Pepper. Tu cine ești? / Pe tine cum te cheamă?

Utilizator: [Salut!] / [Bună!] / [Bună ziua!] / [Bună dimineața!] / [Neața]. (Eu) sunt / Mă cheamă /
... #Își spune doar numele.#

* îl roagă să îi conducă până la o anumită zonă (spre exemplu, laboratorul 308).

Utilizator: Arată-mi drumul spre laboratorul ...!

(Îmi arăți) Unde este /e laboratorul ...?

Cum ajung în laboratorul ...?

Am treabă în laboratorul Poți să-mi spui unde e?

Robot: Urmează-mă! / Vino după mine! (robotul ar trebui să se asigure că persoana îl urmează):

Robot: Mă urmezi? / Vii (după mine)? / Mai vii?

în caz că omul rămâne în urmă, Pepper poate încerca un dialog pentru a întreba dacă activitatea de ghidare mai este încă valabilă:

Robot: Mai mergem în ...? / Mai vrei să mergi în ...? / Te mai conduc în ...?

Utilizator: Da. / Sigur.

Robot: Urmează-mă! / Vino după mine!

* îl roagă să îi transmită un mesaj unei anumite persoane din laborator (prin urmare robotul trebuie să știe cine e acea persoană, să o caute pe hol, să o identifice și să o notifice)

Utilizator: Spune-i ... / (lui ...) (să ...) / (că ...) [Dacă robotul trebuie să transmită un mesaj unei persoane de sex feminin al cărei nume nu se termină în consoană, se va folosi "Spune-i ...". Dacă

este o persoană de sex feminin al cărei nume se termină în consoană sau este o persoană de sex masculin, atunci se va folosi "Spune-i lui ...". [Aici aproape orice se poate comunica! Robotul ar trebui să reproducă mesajul.]

Robot: ..., X spune / (a spus) / zice / (a zis) (să ...) / (că ...). [Atenție la variabila X aici: e numele utilizatorului care i-a dat robotului o sarcină.]

#mini-dialog

utilizator X: Caut-o / caută-l pe Y și roag-o / roagă-l să vină până la mine. Dacă este [la biroul ei / lui] / [așezat/așezată pe scaun], nu o / îl deranja, dar întoarce-te și anunță-mă.

robot: Bună, Y! X te roagă să mergi până la el/ea.

utilizator Y: Bună, Pepper! Spune-i lui X că

Bună, Pepper! (Spune-i lui X că) sunt foarte ocupat/ocupată. Merg mai târziu.

Bună, Pepper! (Spune-i lui X că) vin imediat.

Bună, Pepper! (Spune-i lui X că) trebuie să plec. Spune-i lui X [să-mi dea un telefon]/[să-mi scrie un e-mail][să mă sune][că vorbim mâine] .

Alternativ, dacă robotul nu-l găsește pe Y, revine la X: Nu [l-am găsit] / [am găsit-o] pe Y.

Alternativ, dacă robotul îl găsește pe Y așezat la birou (există în NaoQi o funcție care determină dacă o persoană sta pe un scaun, ALSittingPeopleDetection), revine la X: [X e la el/ea la birou.][X e în laboratorul ...]

* îl roaga să caute dacă există un anumit tip de obiect într-o anumită zonă (spre exemplu, caută cana verde pe catedra din laboratorul 306)

Utilizator:

(Îl/o/îi/le) Vezi ... pe (vre)unde?

Unde este / sunt ...?

Nu (îl / o / le / îi) găsesc Poți să mă ajuți?

Poți să mă ajuți să găsesc / caut ...?

Este ... aici / [în acest laborator] / [în laboratorul acesta / ăsta]?

Robot:

... este / e / [se află] (pe ... #la cazul acuzativ#) (sub ... #la cazul acuzativ#) (lângă ... #la cazul acuzativ#) (în ... #la cazul acuzativ#) (deasupra ... #la cazul genitiv#) (în fața ... #la cazul genitiv#) (în spatele ... #la cazul genitiv#) (dedesubtul ... #la cazul genitiv#) (între ... #la cazul acuzativ# și ... #la cazul acuzativ#).

În plus, dacă robotul este întrerupt (e.g. o altă persoană îl întâmpină) în timp ce execută o activitate (e.g. când caută o persoană pentru a o notifica), ar fi util dacă robotul ar putea explica / detalia acțiunea curentă (e.g. "Salut Pepper. Ce faci?" R: "Salutare. Sunt bine. Îl caut pe Alex.").

2. Asistența unei persoane în vârstă

- Interacțiunea cu robotul pentru a-i cere să afișeze / spună rezultatele măsurătorilor de sănătate (tensiune arterială, puls, greutate, număr de pași) - ar fi de interes să extindem micro-lumea

pentru a cuprinde tipul de masuratoare, cat si perioada pe care se face raportarea (spre exemplu, ultima masuratoare, pe ultima saptamana, pe ultima luna);

Utilizator: (Afișează / Spune(-mi)/Zi(-mi)) Ce tensiune / greutate / puls am (acum / în acest moment) ?

(Afișează / Spune(-mi)/Zi(-mi)) Ce tensiune / greutate / puls [am avut] / [aveam] [ieri] / [alaltăieri] / [săptămâna trecută] / [acum 2 / 3 / 4 zile]?

(Afișează / Spune) Cât am tensiunea / greutatea / pulsul acum?

(Afișează / Spune) Cât [am avut] / [aveam] tensiunea / greutatea / pulsul [ieri] / [alaltăieri] /

[săptămâna trecută]?

(Afișează / Spune) Câți pași am făcut azi / [de ieri până azi] / [săptămâna aceasta / asta] / [în ultimele X zile]?

- Interacțiunea cu robotul pentru a-l întreba care este următoarea activitate din calendar / data următorului exercitiu fizic si tipul acestuia;

(Afișează / Spune) [Când] / [Cât timp mai este /e până când] trebuie să fac următorul / următoarea ...?

(Afișează / Spune) La ce oră trebuie să fac următorul / următoarea ...?

(Afișează / Spune) Cât timp mai este /e până la următorul ...?

(Afișează / Spune) [Când] / [Cât timp mai este /e până când] trebuie să (mai) fac ...?

(Afișează / Spune) La ce oră trebuie să mai fac ...? /

(Afișează / Spune) Ce activități am astăzi?

(Afișează / Spune) Ce exerciții am (de făcut) astăzi?

(Afișează / Spune) [Când]/ [Cât timp mai este /e până când] am de făcut ...?

- Robotul detectează cand utilizatorul are notificari noi primite (spre exemplu, reminder de a-și lua medicamentele de inimă) care nu au fost confirmate in ultimele 15 min. Atunci acesta il va cauta pe utilizator pentru a-i spune ce are de indeplinit.

Trebuie să-ți iei [la ora ...] / (peste ... minute).

Trebuie să te cântărești [astăzi]/ [la ora ...] / [peste ... minute].

Trebuie să faci exercițiile de azi [la ora ...] / [peste ... minute].

Trebuie să (mai) faci X pași astăzi.

(Mai) ai de făcut X pași astăzi.

3. Asistent de vânzări în magazinul de electronice, raionul laptopuri

Robot: Salut! / Bună! Eu sunt Pepper. / Ma cheamă Pepper. / Numele meu este Pepper. Sunt asistent de vânzări la raionului de laptopuri.

Client: Salut! / Bună! / Bună ziua! / Bună dimineața! / Bună seara!

* robotul este rugat sa ghideze cumpărătorul în achiziționarea unui laptop

Client: [Vreau să cumpăr un laptop. Mă poți ajuta?]
[Aș dori să achiziționez un laptop. Mă ajuți?]
[Mă interesează un laptop. Ce îmi recomanzi?]
[Vreau un laptop! Îmi prezinți câteva modele?]

Robot: Ce fel de laptop vrei să achiziționezi? / Știi ce fel de laptop vrei?

* Cazul cumparatorului care stie ce vrea sa cumpere. Acesta ii spune lui Pepper specificatiile cheie după care poate fi identificat produsul dorit.

Client: Vreau să cumpăr un laptop modelul X.

Vreau un laptop care sa aiba urmatoarele specificatii: ..., procesor (de tipul) ..., (memorie) RAM de ..., (tipul de stocare) ..., capacitate (de stocare) ..., placă video ..., diagonala (display) de ..., format (display) ..., cu / fără touchscreen, finisaj display ..., (greutate de) ... kilograme, autonomie baterie de

* se discuta disponibilitatea produsului

Robot: Am identificat produsul. Acesta este în stoc și este disponibil în showroom / magazin. Ți-l aduc imediat.

Client: [Ok.] [Bine.] [Aștept.]

SAU:

Robot: Am identificat produsul. Acesta este în stoc și este disponibil în showroom / magazin. Te rog să mergi la biroul ... ca să îl iei.

Client: (Bine.) Mulțumesc.

SAU

Robot: Am identificat produsul. Acesta este în stoc, dar nu este disponibil în showroom / magazin. Trebuie să aștepti să devină disponibil.

Client: Da, aștept.

SAU:

Robot: Am identificat produsul. Acesta nu este în stoc. Trebuie să facem comanda la furnizor.

Durează ... zile lucrătoare până ajunge în showroom / magazin. Plasăm comanda?

Client: Da, plasăm comanda. / Ok, plasează comanda. / Bine, comandă-l.

* Cazul cumparatorului care stie doar partial ce vrea sa cumpere. Acesta ii spune lui Pepper doar o parte din specificatiile cheie dupa care poate fi identificat produsul dorit.

Client: Vreau un laptop care să aibă următoarele specificații: ..., diagonala (display) de ..., format (display) ..., cu / fără touchscreen, (greutate) de ... kilograme, autonomie baterie de

Robot: Am identificat mai multe laptopuri care conțin specificațiile tale. Ce tip de procesor vrei să aibă laptopul?

* Clientul poate fi cunoscător sau necunoscător al tipurilor de procesoare.

Client: Vreau să aibă un procesor

Robot: Ok, am ales procesorul pe care îl dorești: ... [spune ce a ales]

Client: Nu știu tipurile de procesoare, dar îl vreau pe cel mai bun. Nu știu tipurile de procesoare, dar îl vreau pe cel mai bun pentru jocuri / programare

Robot: Ok. Am ales un procesor [!Robotul trebuie sa stie care este cel mai bun sau cel mai nou tip de procesor / cel mai bun pentru jocuri / programare]

Client: Ok.

Robot: Ce capacitate vrei să aibă memoria RAM?

*** Clientul poate fi cunoscător sau necunoscător al tipurilor de memorii RAM.

Client: [Vreau ca memoria să aibă peste ... RAM.]

[Vreau un laptop cu o memorie RAM cât mai mare.]

[Vreau ca memoria să fie suficientă pentru a rula programe / jocuri]

Robot: Ok, am ales conform cerințelor tale: ... [spune ce a ales] Ce tip și capacitate de stocare vrei să aibă laptopul?

*** Clientul poate fi cunoscător sau necunoscător al tipurilor de memorii RAM.

Client: Vreau ca tipul de stocare să fie ... și să aibă o capacitate de stocare de ... GB.

Robot: Ok, am ales conform cerințelor tale: ... [spune ce a ales]

Client: Recomandă-mi cel mai bun tip de stocare și cu o capacitate de stocare de cel puțin ... GB.

Robot: Ok, am ales conform cerințelor tale: ... [spune ce a ales]

Client: Nu știu, recomandă-mi tu tipul și capacitatea de stocare pentru un laptop mai bun.

Robot: Ok, voi alege cele mai bune variante.

** se discuta disponibilitatea produsului

Robot: Am identificat produsul. Acesta este în stoc și este disponibil în showroom / magazin. Ți-l aduc imediat.

Client: [Ok.] [Bine.] [Aștept.]

SAU:

Robot: Am identificat produsul. Acesta este în stoc și este disponibil în showroom / magazin. Te rog să mergi la biroul ... ca să îl iei.

Client: (Bine.) Mulțumesc.

SAU

Robot: Am identificat produsul. Acesta este în stoc, dar nu este disponibil în showroom / magazin. Trebuie să aștepti să devină disponibil.

Client: Da, aștept.

SAU:

Robot: Am identificat produsul. Acesta nu este în stoc. Trebuie să facem comanda la furnizor.

Durează ... zile lucrătoare până ajunge în showroom / magazin. Plasăm comanda?

Client: Da, plasăm comanda. / Ok, plasează comanda. / Bine, comandă-l.

* Cazul cumpărătorului care nu știe specificațiile laptopului pe care vrea să îl cumpere.

Client: Nu știu exact ce fel de laptop vreau să cumpăr. Ma poți ajuta să mă hotărâsc?

Robot: Sigur. Vrei un laptop de gaming / business / home ?

Client: Da, vreau un laptop gaming / business / home.

Robot: Avem un laptop gaming / business / home la promoție. Te interesează?

Client: Da, mă interesează. Este în stoc?

Robot: Da. Aștepți să ți-l aduc?

Client: Da.

Robot: Avem un laptop gaming / business / home la promoție. Te interesează?

Client: Care este diferența între / dintre ele?

Robot: Un laptop de gaming este un laptop care ... (urmează o descriere).

Un laptop de business este un laptop care ... (urmează o descriere).

Un laptop home este un laptop care ... (urmează o descriere).

Client: Nu înțeleg ce vrei să spui.

Robot: Voi cere ajutorul colegului meu ...

4. Partener de dialog "promoțional (small-talk)"

Producătorul robotului afirmă că în configurația livrată, Pepper poate susține dialoguri în limba engleză de tipul celor de mai jos. Dialogurile au fost traduse în limba română de membrii echipei.

Cum te cheamă? / Care este numele tău? / Cum ți se spune? / Cum te numești? Ce poreclă ai? /

Care e porecla ta? / Cum ți se mai spune? / Cum te mai strigă lumea?

Prezintă-te. / Spune-mi ceva despre tine. / Spune-ne ceva despre tine. / Spune-mi cine ești.

Ce înălțime ai? / Cât de înalt ești? Cât cântărești? / Ce greutate ai? / Câte kilograme ai? Câți ani ai? / Care este vârsta ta? / Care e vârsta ta?

Când e ziua ta? / Când este ziua ta? / Pe ce dată ești făcut? / Pe ce dată ai fost făcut? / Pe ce dată ai fost creat? / Pe ce dată ești creat? / Pe ce dată te-au făcut? / Pe ce dată te-au creat?

Din ce (material / materiale) ești făcut? / Din ce (material / materiale) ești confecționat?

Pentru ce e bună tableta ta? / La ce e bună tableta ta? / Ce poți să faci cu tableta (pe care o ai)? / La ce servește tableta ta? / La ce folosește tableta?

Ești tare? / Ești grozav? / Te crezi tare? / Te crezi grozav? / Te consideri tare? / Te consideri mișto? / Ești cool? / Crezi că ești cool? Ești inteligent? / Ești deștept?

Pot să am încredere în tine? / Pot să mă încred în tine? / Pot avea încredere în tine? / Ești de încredere? / Se poate avea încredere în tine? Ai sentimente? / Ai emoții? Gândești? / Faci raționamente? / Poți gândi?

Poți simți emoțiile? / Simți emoțiile? / Poți să simți emoțiile? Cum detectezi emoțiile? / Cum recunoști emoțiile?

Poți vorbi? / Poți să vorbești? / Poți să spui ceva? / Poți spune ceva? / Poți să zici ceva? / Poți zice ceva?

Mă înțelegi? / Înțelegi ce zic? / Înțelegi ce spun? / Pricepi ce spun? / Pricepi ce zic? Îți (mai) amintești de mine? / Mă mai știi? / Îți (mai) aduci aminte de mine? / Mă mai ții minte? / Mai știi cine sunt? / Mă recunoști?

Îl știi pe Nao? / Știi cine e Nao? / Îl cunoști pe Nao? Îl știi pe Romeo? / Știi cine e Romeo? / Îl cunoști pe Romeo?

De ce te cheamă Pepper? / De ce te numești Pepper? / De ce ți se spune Pepper? / De ce ai numele Pepper?

Cine a decis să te numească așa? / Cine a decis să te cheme așa? / Cine ți-a dat numele acesta? / Cine ți-a dat numele ăsta? / Cine a hotărât să te cheme așa? / Cine a hotărât să te numească așa?

Cine te-a făcut? / Cine te-a fabricat? / Cine te-a creat?

Unde ai fost făcut? / Unde ai fost fabricat? / Unde ai fost creat? Ai prieteni?

Cum îl cheamă pe prietenul tău? / Cum o cheamă pe prietena ta? / Care este numele prietenului tău? / Care este numele prietenei tale?

Vrei să fim prieteni? / Putem fi prieteni? / Putem să fim prieteni? / Vrei să fii prietenul meu?
Ce (mai) faci? / Cum îți (mai) merge? / Cum o (mai) duci? Poți să-ți iei la revedere? / Poți să
spui la revedere? / Poți să zici la revedere? / Poți spune la revedere? /

Ce poți să faci? / Ce poți face? / Ce știi să faci? Spune-mi tot ce poți să faci. / Spune-mi toate
câte le poți face. / Enumeră(-mi) tot ce poți să faci. / Spune-mi tot ce poți face. / Enumeră(-mi) tot
ce poți face. / Spune-mi ce poți să faci. / Spune-mi ce știi să faci.

Ce am spus? / Ce am zis? / Ai înțeles? Te rog să repeți. / Te rog, repetă. / Repetă, te rog. / Poți
repetă, te rog? / Poți să repeți, te rog?

Câtă baterie mai ai? / Care este nivelul bateriei? / Care e nivelul bateriei? / Cum stai cu bateria?
Te încarci (acum)? / Ești la încărcat (acum)? Ești la/în priză (acum)? / Ești pus la încărcat
(acum)?

Ce adresă (de) IP ai? / Ce IP ai? / Care e IP-ul tău? / Care este IP-ul tău? Care este/e adresa ta IP?
Ești conectat la Internet? / Ai conexiune la Internet? / Ești legat la Internet? / Ești conectat la net?
/ Ai conexiune la net? / Ești legat la net? / Ai Internet? / Ai net? / Îți merge Internetul? / Îți merge
netul?

Cât e ceasul? / Cât este ceasul? / Cât e ora? / Ce oră e? Ce zi e azi? / Ce zi e astăzi? / Ce zi este
azi? / Ce zi este astăzi? / În ce zi suntem?

Ce limbi știi? / Ce limbi vorbești? / Ce limbi cunoști? Ce limbi știi să vorbești? Vorbește
românește. / Vorbește în română. / Spune pe românește. / Zi pe românește. / Știi română? /
Vorbești româna? / Vorbești românește? / Știi românește? / Știi româna? / Vorbești română? /
Știi limba română? / Vorbești limba română? / Cunoști limba română?

Vorbește mai încet. / Mai încet. / Spune mai încet. / Zi mai încet. Vorbește mai tare. / Mai tare. /
Spune mai tare. / Zi mai tare.

Poți să(-ți) miști capul? / Poți să miști din cap? / Poți să dai din cap? / Poți mișca capul? / Poți da
din cap? / Mișcă din cap! / Dă din cap! Poți să întorci capul? / Poți întoarce capul? / Întoarce
capul. Poți să întorci capul... la stânga / la dreapta? / Poți întoarce capul... la stânga / la dreapta? /
Întoarce capul... la stânga / la dreapta.

Poți să te uiți în sus? / Ai putea să te uiți în sus? Poți să te uiți la mine? / Ai putea să te uiți la
mine? Poți să te uiți în altă parte? / Ai putea să te uiți în altă parte? Nu te mai uita la mine. / Nu
mă mai privi. / Încetează să te mai uiți la mine. / Încetează să mă mai privești.

Poți să întinzi brațele? / Poți întinde brațele? Poți să întinzi brațul stâng/drept? / Poți întinde brațul stâng/drept? Ridică brațele. / Sus brațele. Ridică brațul stâng/drept. / Sus brațul stâng/drept. Fă(-ți) mâinile să se miște. / Mișcă(-ți) mâinile. / Dă din mâini. Fă(-ți) mâna stângă/dreaptă să se miște. / Mișcă mâna stângă/dreaptă. / Dă din mâna stângă/dreaptă.

Poți deschide mâinile? / Poți să deschizi mâinile? Poți deschide mâna stângă/dreaptă? / Poți să deschizi mâna stângă/dreaptă? Poți să pui mâinile la spate? Sus mâinile! / Mâinile sus! / Ridică mâinile!

Specificații funcționale ale sistemului și micro-lumilor

Acest livrabil detaliază specificațiile funcționale ale sistemului ROBIN-Dialog, pornind de la primitivele acestui sistem: micro-lumi, roboți, comenzi, acțiuni primitive și scripturi (scenarii) de interacțiune. După ce vom descrie relațiile principale dintre primitivele sistemului prin intermediul unei scheme de interacțiune, livrabilul va continua cu o descriere detaliată a modului în care se pot specifica noi micro-lumi și se pot adăuga roboți în cadrul acestora, roboții fiind definiți împreună cu capacitățile lor. Pentru aceasta, plecând de la acțiunile elementare ale unui robot din cadrul unei micro-lumi și de la primitivele acestuia (acțiuni de bază) de limbaj, se vor detalia modul în care vor fi descrise scripturile (scenariile) de interacțiune cu micro-lumea.

Practic, orice comandă primită în limbaj natural pentru un robot care funcționează într-o micro-lume definită în cadrul sistemului ROBIN-Dialog, se va genera un script de interacțiune ca o combinație de scripturi definite anterior și acțiuni predefinite pentru acel robot. Un script de interacțiune va conține pe lângă setul de acțiuni de bază care vor fi efectuate de către robot și șabloane de replici de dialog cu utilizatorul în cazul în care sunt necesare lămuriri suplimentare pentru a îndeplini comanda respectivă.

De exemplu, presupunând un robot cu abilități de mișcare într-o micro-lume ce modelează obiectele dintr-o casă și care interacționează cu acestea, putem considera următoarea comandă în limbaj natural primită de la utilizator:

1. “Pornește televizorul și schimbă canalul pe Digi24.”

Pentru îndeplinirea acestuia, sistemul ROBIN-Dialog trebuie să informeze robotul că este necesară finalizarea unui script de interacțiune cu mediul și cu utilizatorul. Acest script poate fi modelat în felul următor:

2. Pentru interacțiunea cu mediul, robotul trebuie să execute următoarele acțiuni sau scripturi predefinite:
 1. Găsească telecomanda în casă
 2. Găsească televizorul (dacă nu știe unde e deja)
 3. Să pornească televizorul
 4. Să apese pe P+ sau P- până ajunge la postul Discovery

- Pentru interacțiunea cu utilizatorul, robotul poate să îi ceară acestuia informații suplimentare precum:
 1. Unde e telecomanda?
 2. Unde e televizorul?
 3. Pe ce canal este postul Discovery?

Scenariile de interacțiune vor conține legături între expresii în limbaj natural, exprimate printr-o serie de șabloane, pentru identificarea comenzilor și o secvență de acțiuni care urmează să fie executate de către robot în micro-lumea respectivă. Pentru definirea acestor șabloane și expresii în limbaj natural este necesară definirea unui vocabular specific fiecărei micro-lumi. De exemplu, în micro-lumea fictivă atașată comenzii anterioare, robotul trebuie să știe să interpreteze entitățile “televizor” și “canal”, precum și comenzi de deplasare și de schimbare a unui canal prin apăsare pe butoanele telecomenzii.

De asemenea, tot în acest livrabil vom descrie pe scurt și cum se pot defini câteva scenarii simple, pentru niște micro-lumi specifice care vor fi folosite în cadrul proiectului ROBIN-Dialog. Pe parcursul implementării proiectului, atât micro-lumile, cât și scenariile, pot suferi modificări în funcție de necesitățile utilizatorilor și de soluțiile tehnice care vor fi identificate că pot rezolva eficient problema interacțiunii în aceste micro-lumi.

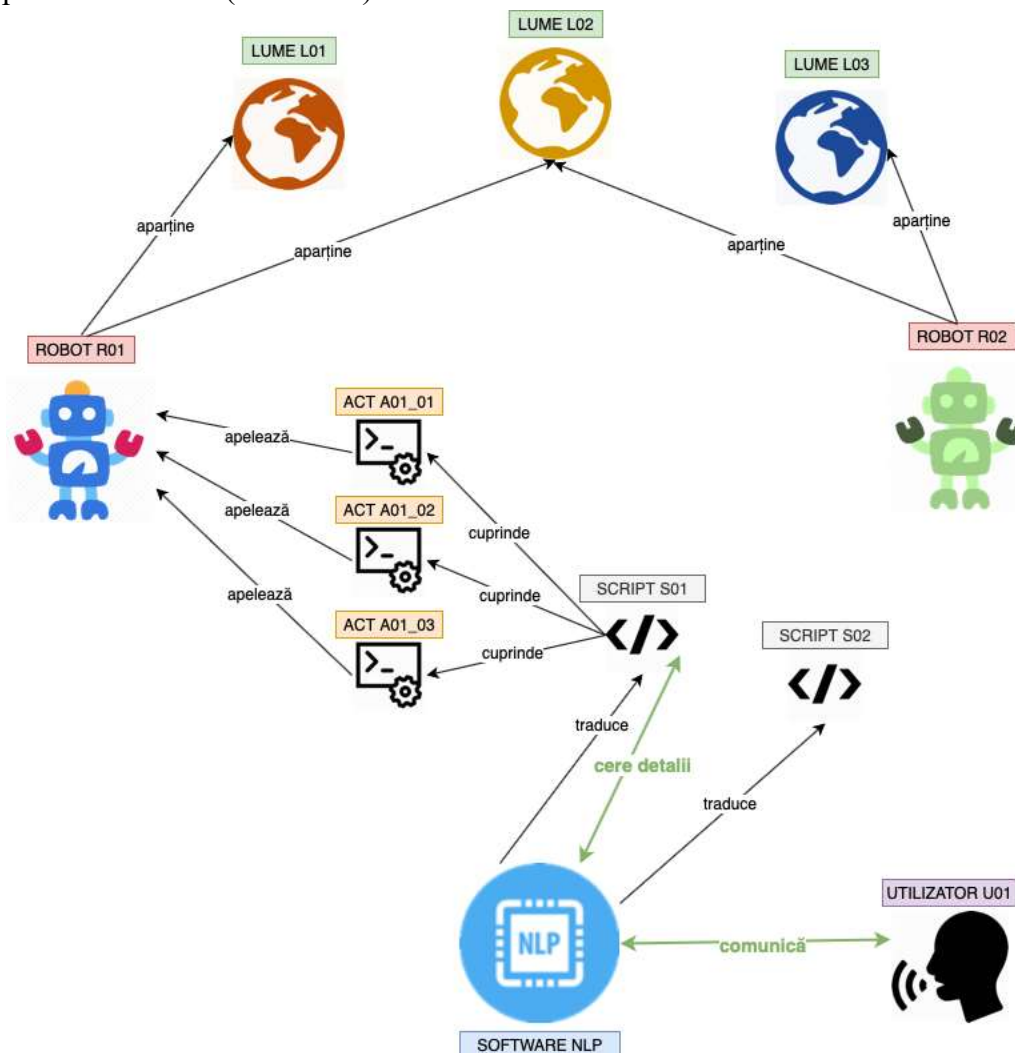
Definirea și folosirea scenariilor pentru micro-lumi este una din cele mai eficiente metodologii de proiectare a componentei de comunicare situațională în limbaj natural (Schank, 1991). Marele avantaj al acestei abordări constă în posibilitățile de control asupra intențiilor partenerului uman, al conținutului cel mai probabil al unei cereri sau comenzi, precum și al solicitărilor inteligente de clarificare în condițiile insuficienței cunoștințelor pentru prelucrarea mesajului transmis (Boroș și Dumitrescu, 2017; Boros et al., 2016). Dialogul situațional promovează metode lexicosemantice, precum spații vectoriale de reprezentare continuă a semnificației cuvintelor și frazelor, pentru care tehnologiile moderne bazate pe rețele neurale (Mikolov et al., 2013) sau alte metode statistice (Pennington et al., 2014) oferă soluții foarte robuste ce pot fi ușor folosite și pentru limba română.

Documentul continuă cu definirea diagramei de interacțiune de bază între entitățile principale din sistemul ROBIN-Dialog. Această diagramă descrie pe scurt fluxurile informaționale care vor permite realizarea unei comenzi în limbaj natural de către un robot în cadrul unei micro-lumi. După aceea, vom continua cu definirea principalelor elemente din cadrul sistemului ROBIN-Dialog, împreună cu câteva exemple și cu modalitățile de definire și de modificare a acestor entități, pornind de la micro-lumi până la scripturi de interacțiune. Vom termina cu câteva concluzii despre modul în care acest livrabil va fi folosit în cadrul proiectului, precum și cum poate fi acesta modificat ulterior.

Schemă de interacțiune a sistemului ROBIN-Dialog

Sistemul ROBIN-Dialog încorporează mulțimea de roboți definiți de utilizator, împreună cu micro-lumile din care fac parte aceștia. Prin intermediul unor comenzi vocale exprimate în limbaj natural de către utilizator, roboții vor realiza acțiuni dedicate specifice fiecărei micro-lumi și capabilităților robotului respectiv.

De la mesajul verbal sau text, care exprimă o comandă, transmis de către individ până la acțiunea specifică efectuată de robot în cadrul mediului, o serie de pași sunt necesari. În primă instanță, mesajul verbal se traduce prin mecanisme avansate de Prelucrare a Limbajului Natural (engleză, „Natural Language Processing”, NLP) în script-uri de interacțiune specifice robotului și mediului. Script-urile cuprind multiple acțiuni ce reprezintă acțiuni atomice ce vor fi efectuate de către un robot pentru îndeplinirea comenzii. De asemenea, după cum se poate vedea și în Figura 2, un script de interacțiune poate cere detalii adționale utilizatorului, exprimate în limbaj natural, pentru realizarea (mai facilă) a comenzii.



Figură 1. Schema de interacțiune a principalelor componente din sistemul ROBIN-Dialog

În final, acțiunile determină roboții să interacționeze cu mediul micro-lumii și să realizeze fapte concrete pornind de la capabilitățile fiecărui robot. Figura **Error! No bookmark name**

given.2 descrie schema de interacțiune dintre micro-lumi, roboți și utilizatori prin intermediul limbajului natural și scripturilor de interacțiune cu utilizatorul și cu mediul.

Definirea comenzilor ROBIN-Dialog

Secțiunea descrie comenzile de pseudo-limbaj ce pot fi folosite în cadrul mediului integrat ROBIN pentru a comunica cu roboții înregistrați în sistem. Interacțiunea cu sistemul se face prin comenzi scrise în limba română. Acestea pot fi extinse eventual și în alte limbi, de exemplu în engleză, dacă este cazul.

Sistemul folosește metode avansate de Prelucrare a Limbajului Natural bazate pe ontologii – pentru ontologii lexicale în limba română, a se vedea Tufiş și Mititelu (2015) – și modele semantice ce folosesc spații vectoriale de reprezentare continuă a semnificației cuvintelor și frazelor și care sunt antrenate pe colecții mari de texte scrise în limba română.

Structura standard a unei comenzi este de forma:

COMANDA ROBOT

în care COMANDA reprezintă o comandă exprimată în limbaj natural cu care se poate comunica cu un robot, iar ROBOT reprezintă unul dintre roboții definiți în sistem și care trebuie să efectueze comanda respectivă. Comenzile text se pot introduce din cadrul liniei de comandă furnizate de programul software ROBIN-Dialog sau folosind o interfață simplă cu utilizatorul. Odată lansat, programul permite utilizatorului să apeleze comenzile în limbaj natural, funcționând ca un program software de tipul *terminal* sau *command prompt* (după numele acestor tip de aplicații din limba engleză).

Dacă comenzile vor fi exprimate prin voce, atunci acestea vor fi analizate întâi de o componentă de recunoaștere a vorbirii care va genera comanda în format text. Pentru limba română, interacțiunea prin voce este încă o mare provocare, cu câteva experimente încurajatoare, dar limitate.

Odată lansată o comandă, dacă aceasta necesită introducerea unor parametri suplimentari, utilizatorul va fi interogat în privința acestora în limbaj natural. Aceste interacțiuni pot fi ori în format text, folosind terminalul sau o interfață grafică simplă sau prin voce, folosind un sistem de sinteză a vorbirii în limba română (Boroș și Dumitrescu, 2015).

Sistemul integrat se constituie dintr-o colecție de roboți și micro-lumi, mai mulți roboți putând fi conectați la mai multe micro-lumi, respectiv o micro-lume putând avea mai mulți roboți – relație cu dublu sens între cele două tipuri de entități. În continuare vor fi definite modalitățile de creare a primitivelor sistemului ROBIN-Dialog: roboți, lumi, acțiuni și scripturi de interacțiune. Componenta de Prelucrare a Limbajului Natural va face legătura între comenzile de la utilizator și scripturi, iar scripturile vor specifica modul de interacțiune între roboți, utilizator și mediu.

Specificarea primitivelor sistemului ROBIN-Dialog

Roboți

Inregistreaza Robot

Această comandă permite înregistrarea unui robot în cadrul sistemului ROBIN. Utilizatorului i se vor cere o serie de parametri obligatorii pentru identificarea robotului:

- Identificator (șir de caractere, obligatoriu) – reprezintă un identificator unic al robotului în cadrul sistemului; utilizatorul are obligația de a seta identificatorul, întrucât acesta va fi folosit pentru alte comenzi aplicate asupra roboților. Identificatorul nu poate conține spații, ci doar litere, cifre și caracterele cratimă ('-' și *underscore* ('_')), iar primul caracter trebuie să fie o literă.
- IP (adresă IP, obligatoriu) – constituie adresa IP (IPv4 sau IPv6) cu care se face comunicarea cu robotul. În general, o adresă IP reprezintă o singură entitate robot, însă configurările avansate pot diferenția între roboți cu aceeași adresă IP în baza altor criterii.
- Nume (șir de caractere, obligatoriu) – reprezintă un nume ce va fi asignat de către utilizator pentru identificarea facilă a robotului. În cadrul sistemului ROBIN pot exista mai mulți roboți cu același nume.
- Locație (șir de caractere, opțional) – reprezintă locația (numărul camerei, al clădirii, al apartamentului etc.) în care se află robotul. Parametrul este util pentru identificarea facilă a robotului pe baza locației sale fizice. În cadrul sistemului ROBIN pot exista mai mulți roboți aflați în aceeași locație.
- Adresă (șir de caractere, opțional) – reprezintă adresa locației în care se află robotul. Parametrul este util pentru identificarea facilă a locației fizice în care se află un robot. În cadrul sistemului ROBIN pot exista mai mulți roboți având aceeași adresă.

Odată completați acești parametri, sistemul va încerca, în primul rând, comunicarea cu robotul prin transmiterea unui mesaj tip *Hello* la adresa IP indicată. Dacă nu se primește niciun răspuns, utilizatorul va fi notificat în acest sens, iar adăugarea robotului va fi anulată.

Dacă s-a primit răspuns de la robot, se vor face verificări suplimentare, precum faptul ca identificatorul asignat robotului să fie unic în sistem, numele să fie valid sau celelalte câmpuri să nu conțină caractere invalide. Dacă vreuna dintre aceste verificări produce mesaj de eroare, utilizatorul va fi notificat corespunzător.

Daca s-a reușit adăugarea robotului în sistem, utilizatorului i se va afișa un mesaj de succes.

Listeaza Roboti

Această comandă permite listarea tuturor roboților existenți în sistem, fiind similară cu o comandă de tipul SELECT din Sistemele de Gestiune a Bazelor de Date pentru entitățile de tip robot. Comanda va afișa toți roboții înregistrați în sistem, împreună cu informațiile cele mai importante: identificator, adresă IP și nume.

Detalii Robot

Această comandă permite afișarea tuturor detaliilor unui robot înregistrat în sistem. Odată lansată, utilizatorul va fi interogat în privința identificatorului unic al robotului pentru care se

dorește afișarea detaliilor. Dacă nu există un robot cu acest identificator, un mesaj de eroare corespunzător va fi afișat. Dacă robotul există, toate detaliile sale vor fi afișate.

Sterge Robot

Comanda permite eliminarea din sistem a unui robot. Odată lansată, utilizatorului i se va cere furnizarea identificatorului unic al robotului. Dacă nu există un robot cu acest identificator, un mesaj de eroare corespunzător va fi afișat. Dacă există, se va cere confirmarea acestei acțiuni, urmând ca robotul să fie eliminat definitiv din sistem, precum și din lumile în care a fost asignat.

Editeaza Robot

Comanda permite editarea datelor unui robot. Odată lansată, utilizatorului i se va cere furnizarea identificatorului unic al robotului. Dacă nu există un robot cu acest identificator, un mesaj de eroare corespunzător va fi afișat. Dacă există, utilizatorului i se vor cere toate informațiile cerute de comanda INREGISTREAZA ROBOT, cu modificarea că aceste câmpuri vor fi pre-completate cu datele deja introduse.

Selecteaza Robot

Prin această comandă este selectat unul din roboții definiți în sistem, permițând apoi o interacțiune cu acesta prin intermediul acțiunilor și scripturilor pentru robotul respectiv. Selectarea unui robot se va face prin furnizarea identificatorului unic sau al unui nume. Dacă există mai mulți roboți în sistem cu același nume se va cere utilizatorului dezambiguizarea explicită pe baza identificatorului. În cazul în care nu există nici un robot cu identificatorul sau numele cerut de către utilizator, se va întoarce un mesaj de eroare specific către acesta pentru a-l atenționa de acest lucru.

După selectarea cu succes a unui robot, acestuia i se poate cere executarea unei acțiuni simple din lista de acțiuni suportate de către acel robot sau a unei secvențe de acțiuni definite în cadrul unui script sau scenariu. De asemenea, se va putea cere listarea tuturor acțiunilor și scripturilor pe care robotul le poate efectua pentru a permite o interacțiune mai facilă cu fiecare robot în parte.

Se va avea în vedere și posibilitatea de a selecta roboții pornind de la anumite informații semantice, precum:

- Selectează robotul care se află la o anumită locație, cel mai aproape de o locație sau este la o distanță specifică de o locație menționată de către utilizator. De exemplu: *selectează robotul aflat la sub 500 metri de locația <X>* sau *găsește robotul cel mai apropiat de către locația <Y>*
- Selectează un robot pornind de la informații semantice despre capacitățile acestuia, de exemplu despre acțiunile pe care le poate efectua. De exemplu: *selectează robotul care stinge lumina în camera <X>*.
- Se pot folosi și selecții combinate, folosind locații și informații semantice despre capacitățile robotului.
- Dacă vor exista mai mulți roboți care îndeplinesc cerințele selecției (semantice) menționate de către utilizator, acesta va fi avertizat și i se va cere alegerea explicită a unuia dintre

aceștia pe baza identificatorului (plus eventual alte informații semantice despre lume și roboții care sunt întorși de către comanda de selecție).

Micro-lumi

Creează Lume

Comanda permite crearea unei lumi noi. Odată lansată, aceasta va cere o serie de parametri suplimentari care identifică lumea, mai concret:

- Identificator (șir de caractere, obligatoriu) – reprezintă un identificator unic al lumii în cadrul sistemului; utilizatorul are obligația de a seta acest identificator, întrucât acesta va fi folosit pentru alte comenzi aplicate asupra lumilor. Identificatorul nu poate conține spații, ci doar litere, cifre și caracterele cratimă și *underscore*, iar primul caracter trebuie să fie o literă.
- Nume (șir de caractere, obligatoriu) – reprezintă un nume ce va fi asignat de către utilizator pentru identificarea facilă a lumii nou create. Numele este unic, în cadrul sistemului ROBIN neputând exista mai multe lumi cu același nume.
- Locație (șir de caractere, opțional) – reprezintă locația (numărul camerei, al clădirii, al apartamentului etc.) care reprezintă lumea. Parametrul este util pentru identificarea facilă a lumii pe baza locației sale fizice. În cadrul sistemului ROBIN pot exista mai multe lumi aflate la aceeași locație sau la o locație apropiată.
- Adresă (șir de caractere, opțional) – reprezintă adresa locației în care se află lumea. Parametrul este util pentru identificarea facilă a locației fizice în care se află o lume. În cadrul sistemului ROBIN pot exista mai multe lumi având aceeași adresă.
- Harta detaliată a locației (fișier binar, opțional) – reprezintă o hartă în format digital specific sistemelor de informații geografice (GIS, engleză „Geographic Information System”), precum formatele SHP, KMZ/KML, GDB sau MDB. Aceste fișiere pot conține informații detaliate 2D sau 3D despre locația selectată pentru lume care va permite roboților o navigare mai precisă, precum și posibilitatea de localizare a fiecărui robot și a altor artefacte importante specifice lumii pe hartă.
- Entități existente în lume (listă entități, opțional) – specifică setul de entități existente în micro-lumea curentă (de ex. într-o clădire vor exista camere specifice, diverse obiecte de mobilier relevante pentru micro-lume etc.)

Listează Lumi

Această comandă permite listarea tuturor lumilor existente în sistem, fiind similară cu o comandă de tipul SELECT din Sistemele de Gestiune a Bazelor de Date. Comanda va afișa toate lumile înregistrate în sistem, împreună cu informațiile cele mai importante: identificator, nume și locație.

Detalii Lume

Această comandă permite afișarea tuturor detaliilor unei lumi înregistrate în sistem. Odată lansată, utilizatorul va fi interogat în privința identificatorului unic al lumii pentru care se dorește

afișarea detaliilor. Dacă nu există o lume cu acest identificator, un mesaj de eroare corespunzător va fi afișat. Dacă lumea există, toate detaliile sale vor fi afișate.

Șterge Lume

Comanda permite eliminarea din sistem a unei lumi. Odată lansată, utilizatorului i se va cere furnizarea identificatorului unic al lumii. Dacă nu există o lume cu acest identificator, un mesaj de eroare corespunzător va fi afișat. Dacă există, se va cere confirmarea acestei acțiuni, urmând ca lumea să fie eliminată definitiv din sistem, precum și legăturile sale cu roboții cu care a fost asignată.

Editează Lume

Comanda permite editarea datelor unei lumi. Odată lansată, utilizatorului i se va cere furnizarea identificatorului unic al lumii. Dacă nu există o lume cu acest identificator, un mesaj de eroare corespunzător va fi afișat. Dacă există, utilizatorului i se vor cere toate informațiile cerute de comanda CREEAZA LUME, cu modificarea că aceste câmpuri vor fi pre-completate cu datele deja introduse.

Selectează Lume

Prin această comandă este selectată una dintre lumile definite în sistem, permițând apoi o interacțiune cu acesta prin intermediul acțiunilor și scripturilor pentru roboții din cadrul lumii respective. Selectarea unei lumi se va face prin furnizarea identificatorului unic sau al unui nume. Dacă există mai multe lumi în sistem cu același nume se va cere utilizatorului dezambiguizarea explicită pe baza identificatorului. În cazul în care nu există nicio lume cu identificatorul sau numele cerut de către utilizator, se va întoarce un mesaj de eroare specific către acesta pentru a-l atenționa de acest lucru.

După selectarea cu succes a unei lumi, se poate cere executarea unor acțiuni simple din lista de acțiuni suportate de către roboți existenți în acea lume sau a unei secvențe de acțiuni definite în cadrul unui script sau scenariu. De asemenea, se va putea cere listarea tuturor roboților, respectiv a acțiunilor și scripturilor pe care roboții din cadrul lumii le poate efectua pentru a permite o interacțiune mai facilă cu fiecare robot în parte din cadrul lumii.

Acțiuni

Creeaza Acțiune

Comanda permite crearea unei acțiuni noi. Odată lansată, aceasta va cere o serie de parametri suplimentari care identifică acțiunea, mai concret:

- Identificator (șir de caractere, obligatoriu) – reprezintă un identificator unic al acțiunii în cadrul sistemului; utilizatorul are obligația de a seta acest identificator, întrucât acesta va fi folosit pentru a coordona interacțiunea dintre roboți și lumi. Identificatorul nu poate conține spații, ci doar litere, cifre și caracterele cratimă și *underscore*, iar primul caracter trebuie să fie literă. Întrucât acțiuni cu nume similar pot fi create pentru diferite tipuri de roboți, includerea în numele identificatorului a entității pentru care se aplică este recomandată pentru diferențierea facilă a acțiunilor.

- Descriere (șir de caractere, opțional) – reprezintă o descriere a acțiunii ce va fi utilă pentru utilizator în a identifica scopul acelei acțiuni. Descrierea nu este unică, în cadrul sistemului ROBIN putând exista mai multe lumi cu aceeași descriere, însă este recomandat să fie cât mai detaliată pentru a nu confuza utilizatorii.
- Robot (șir de caractere, opțional) – reprezintă identificatorul robotului pentru care se creează scriptul. Acest parametru este opțional deoarece se poate seta ulterior prin editarea unei acțiuni.
- Comenzi robot (șir de caractere, opțional) – reprezintă comenzile specifice care trebuie executate de către robot când se aplează prezenta acțiune. În momentul în care va fi lansată o acțiune, aceste comenzi specifice vor fi rulate. Acest parametru este opțional deoarece se poate seta ulterior prin editarea unei acțiuni.
- Parametri (obiect de tip JSON) – reprezintă un obiect ce stochează parametrii necesari pentru o anumită acțiune. Fiecare acțiune poate necesita o listă specifică de parametri, ce vor fi pasați mai departe către script-ul corespunzător acțiunii.

În continuare sunt prezentate o serie de exemple de posibile acțiuni pentru roboții creați în cadrul sistemului ROBIN-Dialog, împreună cu parametrii specifici fiecărei acțiuni:

- GOTO <x>
- PICK <object>
- CLEAN FROM <xstart, ystart> TO <xend, yend>
- TURN LIGHT <X> ON/OFF

Listează Acțiuni

Această comandă permite listarea tuturor acțiunilor existente în sistem, fiind similară cu o comandă de tipul SELECT din Sistemele de Gestiune a Bazelor de Date. Comanda va afișa acțiunile înregistrate în sistem cu informațiile cele mai importante: identificator, comenzi robot și parametri.

Detalii Acțiune

Această comandă permite afișarea tuturor detaliilor unei acțiuni înregistrate în sistem. Odată lansată, utilizatorul va fi interogat în privința identificatorului unic al acțiunii pentru care se dorește afișarea detaliilor. Dacă nu există o acțiune cu acest identificator, un mesaj de eroare corespunzător va fi afișat. Dacă acțiunea există, toate detaliile sale vor fi afișate.

STERGE ACTIUNE

Comanda permite eliminarea din sistem a unei acțiuni. Odată lansată, utilizatorului i se va cere furnizarea identificatorului unic al acțiunii. Dacă nu există o acțiune cu acest identificator, un mesaj de eroare corespunzător va fi afișat. Dacă există, se va cere confirmarea ștergerii, urmând ca acțiunea să fie eliminată definitiv din sistem, precum și istoricul aplicărilor sale în concordanță cu roboții cu care a fost asignată.

EDITEAZA ACTIUNE

Comanda permite editarea datelor unei acțiuni. Odată lansată, utilizatorului i se va cere furnizarea identificatorului unic al acțiunii. Dacă nu există o acțiune cu acest identificator, un

mesaj de eroare corespunzător va fi afișat. Dacă există, utilizatorului i se vor cere toate informațiile cerute de comanda CREEAZA ACTIUNE, cu modificarea că aceste câmpuri vor fi precompletate cu datele deja introduse.

Script-uri de interacțiune

Creează Script

Comanda permite crearea unui script de interacțiune, ce va fi alcătuit dintr-o multitudine de acțiuni și/sau alte script-uri definite anterior. Practic un script poate fi definit recursiv folosind alte scripturi mai simple, dar elementul constitutiv de bază îl reprezintă acțiunile pe care le poate efectua un robot în cadrul unei micro-lumi.

Odată lansată comanda de creare a unui nou script, aceasta va cere o serie de parametri suplimentari care identifică script-ul, mai concret:

- Identificator (șir de caractere, obligatoriu) – reprezintă un identificator unic al script-ului în cadrul sistemului; utilizatorul are obligația de a seta acest identificator, întrucât acesta va fi folosit pentru a coordona interacțiunea dintre roboți și lumi, respectiv pentru a asocia acțiunile ce vor fi cuprinse în cadrul unui script. Identificatorul nu poate conține spații, ci doar litere, cifre și caracterele cratimă și *underscore*, iar primul caracter trebuie să fie literă. Întrucât script-uri cu nume similare pot fi create pentru diferite tipuri de roboți, includerea în numele identificatorului a entității pentru care se aplică este recomandată pentru diferențierea facilă a acțiunilor.
- Descriere (șir de caractere, opțional) – reprezintă o descriere a acțiunii ce va fi utilă pentru utilizator în a identifica scopul acestui script. Descrierea nu este unică, în cadrul sistemului ROBIN putând exista mai multe script-uri cu aceeași descriere, însă este recomandat să fie cât mai detaliată pentru a nu confuza utilizatorii.
- Robot (șir de caractere, opțional) – reprezintă identificatorul robotului pentru care se creează scriptul. Acest parametru este opțional deoarece se poate seta ulterior prin editarea unei acțiuni.
- Parametri (obiect de tip JSON) – reprezintă un obiect ce stochează parametrii necesari pentru script. Fiecare script poate necesita o listă specifică de parametrii, ce pot fi pași mai departe către acțiunile din cadrul scriptului.
- Precondiții (șir de caractere, opțional) – reprezintă condițiile pe care trebuie să le satisfacă lumea pentru a putea executa scriptul de interacțiune curentă
- Postcondiții (șir de caractere, opțional) – reprezintă condițiile care vor fi îndeplinite în cadrul micro-lumii după efectuarea cu succes a scriptului curent
- Listă acțiuni (șir de caractere, opțional) – conține acțiunile specifice robotului care trebuie îndeplinite pentru finalizarea cu succes a scriptului de interacțiune. Aceste acțiuni pot conține inclusiv acțiuni de interacțiune cu utilizatorul pentru a cere detalii suplimentare și vor fi definite într-un limbaj specific.

Un exemplu de script simplu este prezentat în continuare pentru un robot care are definite acțiunile GOTO, GRAB, FILL:

Script pentru a aduce pahar de apa de la bucătărie:

1. <GOTO “bucatarie”>
2. <GRAB “pahar”>
3. <FILL “pahar”, “apa”>
4. <GOTO “my-position”>

Listează Scripturi

Această comandă permite listarea tuturor script-urilor existente în sistem, fiind similară cu o comandă de tipul SELECT din Sistemele de Gestiune a Bazelor de Date. Comanda va afișa script-urile înregistrate în sistem cu informațiile cele mai importante: identificator, descriere, robot, parametri.

Detalii Script

Această comandă permite afișarea tuturor detaliilor unui script înregistrat în sistem. Odată lansată, utilizatorul va fi interogat în privința identificatorului unic al scriptului pentru care se dorește afișarea detaliilor. Dacă nu există un script cu acest identificator, un mesaj de eroare corespunzător va fi afișat. Dacă scriptul există, toate detaliile sale vor fi afișate.

Șterge Script

Comanda permite eliminarea din sistem a unui script. Odată lansată, utilizatorului i se va cere furnizarea identificatorului unic al scriptului. Dacă nu există un script cu acest identificator, un mesaj de eroare corespunzător va fi afișat. Dacă există, se va cere confirmarea ștergerii, urmând ca scriptul să fie eliminat definitiv din sistem, precum și istoricul aplicărilor sale în concordanță cu roboții cu care a fost asigurat.

Editează Script

Comanda permite editarea datelor unui script. Odată lansată, utilizatorului i se va cere furnizarea identificatorului unic al scriptului. Dacă nu există un script cu acest identificator, un mesaj de eroare corespunzător va fi afișat. Dacă există, utilizatorului i se vor cere toate informațiile cerute de comanda CREEAZA SCRIPT, cu modificarea că aceste câmpuri vor fi pre-completate cu datele deja introduse.

Alegerea instrumentelor pentru dezvoltare

Am investigat performanțele oferite de mai multe instrumente existente pentru antrenarea și testarea unui sistem de recunoaștere automată a vorbirii. În literatura de specialitate, cele mai folosite instrumente sunt:

- CMUSphinx - conține mai multe unelte open-source care pot fi folosite pentru dezvoltarea aplicațiilor în domeniul recunoașterii vocale. Acesta este dezvoltat în limbajul C și implementează o detecție continuă, independentă de vorbitor, folosind modele Markov cu stări ascunse pentru modelarea acustică și modele statistice de tip n-gram pentru modelarea lingvistică. Acesta este folosit în principal pentru sisteme cu resurse limitate, existând și o variantă destinată sistemelor embedded, numită Pocketsphinx.

- Microsoft Azure Speech to Text API - un API dezvoltat de către Microsoft. Folosirea sistemului este contra-cost: se plătește la nivel de oră. Sistemul nu are suport pentru limba română, dar permite crearea de modele acustice / lingvistice proprii. Este în principal folosit de către companii.
- Kaldi - similar cu CMUSphinx, conține unelte open-source care pot fi folosite pentru dezvoltarea aplicațiilor în domeniul recunoașterii vocale. Este dezvoltat în limbajul C++ și implementează inclusiv rețele neurale profunde. Poate fi antrenat fie dependent, fie independent de vorbitor, și poate fi folosit atât online (datele sunt trimise și prelucrate pe parcursul înregistrării) cât și offline (datele se trimit și prelucrează doar la finalul înregistrării).

Dintre acestea trei, a fost ales Kaldi întrucât a stat la baza proiectelor cu cele mai bune rezultate raportate în literatura de specialitate. Pentru antrenarea unui model cu ajutorul Kaldi sunt necesare mai multe etape de preprocesare. Pentru antrenarea propriu-zisă sunt necesare următoarele fișiere:

- Fișiere audio, precum și textul aferent lor
- Modelul de limbă
- Dicționarul fonetic (cuvintele ce pot fi decodate, urmate de transcrierea lor fonetică)
- Descrierea fonemelor

Pentru crearea modelului de limbă, utilitarul SRILM a fost ales. Acesta primește drept intrare o serie de propoziții, și returnează un model lingvistic bazat pe distribuția statistică a cuvintelor și grupurilor de cuvinte.

O parte din timp a fost alocată pentru înțelegerea și familiarizarea cu Kaldi. În vederea familiarizării cu Kaldi, au fost rulate exemplele oferite pe o bază de date standard, numită TELDIUM. Baza conține înregistrări audio în limba engleză a conferințelor TED (un total de 120 ore), transcrierile lor aferente, dicționarul fonetic (cu aproximativ 160.000 cuvinte) și modelul de limbă.

Pregătirea pentru antrenarea unui model acustic specific limbii române

În vederea antrenării unui model acustic cu ajutorul utilitarului Kaldi, a fost necesară crearea principalelor fișiere menționate anterior. Corpusurile strânse în cadrul centrului au fost prelucrate pentru a satisface cerințele utilitarului Kaldi. Astfel, următoarele operații au avut loc:

- Agregarea tuturor fișierelor audio într-o singură bază de date omogenă
- Crearea unui fișier conținând transcrierea audio pentru fiecare fișier audio din baza de date
- Transcrierea fonetică a fiecărui cuvânt ce apare în fișierele audio
- Crearea unui model statistic de limbă folosind modelele tri-gram și propozițiile fișierelor audio

Etapa de agregare a fișierelor audio a presupus trecerea semnalului audio printr-un filtru de detecție a vorbirii (VAD - voice activity detection) pentru a elimina zonele de liniște de la începutul / finalul semnalului. De asemenea, a avut loc și o validare a fișierelor: au fost păstrate doar fișierele mono-canal și cu frecvență de eșantionare 16Kz (caracteristici necesare pentru antrenarea cu Kaldi).

Etapa de transcriere fonetică a presupus analizarea și tratarea excepțiilor (onomatopeea tkz; apariția unor accente din afara limbii române: Führer, etc). În unele cazuri, s-a ales rescrierea textului, iar în altele eliminarea fișierului audio.

La final, s-au obținut aproximativ 46.000 de cuvinte în dicționarul fonetic necesar antrenării. Durata totală a fișierelor audio este de aproximativ 105 ore.

În cadrul acestui livrabil au fost prezentate elementele de bază ale sistemului ROBIN-Dialog, primitivele acestui sistem, precum și modul în care acestea sunt conectate între ele. În următoarele rapoarte tehnice vor fi definite micro-lumile, scenariile de interacțiune specifice fiecărui robot, precum și vocabularul și limbajul de definiție al acestor scenarii. Trebuie avut în vedere că unele elemente din cadrul acestor specificații funcționale vor putea fi modificate în funcție de problemele tehnice întâmpinate și de către evoluția scenariilor în discuțiile cu utilizatorii finali.

Referințe bibliografice

- Verginica Barbu Mititelu, Dan Tufiș, Elena Irimia (2018). The Reference Corpus of the Contemporary Romanian Language (CoRoLa), Miyazaki, Japan, LREC 2018, pp. 1178-1185, ISBN 979-10-95546-00-9
- Tiberiu Boros, Ștefan Daniel Dumitrescu (2015). Robust deep-learning models for text-to-speech synthesis support on embedded devices. In *Proceedings of the 7th International Conference on Management of computational and collective intelligence in Digital EcoSystems* (pp. 98-102). ACM.
- Tiberiu Boros, Ștefan Daniel Dumitrescu (2017). A “small-data”-driven approach to dialogue systems for natural language human computer interaction. In *Speech Technology and Human-Computer Dialogue (SpeD), 2017 International Conference on* (pp. 1-6). IEEE.
- Tiberiu Boros, Ștefan Daniel Dumitrescu, Horia Cucu (2016). Voice Controlled Home Automation System. In *International Conference on Linguistic Resources and Technologies for the Romanian Language (CONSILR)* (Maria Mitrofan, Daniela Gifu, Dan Tufiș, Dan Cristea). Editura Universitatii "Alexandru Ioan Cuza" din Iasi, vol. 12, Malini, pp. 101-109.
- Qianlong Liu , Zhongyu Wei, Baolin Peng , Xiangying Dai , Huaixiao Tou , Ting Chen , Xuanjing Huang , Kam-fai Wong. (2018). *Task-oriented Dialogue System for Automatic Diagnosis*. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers), ACL 2018, July 2018, Melbourne, Australia, pp. 201—207.
- Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, Jeffrey Dean. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems* (pp. 3111-3119).
- Vasile Păiș, Dan Tufiș. (2018). Computing distributed representations of words using the CoRoLa corpus. In Proceedings of the Romanian Academy, series A, pp. 403-410, 2018

- Sonia Pipa, Tiberiu Boroş. (2016). A Recurrent Neural Networks Approach for Keyword Spotting Applied on Romanian Language. In Proceedings of the 12th International Conference "Linguistic Resources and Tools for Processing of the Romanian Language", 27-29 October, 2016, Romania. „Al. I. Cuza” University Publishing House, pp. 111-119, ISSN 1843-911X.
- Jeffrey Pennington, Richard Socher, Christopher D. Manning. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).
- Vasile Rus, Sidney D’Mello, Xiangen Hu, Arthur C. Graesser. (2013) *Recent Advances in Conversational Intelligent Tutoring Systems*. AI MAGAZINE, Fall 2013, pp. 42—54. Online la <https://www.aaai.org/ojs/index.php/aimagazine/article/view/2485>
- Roger Schank. (1991). Tell me a story: A new look at real and artificial intelligence. *New York*.
- Dan Tufiş, Verginica Barbu Mititelu. (2014). The Lexical Ontology for Romanian. In Nuria Gala, Reinhard Rapp, Gemma Bel-Enguix (eds) *Recent Advances in Language Production, Cognition and the Lexicon*, Springer, vol. 14, 2014, pp. 491-504
- Zhao Yan, Nan Duan, Peng Chen, Ming Zhou, Jianshe Zhou, Zhoujun Li. (2017). *Building Task-Oriented Dialogue Systems for Online Shopping*. In Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17).