

# SPARC

## Servicii Performante de Asistență a Clienților prin Platforme Robotice

cod: PN-III-P2-2.1-BG-2016-0425

Grant in cadrul Programului Național PNIII - Programul 2 - Creșterea competitivității economiei românești prin cercetare, dezvoltare și inovare, Transfer de cunoaștere la agentul economic „Bridge Grant”

### Etapa 3

## Definitivare module funcționale, testare, evaluare și transfer către agentul economic

### Raport științific și tehnic

#### Obiectiv proiect

**Principalul obiectiv al proiectului** este proiectarea și implementarea unei platforme pentru realizarea integrată a planificării sarcinilor roboților de interes în proiect, comanda execuției planurilor definite și monitorizarea acțiunilor acestora. Platforma trebuie să permită personalizarea facilă a programului robotului, îndeosebi pentru scenarii de asistență robotică, dar și crearea unui comportament adaptiv la anumite condiții neprevăzute.

Agentul economic care face obiectul actualei propuneri de proiect, Centrul IT pentru Știință și Tehnologie SRL (CITST) este un IMM orientat spre cercetare și dezvoltare de produse inovatoare. Tehnologia disponibilă la CITST pe care se bazează proiectul este robotul TIAGO. CITST vrea să exploateze mai bine tehnologia robotică disponibilă în companie oferind-o ca serviciu (închiriere, leasing) pentru potențialii clienți. Scenariile de utilizare avute în vedere se referă la promovare de produse și/sau asistență în locuri publice: centre comerciale, muzee, bănci, expoziții. CITST are în vedere și achiziționarea unui robot Pepper în același scop. Tehnologia disponibilă la UPB este reprezentată de roboții Baxter de la Rethink Robotics și Pepper de la Aldebaran Robotics.

Colectivul proiectului este format din membrii grupului de cercetare AIMAS de la Facultatea de Automatică și Calculatoare din UPB și de studenți din cadrul programului de masterat Artificial Intelligence, tot de la Facultatea de Automatică și Calculatoare.

# Cuprins

<b>1. Introducere</b>	3
<b>2. Scenarii de validare</b>	4
<b>3. Îmbunătățiri hardware ale robotului Pepper</b>	5
<b>4. Arhitectura și evaluarea modulelor comportamentale</b>	6
4.1. Modulul de navigare și mapare	7
4.2. Modulul de recunoaștere și localizare a persoanelor	10
4.3. Integrare cu sisteme de context externe	13
4.4. Modulul de Prezentare și Amuzament	15
<b>5. Arhitectura platformei de comportamente și a ciclului de viață al robotului</b>	16
<b>6. Interacțiune vocală în limba română</b>	18
<b>7. Interfața de declanșare a modulelor comportamentale</b>	19
<b>8. Stagii masteranzi și doctoranzi</b>	20
<b>9. Concluzii</b>	21
Bibliografie	23

# 1. Introducere

În etapa anterioară a proiectului au fost prezentate eforturile echipei de a crea un set vast de module de comportament comportamente (eng. behavior models) care să permită definirea funcționalității robotului în termenii paradigmei orientate spre obiectiv (eng. goal driven programming).

□

Cea de-a treia și ultima etapă a proiectului SPARC a avut ca obiectiv principal îmbunătățirea și definitivarea modulelor menționate anterior, integrarea lor prin intermediul unui modul ce definește ciclul de viață al robotului, precum și definirea și implementarea unor scenarii de validare a modulelor dezvoltate.

□

În rubrica de analiză din etapa anterioară, se evidențiază faptul ca, pentru tipul de clienți și obiective urmărite de partenerul CITST, robotul Pepper este mai indicat spre utilizare și exploatare decât Tiago. Principalul avantaj al robotului Pepper este aspectul său mult mai plăcut, precum și existența multiplelor mijloace prin care utilizatorul uman poate interacționa cu robotul.

Cu toate acestea, în urma interacțiunilor ulterioare cu robotul Pepper, echipa de proiect a ajuns la concluzia (confirmată de altfel de experți externi laboratorului care au lucrat cu Pepper), ca hardware-ul (îndeosebi senzorii laser) ai robotului sunt insuficienți pentru a putea atinge nivelul dorit de eficacitate și utilitate a modulelor dezvoltate (cu precădere cele de navigare).

Astfel, în etapa a treia, echipa de proiect a investit efort suplimentar pentru a augmenta capacitățile hardware ale robotului Pepper.

Raportul de față se organizează astfel. În Secțiunea 2 sunt prezentate cele două scenarii care stau la baza validării modulelor comportamentale dezvoltate: cel de asistență robotică pentru persoane în vârstă și cel de prezentare a laboratorului (ghidare pe hol, căutare de persoane, descriere a membrilor laboratorului etc).

În Secțiunea 3 sunt prezentate augmentările hardware aduse robotului Pepper, de care a fost nevoie pentru realizarea pe deplin a capacităților de navigare pe distanțe mari și de ocolire a obstacolelor.

Secțiunea 4 oferă detalii despre modulele comportamentale dezvoltate și despre validarea lor în cele două scenarii menționate.

În secțiunea 5 este prezentat modul în care se realizează combinarea modulelor comportamentale dezvoltate pentru a rezolva obiective complexe (e.g. caută o persoană din laborator și informează-o asupra unei notificări).

Secțiunea 6 intră în detalii asupra modului în care comenzile care declanșează comportamente de bază sau complexe pot fi date în limba română, iar în Secțiunea 7 este descris modul prin care aceleași comportamente pot fi configurate și declanșate cu ajutorul unei simple interfețe grafice.

Secțiunea 8 prezintă concluziile și încheie raportul.

## 2. Scenarii de validare

1.

În cadrul proiectului SPARC au fost definite doua scenarii principale de validare, care au fost agreate de către partenerul CITST. Cele două scenarii au avut menirea de a exemplifica și testa capabilitățile dezvoltate în două ipostaze diferite: prima, cea de asistență robotică personală, pentru o categorie aparte de persoane (cele în vârstă), iar cea de-a doua implicând interacțiunea cu și detectarea de multiple persoane, orientare și navigare în spațiu.

**Scenariul de asistență pentru persoanele în vârstă** a inclus și anumite dezvoltări realizate în cadrul unui alt proiect (CAMI<sup>1</sup>).

Scenariul prevede următoarele tipuri de episoade de interacțiune în care robotul se face util:

Notificări pierdute: persoana în vârstă primește din partea sistemului CAMI notificări care îi aduc aminte să-și ia medicamentele, să execute o măsurătoare de sănătate (e.g. luarea tensiunii și a greutății dimineața devreme, înainte de mic-dejun).

Aceste notificări sunt transmise persoanei prin intermediul telefonului mobil. Totuși, se întâmplă frecvent, ca persoana în vârstă să nu aibă telefonul la îndemână, fiind în casă, nici să aibă obiceiul de a-l verifica periodic.

În acest caz, Pepper, robotul de companie, devine proactiv, observând că persoana în cauză a ignorat notificările mai bine de 15 min. Robotul va căuta persoana în vârstă, navigând prin apartament, începând cu ultima poziție cunoscută a persoanei respective. În momentul în care Pepper găsește persoana, o va anunța cu privire la notificările pierdute. Persoana are posibilitatea de a le vedea pe tableta robotului și de a le asculta prin vorbirea robotului. De asemenea, persoana poate interacționa cu tableta pentru a parcurge toate notificările, le poate confirma prin apăsarea unui buton sau prin comandă vocală.

Verificarea stării de sănătate sau informații generale: robotul Pepper este în permanență capabil să răspundă la întrebările persoanei în vârstă. Aceasta poate cere robotului să îi prezinte, pe tabletă și verbal, starea sa curentă de sănătate (ultimele măsurători de tensiune, puls sau greutate, trendul ultimelor zile, numărul de pași efectuați, sau orele dormite).

De asemenea, persoana în vârstă poate ruga robotul să-i prezinte informații de interes cotidian precum următoarea activitate planificată în calendarul personal sau starea vremii.

În raportul de față se va vedea că aceste episoade de interacțiune utilizează modulele comportamentale care permit detecția persoanelor. navigarea, accesarea notificărilor precum și cele de acces la informații de interes general sau pentru amuzament.

---

<sup>1</sup> Proiectul CAMI: <http://camiproject.eu>

**Scenariul de ghid și prezentator al laboratorului de cercetare** este similar celui din propunerea inițială de proiect (ghid la expoziție sau muzeu), având însă avantajul unui mediu mai complex (un laborator și un hol întreg al unei clădiri de cercetare din campusul UPB) și a posibilității de testare continue.

Episoadele de interacțiune presupun următoarele:

- Construirea și memorarea hărții de obstacole fixe (pereti, mese) în laborator și pe holul etajului.
- Identificarea persoanelor și recunoașterea acestora după față, precum și plasarea lor în mod dinamic pe hartă (atâta timp cât sunt în raza vizuală a robotului)
- Navigarea în harta creată, ocolind obstacolele dinamice
- Prezentarea, la cerere, a detaliilor despre membrii laboratorului și despre cercetarea desfășurată în laborator

Cu elementele de mai sus pot fi puse în practică scenarii de interacțiune precum:

- Robotul Pepper este rugat să ghideze un utilizator până la un anumit laborator de pe hol
- Robotul Pepper este rugat să caute un utilizator. Va face asta pornind de la ultima poziție în care acesta a fost văzut, încercând apoi în laboratoarele cunoscute, iar dacă nu este găsit, robotul se întoarce la poziția lui desemnată drept "acasă".
- Robotul este rugat să vorbească despre unul din membrii laboratorului.
- Robotul este rugat să prezinte activitatea de cercetare din laborator.

În esență, episoadele de interacțiune menționate validează setul de comportamente necesare unei activități de ghid / asistență la expoziții a robotului, având în cazul de față specificul unui laborator de cercetare.

### **3. Îmbunătățiri hardware ale robotului Pepper**

Datorită limitărilor identificate în timpul lucrului cu robotul umanoid Pepper, atât software cât și hardware, modulul de navigare și mapare a fost îmbunătățit prin adăugarea unui echipament LIDAR extern (fig.1). Acest lucru a dus la extinderea capacității de mapare de la 1.5 m folosind echipamentele de pe robot la 10 m. De asemenea frecvența datelor a crescut considerabil până la 10 Hz. Echipamentul LIDAR este așezat în partea din față a robotului, așa cum se poate observa în imaginile de mai jos:

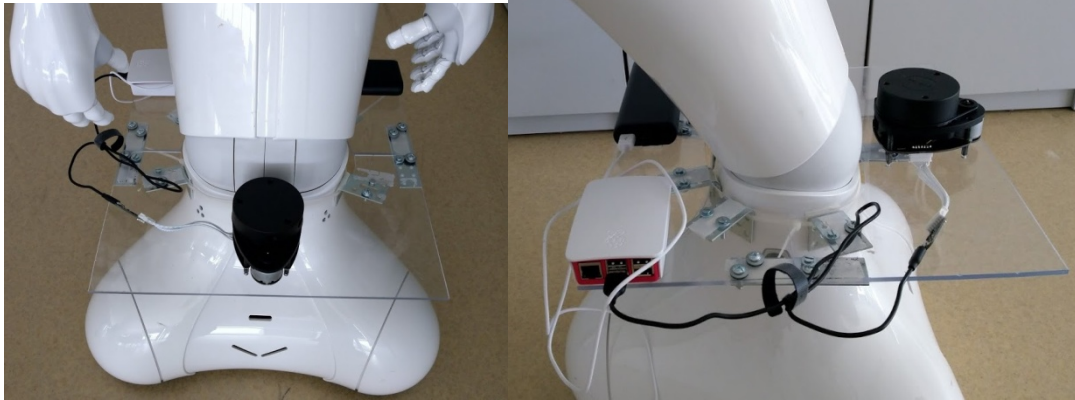


Fig. 1. Platformă externă pentru navigare.

Senzorul este conectat la un Raspberry Pi 3 ce rulează un nod ROS și alimentat la o baterie externă cu durată extinsă de viață. Nodul ROS este folosit pentru achiziția datelor. Acesta publică pe un topic ROS datele sub forma unui mesaj de tip `laser_scan2` ce este preluat de modulul de mapare și navigare.

Raspberry Pi 3 este de asemenea folosit pentru achiziția audio de la un microfon extern. Întrucât software-ul NaoQI nu permite folosirea unor alte module de recunoaștere vocală, iar cel existent nu poate fi folosit în limba română, a fost adăugată și o placă audio ce este folosită pentru a putea adăuga un microfon extern. Fluxul audio este preluat și folosit mai departe de modulul de recunoaștere vocală.

#### 4. Arhitectura și evaluarea modulelor comportamentale

Arhitectura sistemului este modulară și adaptabilă pe orice alt robot, folosind framework-ul ROS pentru a asigura compatibilitatea cu orice robot capabil să ruleze un nod ROS.

---

<sup>2</sup> [http://docs.ros.org/melodic/api/sensor\\_msgs/html/msg/LaserScan.html](http://docs.ros.org/melodic/api/sensor_msgs/html/msg/LaserScan.html)

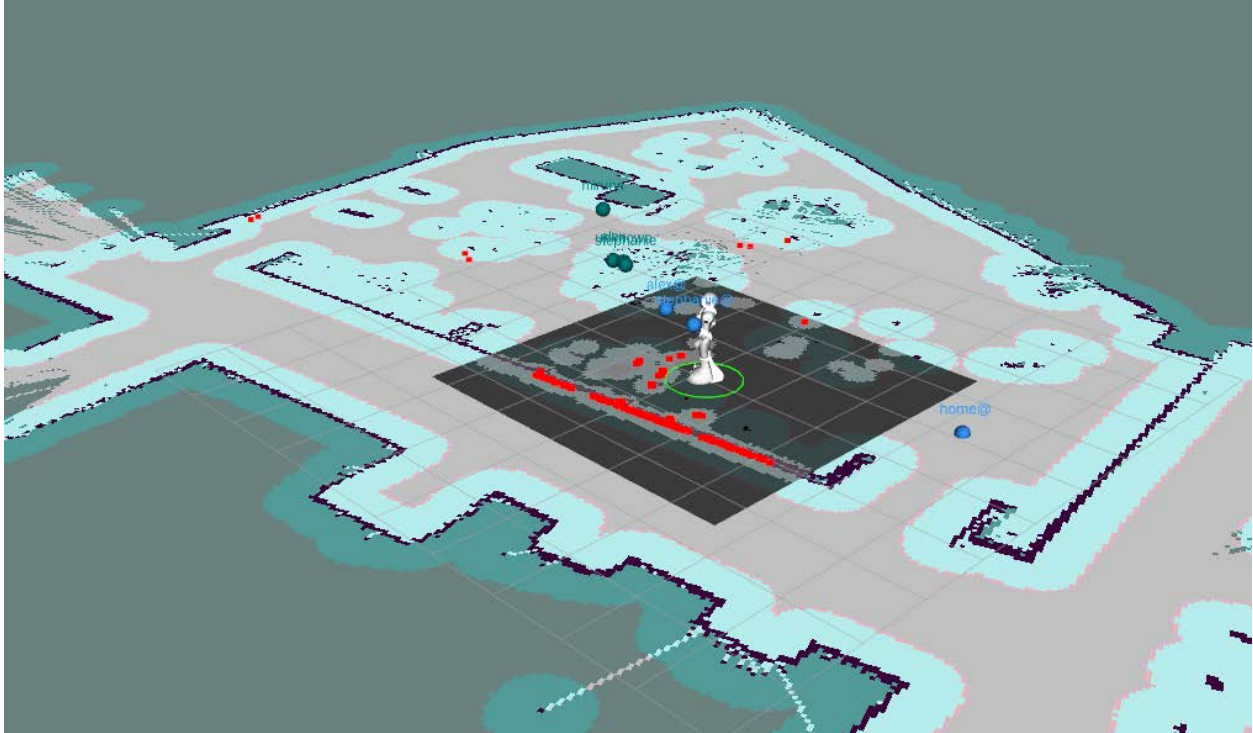


Fig. 2. Integrarea între modulele de navigare și vedere computațională.

#### 4.1. Modulul de navigare și mapare

În vederea mării mediului înconjurător, s-a folosit pachetul ROS `hector_slam`<sup>3</sup> ce implementează un filtru Kalman extins pentru localizare și mapare. Abordarea probabilistică a algoritmului este potrivită pentru Pepper în special întrucât eroare de odometrie a robotului este de aproximativ 9 grade după o rotație completă a robotului de 360 de grade așa cum este prezentat în fig.2 și fig 3.

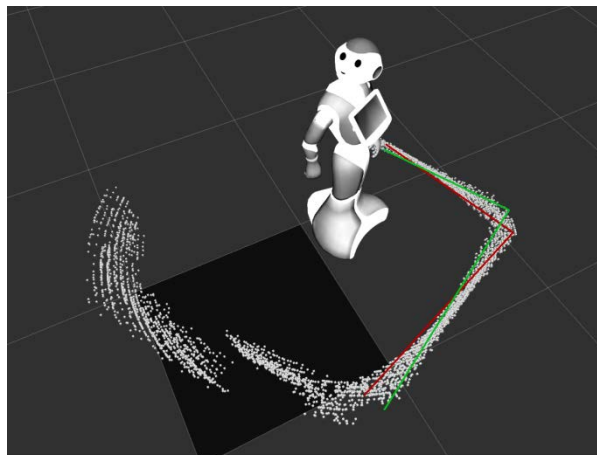


Fig. 3. Eroarea de odometrie a robotului Pepper.

<sup>3</sup> [http://wiki.ros.org/hector\\_slam](http://wiki.ros.org/hector_slam)

După ce se realizează maparea mediului înconjurător, se salvează harta în format PNG împreună cu meta datele despre rezoluție. Harta este un discretă, fiecare celulă a hărții (sau pixel a imaginii) cuantificând probabilitatea ca o porțiune din lumea reală să fie ocupată sau nu. Fiecare scanare cu LIDAR-ul poate mări sau reduce această probabilitate. Etapa de mapare este foarte importantă în procesul de navigare, fiind baza pentru întreg sistemul, permițând ca robotul să navigheze la o poziție predefinită pe hartă. Un exemplu de hartă se poate vedea în fig. 4.



Fig. 4. Reprezentarea mediului folosind LIDAR.

Calitatea hărții depinde puternic de viteza de deplasare a robotului și de calitatea senzorului LIDAR. Spre exemplu folosind senzorii cu care robotul Pepper este echipat rezultatele nu au fost încurajatoare, întrucât este foarte important pentru algoritm găsirea unor puncte cheie (numite landmarks) pentru a putea localiza robotul. Întrucât distanța de măsurare a senzorilor cu care este echipat robotul este de 1.5 m, acest lucru este foarte dificil fiind mult mai greu de identificat punctele cheie. Un exemplu de hartă realizată cu senzorii robotului se poate observa în fig.5.

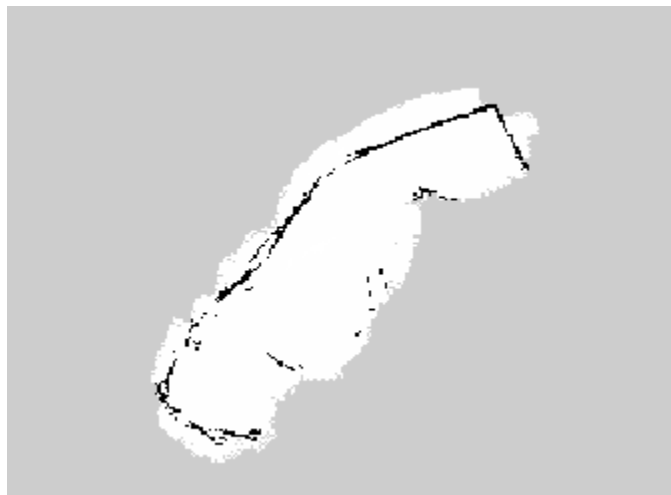


Fig. 5. Reprezentarea mediului folosind senzorii robotului Pepper.



Așa cum se poate observa, folosind un LIDAR extern s-a îmbunătățit cu mult capacitatea de a mapa linii drepte, precum și de a păstra distanțele cu o acuratețe mare.

Pentru a face robotul să navigheze folosind harta mapată anterior, este necesar ca acesta să poată să își găsească punctul de plecare inițial și să își actualizeze poziția la fiecare pas. Pentru acest lucru s-a folosit modulul amcl<sup>4</sup>. Modulul face o căutare încercând să potrivească punctele laser cu porțiunea din hartă. Întrucât spațiul de căutare este unul foarte mare pentru o hartă ce mapează spații mari, este necesar ca algoritmul să primească o poziție inițială de la care să înceapă căutarea.

După localizarea robotului acesta poate să înceapă etapa de navigare. Pentru a putea realiza acest lucru și pentru a reduce limitările cauzate de maparea într-un singur plan, harta a fost transformată. Fiecărei celulă i-a fost atribuită un cost. Costul este aproape de infinit pentru celulele ocupate. Toate celulele din jurul celulelor ocupate au un cost ridicat, liniar cu distanța până la o celulă ocupată. Raza este un parametru ce poate fi modificat și este setată la 30 cm pentru a ne asigura că robotul păstrează o distanță destul de mare până la cel mai apropiat obstacol. Rezultatul hărții modificate se poate observa în fig. 6.

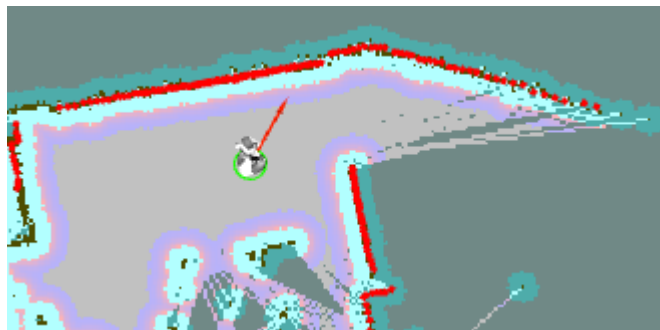


Fig. 6. Reprezentarea mediului după integrarea cu algoritmul de navigare.

Cu roșu se pot observa punctele de la LIDAR ce sunt folosite pentru localizare. De la negru, albastru deschis, mov și apoi gri, costul de a naviga prin acea celulă scade. Algoritmul A\* este apoi folosit pentru a găsi drumul cel mai scurt. Din felul în care este realizat algoritmul, robotul va încerca mereu să găsească drumul cu costul cel mai mic, ceea ce îl va îndruma mereu prin celulele gri. Dacă costul trece de un anumit prag, robotul va renunța să găsească o cale, lucru ce va face imposibilă atingerea obstacolelor statice. În momentul primirii un obiectiv de navigare, Pepper va planifica calea cea mai scurtă până la acesta.

Pentru a putea face robotul să navigheze în medii dinamice, A\* este din nou rulat pe o porțiune din jurul robotului. În momentul în care o persoană sau un alt obstacol dinamic trece prin fața robotului, acesta va replanifica într-o rază de 3-4 m în jurul acestuia pentru a încerca să îl ocolească. În figura de mai jos putem două rulări ale algoritmului. Cu linie galbenă se poate observa planul global, realizat pe harta statică pentru a ajunge la destinație. Cu gri închis în jurul robotului se poate observa aria în care se încearcă ocolirea obstacolelor dinamice.

<sup>4</sup> <http://wiki.ros.org/amcl>

Acestea apar cu culori accentuate în jurul robotului. Cu un cerc verde identificăm aria estimată a robotului. Aceasta este mai mare decât dimensiunea reală a acestuia pentru a asigura distanță potrivită la trecerea prin spații strâmte cum este ușa (fig.7)

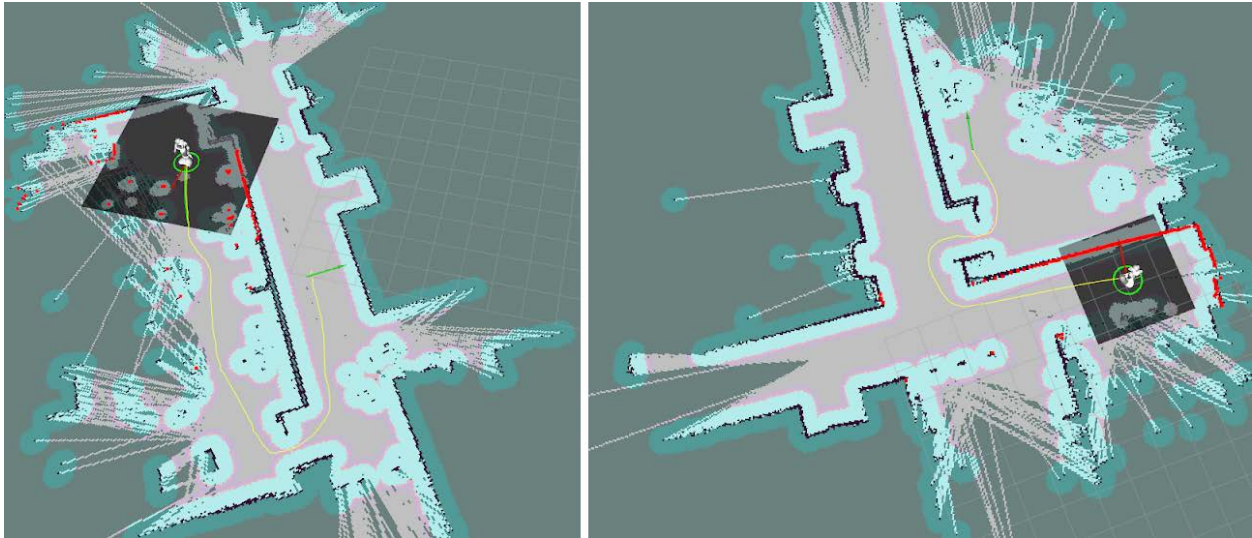


Fig. 7. Reprezentare a algoritmului de navigare în mediu.

#### 4.2. Modulul de recunoaștere și localizare a persoanelor

Detecția, recunoașterea și localizarea persoanelor este făcută în timp real, cu rezultate precise obținute aproape instant. Pentru problema detecției de persoane, proiectul folosește rețeaua YOLO (You Only Look Once) care se potrivește bine cerințelor sistemului, fiind capabilă să proceseze imagini la 40 de cadre pe secundă, mai mult decât capacitățile robotului de 30 de cadre pe secundă.

Integrarea dintre robot și rețeaua YOLO se face separat pe o altă mașină, cu putere de procesare mare. Pentru a obține fluxul de imagini al robotului, sistemul folosește modulul ALVideoDevice din NAOqi care întoarce două tipuri de imagini, color și de adâncime.

În imaginea de mai jos, sunt prezentate rezultatele rețelei YOLO, în stânga pe imaginea color, iar în dreapta pe imaginea de adâncime. Fiecare detecție obținută de rețea are și un scor asociat, care reprezintă scorul de încredere corespunzător detecției.

Camerele prezintă un mic defazaj, care este rezolvat prin deplasarea pe axele x și y a detecțiilor cu numărul corespunzător de pixeli.

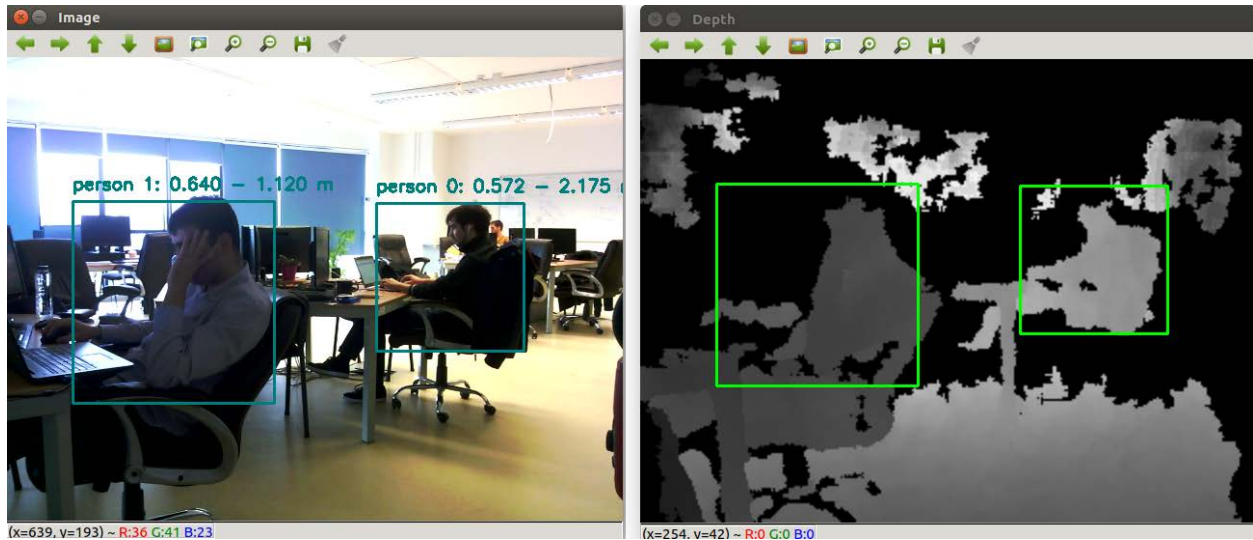


Fig. 8. Rezultatul rețelei YOLO aplicată pe imaginile robotului Pepper.

Pentru a face robotul să-și orienteze capul către fața omului, fiecare detecție este transformată din dreptunghi într-un pătrat corespunzător zonei superioare a corpului (fig.8) Distanța de la robot la persoana detectată este obținută calculând valoarea mediană a pixelilor în zona de detecție. Pentru o precizie mai bună a valorii distanței, fiecare detecție trece printr-o rețea de segmentare, care obține doar pixelii care aparțin persoanei din imagine, iar valoarea mediană se obține doar din pixelii respectivi. Valoarea mediană se aplică pe imaginea de adâncime, care este o matrice în care fiecare pixel reprezintă distanța către acel punct, cu valori între 0.4 și 3.68 metri. Problema cu această imagine este faptul că pentru obiectele foarte apropiate sau foarte depărtate valoarea este aceeași. Pentru a rezolva această problemă am luat în considerare dimensiunea detecției, astfel că dacă este mai mare decât un prag prestabilit, atunci se consideră că obiectul se află la mai puțin de 0.4 metri.

Estimarea poziției 3D a persoanei detectate se face pe baza aproximărilor unghiulare din image și pe baza parametrilor camerei robotului. Figura de mai jos arată geometria pe care se bazează estimarea poziției (fig. 9).

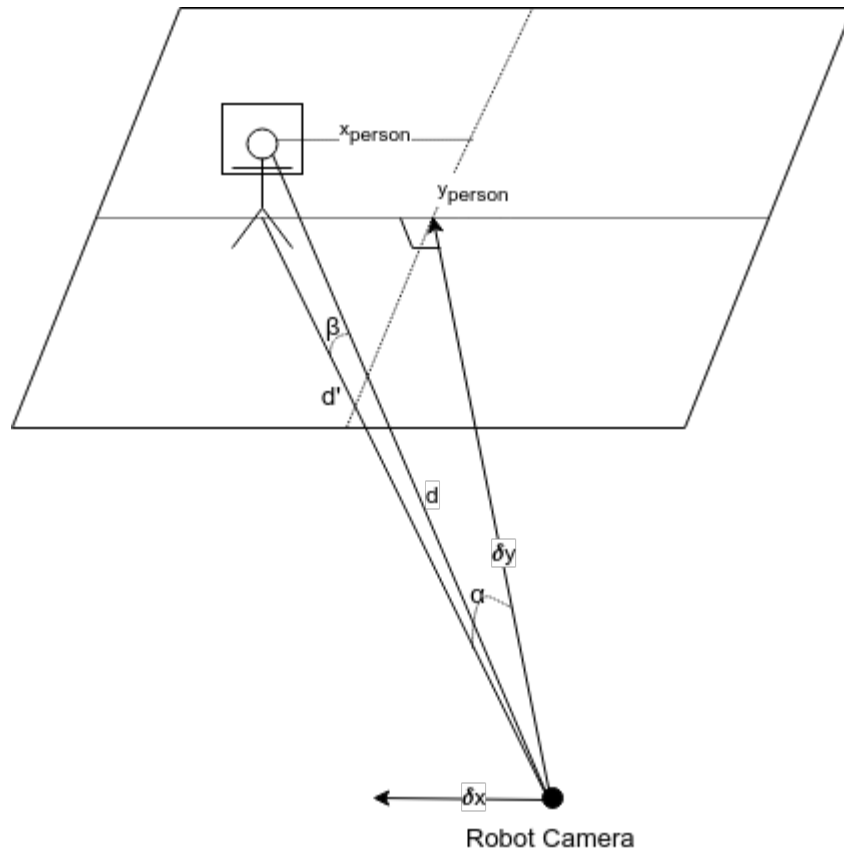


Fig. 9. Metoda de estimarea a poziției 3D a obiectelor detectate.

Unghiurile  $\alpha$  și  $\beta$  sunt calculate pe baza poziției detecției în imagine față de centrul imaginii relativ la unghiurile maxime ale camerei. Astfel că, formulele de calcul ale unghiurilor  $\alpha$  și  $\beta$  sunt:

$$\alpha = \text{hfov\_camera} * (x\_person / \text{width\_image})$$

$$\beta = \text{vfov\_camera} * (y\_person / \text{height\_image})$$

Pentru a calcula deplasarea reală față de poziția robotului se ia în considerare distanța calculată pentru persoana detectată, astfel că:

$$\delta x = d * \cos(\beta) * \sin(\alpha), \quad d = \text{distan}$$

$$\delta y = d * \cos(\beta) * \cos(\alpha)$$

Astfel că estimarea 3D a unei persoane detectate este  $(d, \delta x, \delta y)$ .

Recunoașterea fețelor este folosită pentru a identifica persoana cu care robotul interacționează. Pentru recunoașterea fețelor se folosește rețeaua FaceNet, care oferă rezultate precise într-o varietate de condiții de luminozitate și orientări ale feței. Rezultatul rețelei include poziția feței în imagine și numele persoanei detectate, cu o probabilitate asociată. Informațiile obținute de această rețea sunt combinate cu informațiile obținute de sistemul YOLO, astfel încât dacă

YOLO nu a detectat o persoană care să se potrivească cu o detecție de față, aceasta nu va fi luată în considerare (fig.10).

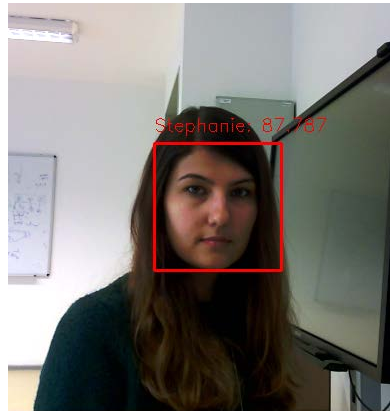


Fig. 10. Rezultatul rețelei FaceNet aplicată pe imaginile robotului Pepper.

### 4.3. Integrare cu sisteme de context externe

Integrarea cu sistemul CAMI a fost realizată modular prin realizarea unor interfețe multi-modale compatibile cu funcționalitățile robotului Pepper. Toate interfețele grafice sunt optimizate atât pentru afișare desktop cât și mobile pentru a asigura o interacțiune optimă folosind tableta robotului. De asemenea robotul redă informațiile esențiale și vocal permițând totodată și interacțiunea cu interfața grafică prin comenzi vocale.

Modulul de notificări implementează la fel ca cel de interacțiune în limba vorbită un nod ROS ce este responsabil de lansarea comportamentelor legate de notificări. Robotul în momentul primirii unei comenzi de la modulul de notificări va pleca în căutarea persoanei căreia îi este adresată notificarea. Comportamentul va fi prezentat detaliat în capitolul Arhitectura platformei de comportamente și a ciclului de viață al robotului. În momentul în care persoana căutată este găsită, pe tableta va fi afișată o interfață multimodală ce permite interacțiuni tactile și interacțiuni în limba română folosind modulul descris în Interacțiune vocală în limba română. În interfață este prezentată notificarea, pe care robotul o va citi utilizatorului și permite comenzi precum „următoarea”, „înapoi”, „ok” și „anulează”.

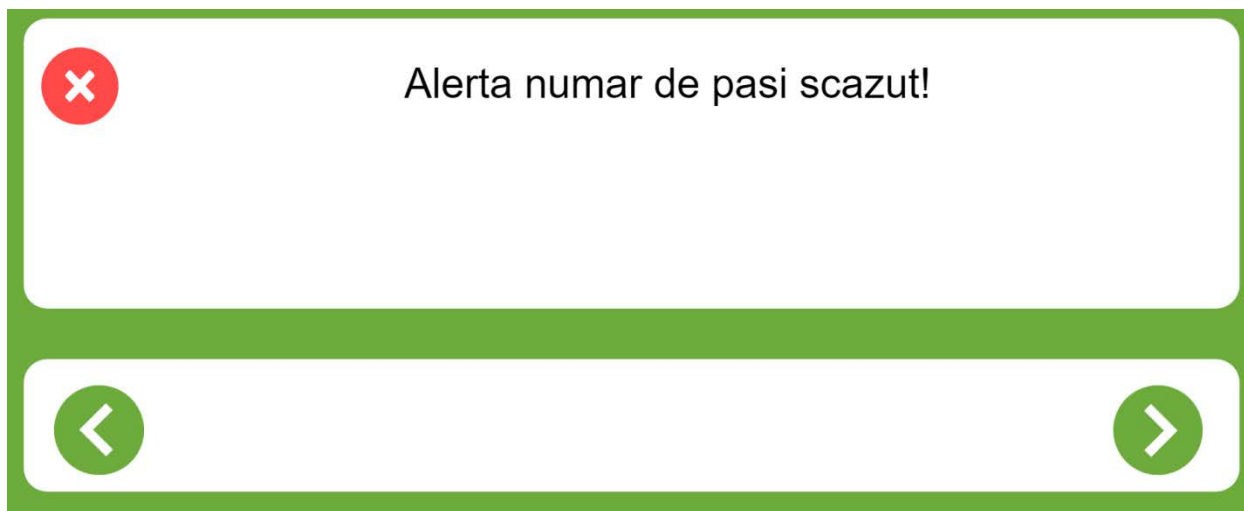


Fig. 11. Exemplu de notificare afișată pe tableta robotului.

Modulul de afișare grafică a informațiilor legate de starea de sănătate a utilizatorului (fig.11 și fig. 12) este implementat pentru a putea fi vizualizate optim pe tableta robotului. Robotul poate primi comenzi pentru afișarea informațiilor legate de puls, tensiune arterială, număr de pași, somn și greutate. Interfața afișează ultimele 20 de măsurători, grafic la fel ca în următoarea figură.

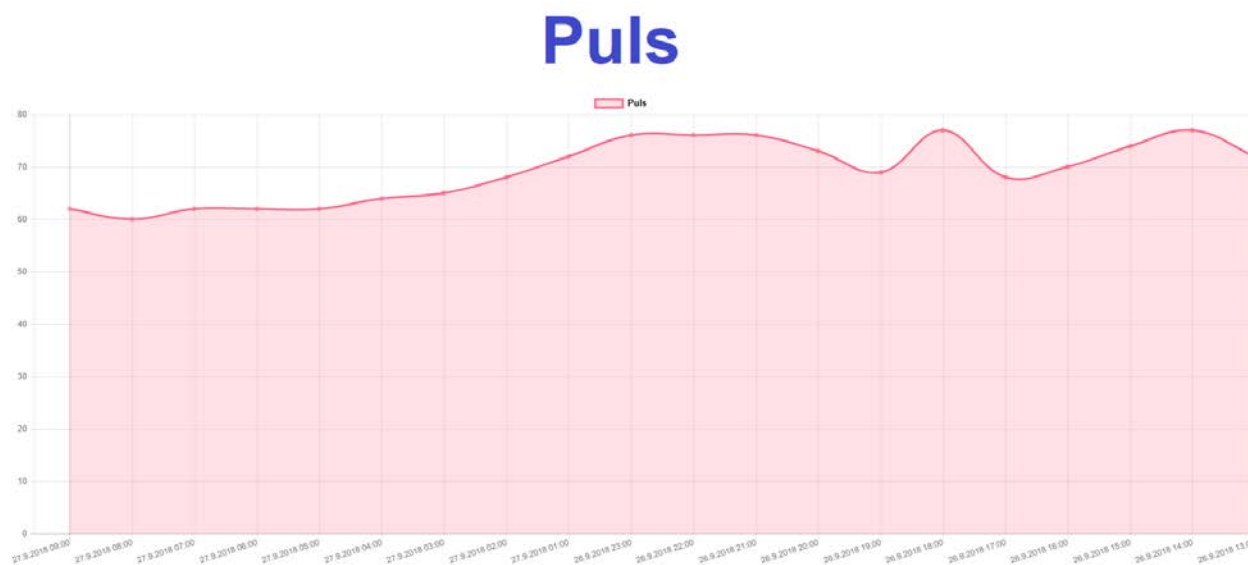


Fig. 12. Exemplu de măsurători de sănătate afișate pe tableta robotului.

Datele despre starea de sănătate sunt preluate din platforma CAMI prezentată în [Scenarii de validare](#). Interfața este web și preluată de robot de la un server web local, putând prezenta orice fel de informații. Implementarea permite o extindere facilă pentru a integra alte sisteme ce expun interfețe web. Pentru a realiza o nouă interfață pentru robot, tot ce trebuie realizat este definirea unor clase CSS ce sunt preluate de robot pentru a fi redactate vocal sau pentru a interacționa cu elemente din interfață la primirea comenzilor predefinite.

#### 4.4. Modulul de Prezentare și Amuzament

Modulul de prezentare și amuzament se referă la aplicațiile instalate pe robot, care pot atrage oamenii prin gesturi și vorbire.

- Recunoașterea fețelor - comportamentul de recunoaștere a persoanei este un modul care încearcă să recunoască persoana din fața sa. Acesta este activat de comanda vocală "Who am I?". Pentru a ști că acest comportament a început, am adăugat o fraza suplimentară înainte de recunoaștere: "Let me think.". După aceea, robotul va căuta în baza sa de date și după ce se va găsi un rezultat, acesta va spune numele persoanei dacă este cunoscut. Dacă există mai multe potriviri, robotul va spune toate numele potrivite. Dacă nu există nici un nume potrivit, acesta va răspunde că nu cunoaște persoana.
- Învățarea unei fețe - acest modul este util pentru a adăuga o persoană în baza de date a robotului. Aceasta este declanșată de comanda "Remember me". După aceea, utilizatorul trebuie să-și spună numele și să stea în fața robotului timp de câteva secunde, pentru a permite fotografierii clare ale feței.
- Mișcare comandată vocal - Am adăugat un comportament simplu pentru mișcare, astfel încât robotul să poată fi mutat pe baza comenzilor vocale. Are 4 opțiuni: "Move forward", "Move backward", "Move left" și "Move right". Robotul se va deplasa în direcția specificată cu un metru și numai dacă nu există niciun obstacol.
- Dans pe muzică - Această funcționalitate implică mai multe module: recunoașterea vorbirii, animația și redarea audio. Este activată de o comandă vocală "Play music". După aceasta, utilizatorul poate alege între 3 tipuri diferite: "disco", "rock" și "thai". Robotul va reda o melodie bazată pe opțiunea selectată și va începe să danseze în consecință.
- Realizarea de fotografii - Comportamentul se activează prin comanda vocală "Take a picture", iar pentru aceasta utilizatorul trebuie să atingă mâna robotului. Când robotul sesizează atingerea, va face poza și o va afișa pe tabletă.
- Prezentarea laboratorului - În cadrul acestei aplicații robotul vorbește despre laborator, despre teme de cercetare din cadrul acestuia, și afișează imagini pe tabletă cu o parte din activitățile surprinse în laborator.
- Recunoașterea emoțiilor - Acest modul se activează atunci când o față este detectată. Dacă se recunosc emoțiile "sad" sau "neutral", atunci robotul îi spune utilizatorului să se înveselească, trimițându-i un sărut.
- Prezentarea scopului robotului - Acest modul presupune că robotul își prezintă misiunea în cadrul laboratorului și lucrurile pe care le poate face să fie cât mai uman. Acest modul este augmentat prin muzică și dans.
- Imitarea animalelor - Această funcționalitate presupune ca robotul să imite diferite animale. Cele 3 animale pe care le poate imita sunt șoarece, gorilă și elefant.

## 5. Arhitectura platformei de comportamente și a ciclului de viață al robotului

Capacitatea robotului de a executa scenarii complexe este dată de modulul de planificare. Modulul de planificare este responsabil cu rularea comportamentelor. Un comportament este format dintr-o serie de obiective ce pot fi rulate atomic. Lista de obiective atomice ce poate fi rulată este următoarea:

- Navigarea la o poziție (x, y) de pe hartă
- Redarea vorbită a unui text predefinit
- Explorarea câmpului vizual pentru a identifica o persoană sau obiect
- Afișare pe tabletă

Obiectivele sunt combinate într-un arbore binar. Fiecare obiectiv poate fi privit ca un nod în acest arbore. Rularea acestui obiectiv este finalizată prin succes sau eșec ce va duce la rularea nodului corespunzător. În acest mod se pot realiza comportamente complexe. Spre exemplu comportamentul de găsire a unei persoane este definit ca în fig. 13.

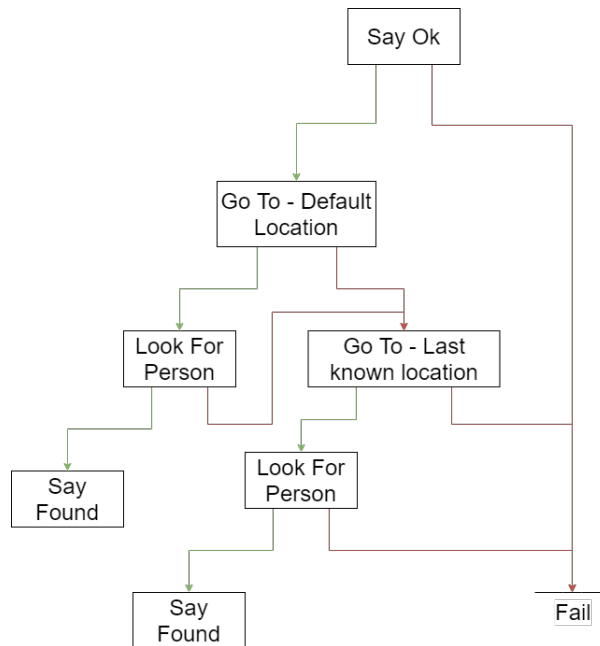


Fig. 13. Comportamentul de găsire a unei persoane în mediu.

Arhitectura permite rularea mai multor comportamente. În momentul în care două comportamente se declanșează, fiecare din nodurile obiectiv sunt adăugate în coada de priorități. Spre exemplu dacă un obiectiv de navigare este adăugat în coadă, iar ulterior unul de redare vorbită, obiectivul de navigare este pus pe pauză, se rulează redarea vorbită, iar după aceea robotul revine la navigare. Acest lucru permite interacțiuni simultane cu robotul și acesta poate îndeplini mai multe obiective cu un număr minim de întreruperi. Arhitectura sistemului de planificare poate fi observată în figura următoare.



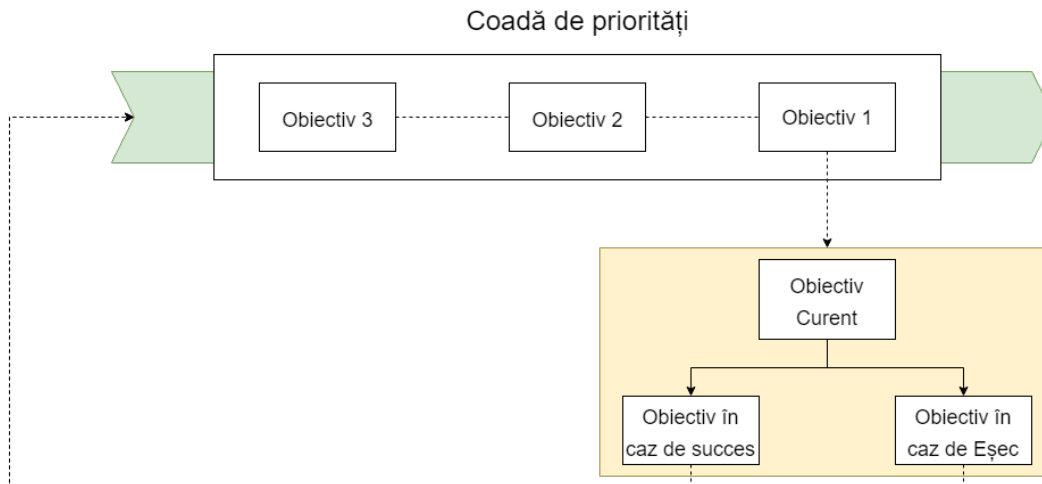


Fig. 14. Coadă de priorități pentru executarea comportamentelor robotului.

Modulul asigură într-un mod similar capacitatea declanșării de comportamente predefinite prin interfețele grafice afișate pe tableta robotului sau prin orice alt modul ce implementează un ROS bridge (fig. 14). Un ROS bridge oferă capacitatea unor sisteme externe, ce nu rulează noduri de ROS să interacționeze cu sisteme ROS. De aceea funcționalitatea este potrivită pentru oferirea unor interacțiuni externe prin publicarea de mesaje JSON standard pe un topic oferite de modulul de planificare „/commands”. Formatul JSON trebuie să conțină comportamentul definit prin cheia „intent”, parametrii obligatorii pentru comportamentul ales și parametrii opționali ce pot fi folosiți.

În exemplul următor este definit un comportament ce va lansa afișarea datelor meteorologice din București din data descrisă (fig. 15).

```

▼ object {3}
  intent : say
  ▼ mandatory_entities [1]
    ▼ 0 {1}
      target : weather
  ▼ optional_entities [1]
    ▼ 0 {2}
      datetime : 20.09.2018
      location : Bucharest

```

Fig. 15. Exemplu de mesaj JSON care se traduce în comportamentul "say" al robotului.

## 6. Interacțiune vocală în limba română

Interacțiunea cu robotul Pepper poate fi făcută atât în limba engleză, cât și în limba română. Sistemul funcționează la fel, indiferent de limba selectată. Pentru recunoașterea vorbirii am folosit serviciile de recunoaștere de la Google, care oferă rezultate precise în timp real. Pentru a parsa și înțelege comportamentul dorit dintr-o propoziție [9], proiectul utilizează serviciul wit.ai [10].

Wit.ai este sistemul care extrage entitățile relevante din propozițiile primite ca parametru, în cazul de față acțiunea dorită și parametri asociați acesteia. Sistemul a fost antrenat înainte de utilizare cu propoziții specifice scenariilor proiectului, iar entitățile definite în procesul antrenării sunt următoarele:

- intent = specifică tipul acțiunii care reiese din propoziție. Valorile posibile sunt: *hello* (salută pe cineva), *stop*, *say* (spune ceva), *go to* (navighează către o poziție), *find* (găsește o țintă în mediu), *look* (se uită împrejur după o țintă), *health* (afișează statistici de sănătate), *reminders* (afișează notificările unei persoane)
- target = parametrul pentru acțiunile de tipul *say*, *go to*, *find*, *look* și *reminders*.
- health\_entity = parametrul pentru acțiunile de tipul *health*

Un exemplu de extragere a entităților din propoziție este în imaginea de mai jos, unde propoziția introdusă este "go to Stephanie" (fig.16).

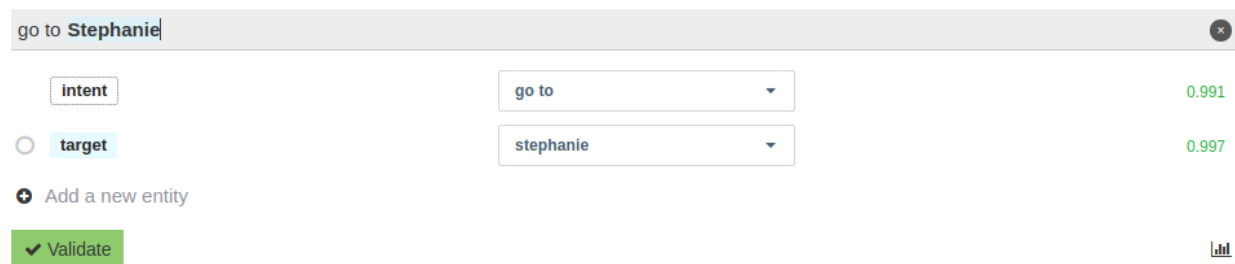


Fig. 16. Exemplu de răspuns al serviciului wit.ai.

La nivel de arhitectură, sistemul de înțelegere a vorbirii are la bază modelul publisher-subscriber. Modulul care publică informații, publisherul, detectează fluxul audio și obține propozițiile folosind serviciul Google. Modulul care așteaptă informații, subscriberul, preia propozițiile și le parsează folosind serviciul wit.ai. Cele două module lucrează asincron, independent unul de celălalt. O reprezentare grafică a arhitecturii este în fig.17.

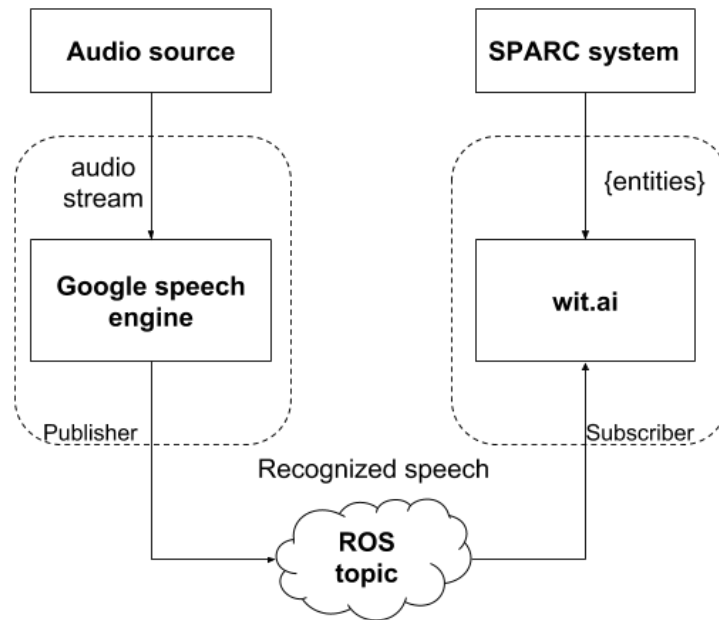


Fig. 17. Arhitectura modului de interacțiune vocală.

## 7. Interfața de declanșare a modulelor comportamentale

Interfața de declanșare a modulelor comportamentale expune toate comportamentele ce pot fi rulate de robot în cadrul sistemului SPARC. Orice declanșare a comportamentelor se realizează prin trimiterea unui mesaj JSON în formă predefinită către nodul de planificare descris în Arhitectura platformei de comportamente și a ciclului de viață al robotului.

Behaviour Manager

Intent

say

Entities

target

weather

datetime

02 / 07 / 2018

Submit

Fig. 18. Interfața de declanșare a modulelor comportamentale.

Orice interfațare cu sistemul se realizează prin ROS bridge-ul descris anterior. Interfața realizată se poate utiliza pentru a declanșa toate cele 7 comportamente (look, say, stop, hello, find, go\_to, display) predefinite prin selectarea comportamentului și introducerea argumentelor corespunzătoare.

## 8. Stagii masteranzi și doctoranzi

Pe parcursul acestei etape, s-au realizat stagii de practica efectuate de un doctorand și un masterand la agentul economic, Centrul IT pentru Știință și Tehnologie SRL (CITST), un IMM orientat spre cercetare și dezvoltare de produse inovatoare.

Stagiile de practică s-au desfășurat precum urmează.

*Alexandra Ghită*, student masterat până în iulie 2018 și proaspăt studentă la doctorat în anul I, școala doctorală de Calculatoare și Tehnologia Informației a UPB, a desfășurat un stagiu în perioada Mai – Iunie 2018 (fiind studentă la masterul de Inteligență Artificială a UPB). În timpul stagiului a efectuat teste și dezvoltări legate de identificarea persoanelor de către robot în condiții variate de lumină.

*Mihai Trăscău*, student doctorand în cadrul școlii doctorale de Calculatoare și Tehnologia Informației a UPB, a desfășurat un stagiu în perioada Iulie - August 2018. În timpul stagiului a evaluat și testat capacitatea robotului de recunoaștere a activității persoanei (e.g. stat în picioare, șezut pe scaun, șezut pe canapea) pe baza camerei RGB-D a acestuia.

## 9. Concluzii

În proiectul SPARC, obiectivul principal este acela de a înzestra roboții disponibili operatorului economic CITST cu o serie de comportamente de bază care pot fi combinate în episoade de interacțiune complexe, după nevoia clienților operatorului economic.

În consecință obiectivele tehnice asociate au fost:

- Dezvoltarea unei baze de date de comportamente (a se vedea Secțiunea 4)
- Dezvoltarea unui planificator care primește scopuri și care descompune comportamente și sarcini complexe în comportamentele simple (a se vedea Secțiunea 5)
- Realizarea managerului de scenariu de lucru al robotului, împreună cu o interfață grafică intuitivă (a se vedea Secțiunea 7)
- Adaptarea componentelor de recunoaștere a vorbirii la limba română (a se vedea Secțiunea 6)

De remarcat este faptul că toate modulele comportamentale, precum și integrarea unor sisteme externe robotului s-a realizat folosind biblioteci existente din sau dezvoltând noduri proprii peste frameworkul ROS<sup>5</sup>.

Acest mod de lucru permite un grad mare de adaptabilitate a soluției dezvoltate la mai multe categorii de roboți, printre care și cei disponibili operatorului economic CITST (Pepper și Tiago), și de aceea este de natură a aduce plus-valoare semnificativă.

### Module dezvoltate

Pentru a atinge obiectivele propuse au fost dezvoltate următoarele tipuri de module:

- mapare a mediului înconjurător și salvare ca hartă pentru navigare ulterioară
- navigare în hartă salvată cu ocolirea obstacolelor dinamice
- detecția de persoane și plasarea lor pe hartă ca obiect dinamic
- recunoașterea persoanelor după față și plasarea lor pe hartă ca obiect dinamic
- recunoașterea codurilor QR pentru memorarea poziției obiectivelor de interes (e.g. laboratorul 308, laboratorul 303, lifturi de pe hol, toaleta)
- recunoașterea obiectelor de interes general (e.g. obiecte de mobilier - scaun, masa, tabla)
- modul de compoziție ierarhică a comportamentelor de bază (sub formă de arbore binar)
- modul de transmitere a comenzilor în limba română
- modul de integrare a unui sistem contextual extern (e.g. comunicare și vizualizare notificări neconfirmate, afișare și comunicare parametrii de sănătate, notificări vreme)
- modul de amuzament (dans și cântat, luarea de fotografii)

### Dezvoltări viitoare

În continuarea dezvoltărilor din proiectul SPARC, echipa de proiect dorește să se concentreze pe următoarele subiecte:

---

<sup>5</sup> Framework-ul ROS: <http://www.ros.org/>

- Îmbogățirea bazei de date de comportamente (e.g. abilitatea de a urmări un utilizator, cu ocolire de obstacole, abilitatea de a recunoaște care utilizator vorbește dintr-un grup de persoane, îmbogățirea dialogurilor existente, sinteza de voce în limba română)
- Îmbunătățirea modulului de planificare a ciclului de viață al robotului prin modelarea comportamentelor sub forma paradigmei Belief Desire Intent (BDI) din domeniul sistemelor multi-agent
- Îmbogățirea percepției contextuale a robotului prin integrarea lui drept consumator de informații deduse pe baza altor senzori dintr-un mediu inteligent (e.g. senzori de prezență, senzori de contact, de temperatura și lumină ambientală)
- Dezvoltarea capabilităților de recunoaștere a activității persoanei (e.g. a se uita la televizor, a citi o carte, a pregăti mâncare) prin combinarea informației vizuale (prin camera proprie) cu cea venită de la sisteme de context externe (e.g. localizarea persoanei într-o cameră, prin senzori de prezență)

Toate aceste obiective de dezvoltare viitoare vor duce la crearea de noi biblioteci ROS, pe care echipa de proiect are intenția de a le licenția în regim open-source către o comunitate largă de cercetători și dezvoltatori în domeniul roboticii.

Exploatarea economică a modulelor existente și viitoare se face prin vânzarea know-how-ului de pregătire și implementare a scenariilor complexe de interacțiune create pe baza comportamentelor de bază.

## Diseminare

Proiectul SPARC a fost diseminat astfel:

- lucrarea acceptata, sustinuta si publicata *Alexandra Ștefania Ghiță, Miruna-Ștefania Barbu, Alexandru Gavril, Mihai Trăscău, Alexandru Sorici, and Adina Magda Florea. "User Detection, Tracking and Recognition in Robot Assistive Care Scenarios." In Annual Conference Towards Autonomous Robotic Systems, pp. 271-283. Springer, Cham, 2018.* prezentată în cadrul conferinței TAROS 2018, Bristol, Anglia, una dintre cele mai prestigioase conferințe în domeniul roboticii (in curs de indexare WOS).

- a fost transmisă spre evaluare lucrarea

*Mihai Trăscău, Mihai Nan and Adina Magda Florea, „Action Recognition using 3D Skeletal Joints”* la revista *SENSORS*, Impact Factor: 2.475 (2017) ; 5-Year Impact Factor: 3.014 (2017)  
**Manuscript ID: sensors-387078**

De asemenea, proiectul a fost popularizat prin intermediul aparițiilor în media națională (e.g. Digi TV<sup>6</sup>, Antena 3<sup>7</sup>, Realitatea, TVR), precum și prin participarea la diverse expoziții de popularizare a dezvoltărilor tehnologice (e.g. Bucharest Tech Week<sup>8</sup>, Tech Fest<sup>9</sup>)

<sup>6</sup> Digi TV - Emisiunea Jurnalul de Științe, 16 iunie 2018 - <https://www.digi24.ro/emisiuni/jurnalul-de-stiinte/jurnalul-de-stiinte-16-iunie-16-30-948209>

<sup>7</sup> Antena 3 - Emisiunea În Premieră - Vin Roboții - [https://www.youtube.com/watch?v=h\\_Mk8WwXAsw](https://www.youtube.com/watch?v=h_Mk8WwXAsw)

<sup>8</sup> Bucharest Tech Week: <https://techweek.ro/>

<sup>9</sup> Tech Fest Bucharest: <https://2018.techfest.ro/bucharest/>

## Bibliografie

[1] Stuckless, R., "Developments in real-time speech-to-text communication for people with impaired hearing", In M. Ross (Ed.), "Communication access for people with hearing loss", pp. 197-226, MD: York Press, Baltimore, 1994.

[2] A Prochazka et.al. Microsoft Kinect Visual and Depth Sensors for Breathing and Heart Rate Analysis, Sensors, 16(7), 2016, <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4970046/> accesat dec 2017

[3] Ahmad Jalal, Shaharyar Kamal, and Daijin Kim, Human Depth Sensors-Based Activity Recognition Using Spatiotemporal Features and Hidden Markov Model for Smart Environments, Journal of Computer Networks and Communications, Volume 2016 <https://www.hindawi.com/journals/jcnc/2016/8087545/>, accesat dec 2017

[4] Vennila Megavannan, Bhuvnesh Agarwal, R. Venkatesh Babu, Human Action Recognition using Depth Maps, [http://www.serc.iisc.ernet.in/~venky/Papers/kinect\\_action\\_recognition\\_spcom12.pdf](http://www.serc.iisc.ernet.in/~venky/Papers/kinect_action_recognition_spcom12.pdf), accesat dec 2017

[5] Miriam Buonamente, Haris Dindo, and Magnus Johnsson, Action Recognition based on Hierarchical Self-Organizing Maps, 2015, <http://ceur-ws.org/Vol-1315/paper7.pdf> accesat dec 2017

[6] Yong Du, Wei Wang, Liang Wang, Hierarchical Recurrent Neural Network for Skeleton Based Action Recognition [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2015/papers/Du\\_Hierarchical\\_Recurrent\\_Neural\\_2015\\_CVPR\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2015/papers/Du_Hierarchical_Recurrent_Neural_2015_CVPR_paper.pdf) accesat dec 2017

[7] Amir Shahroudy et.al. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis, 2015 [https://www.cv-foundation.org/openaccess/content\\_cvpr\\_2016/papers/Shahroudy\\_NTU\\_RGBD\\_A\\_CVPR\\_2016\\_paper.pdf](https://www.cv-foundation.org/openaccess/content_cvpr_2016/papers/Shahroudy_NTU_RGBD_A_CVPR_2016_paper.pdf) accesat dec 2017

[8] Forrest N. Iandola, et.al. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size, 2016 <https://arxiv.org/abs/1602.07360> accesat dec 2017

[9] A. F. Gavril, M. Trascau and I. Mocanu, "Multimodal Interface for Ambient Assisted Living," 2017 21st International Conference on Control Systems and Computer Science (CSCS), Bucharest, 2017, pp. 223-230. doi: 10.1109/CSCS.2017.38

[10] <https://wit.ai/>, accesat septembrie 2018.