Roboţii şi Societatea:
Sisteme Cognitive pentru Roboţi Personali şi Vehicule Autonome

Număr contract 72PCCDI **din** 01/03/2018

Etapa IV - 2021

Raport ştiinţific şi tehnic

Consorțiu
Coordonator proiect: UNIVERSITATEA POLITEHNICA DIN BUCURESTI
P1: INSTITUTUL DE CERCETARI PENTRU INTELIGENTA ARTIFICIALA „MIHAI DRAGANESCU"
P2: INSTITUTUL DE MATEMATICA "SIMION STOILOW" AL ACADEMIEI ROMANE
P3: UNIVERSITATEA TEHNICA DIN CLUJ - NAPOCA
P4: UNIVERSITATEA "DUNAREA DE JOS"

Site proiect http://aimas.cs.pub.ro/robin/

1. Introducere

ROBIN este un proiect centrat pe utilizator care proiectează sisteme şi servicii pentru utilizarea roboților într-o societate digitală interconectată și permite companiilor realizarea de produse și servicii complexe, inteligente și performante destinate utilizatorilor și societății în ansamblu. Proiectul se referă la o gamă diversă de roboţi: roboţi asistivi care vin în sprijinul persoanelor cu nevoi speciale, roboţi de interacţiune cu clienţii şi roboţi software care pot fi instalaţi pe vehicule în scopul realizării unei conduceri autonome sau semi-autonome. Proiectul combină tehnici și tehnologii avansate de inteligență artificială, interacțiune om-robot, interacțiune cu un mediu pervasiv, arhitecturi și prelucrări în Cloud, arhitecturi Cloud-Edge si Cloud-Robotics.

Figura 1.1 prezintă conceptul general al proiectului, atât funcțional cât și arhitectural, prin care am dezvoltat sisteme și servicii științifice și tehnologice performante, construite peste platforme interoperabile, cu capacități de structurare și procesare inteligentă a datelor și oferite comunității științifice și agenților economici.

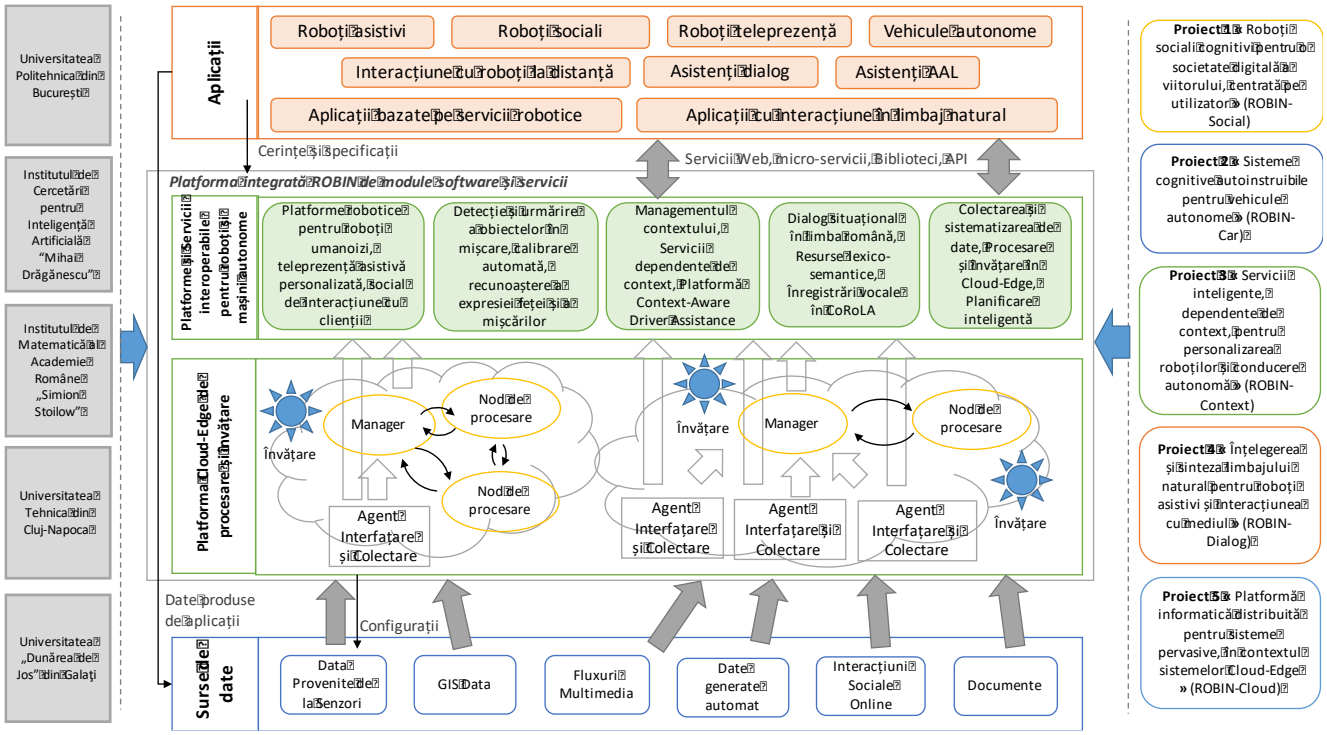

Figura 1.1. Prezentarea conceptului integrat a proiectului ROBIN

Proiectul ROBIN este alcătuit din cinci proiecte componente:

ROBIN-Social: ***Roboți sociali cognitivi pentru o societate digitală a viitorului, centrată pe utilizator.*** Scopul proiectului este realizarea unei soluții integrate și ușor configurabile pentru personalizarea roboților asistivi (pt. persoane cu nevoi speciale) și sociali (pt. clienți), soluție bazată pe tehnici avansate de inteligență artificială care pot oferi roboților un caracter cognitiv și autonom.

ROBIN-Car: ***Sisteme cognitive autoinstruibile pentru vehicule autonome.*** Scopul proiectului constă în dezvoltarea de metode de vedere computaţion- ală care să rezolve o gamă mai largă şi mai sofisticată de sarcini de asistenţă în pilotaj, realizarea unor module inteligente pentru „Hands-off driving" și „Automated driving" și un sistem prototip care va fi testat pe un autovehicul electric semi-autonom.

ROBIN-Context: ***Servicii inteligente, dependente de context, pentru personalizarea roboţilor şi conducere autonomă.*** Scopul proiectului este crearea unei platforme suport pentru definirea / reprezentarea semantică și gestiunea facilă și eficientă a datelor ce devin context în scenarii de asistență robotică personalizată și sisteme ADAS (Advanced Driving Assistance Systems).

ROBIN-Dialog: ***Înțelegerea și sinteza limbajului natural pentru roboți asistivi și interacțiunea cu mediul.*** Scopul proiectului presupune dezvoltarea unei serii de scenarii pentru câteva micro-lumi și tehnologia de prelucrare a limbii române pentru dialoguri situaționale în aceste micro-lumi.

ROBIN-Cloud: ***Platformă informatică distribuită pentru sisteme pervasive, în contextul sistemelor Cloud-Edge***. Scopul proiectului îl constituie crearea unei platforme suport pentru colectarea de date provenind de la senzorii unor sisteme suport pentru roboți (ex. IoT), oferirea de mecanisme de procesare și învățare combinând modele Cloud cu dispozitive aflate aproape de sursa de colectare, oferirea de biblioteci suport pentru procesare inteligentă / semantică a datelor, și în final dezvoltarea de servicii Web centrate spre agenți economici interesați de folosirea datelor stocate la nivelul platformei.

2. Proiectul component ROBIN-Social

Parteneri ai proiectului ROBIN-Social

UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI (CO)
INSTITUTUL DE CERCETĂRI PENTRU INTELIGENȚĂ ARTIFICIALĂ „MIHAI DRĂGĂNESCU"
INSTITUTUL DE MATEMATICĂ "SIMION STOILOW" AL ACADEMIEI ROMÂNE

Validare și testare acceptanță platformă robotică

S-a lucrat la îmbunătățirea platformei AMIRO (AMbIent RObotics, fig.1), platformă software bazată pe ROS și deci utilizabilă pe mai multe platfome hardware robotice. Platforma AMIRO (AMbIent RObotics) a fost dezvoltată pentru pe robotul Pepper[1] (robot existent în laboratorul AI-MAS al UPB și la UTCluj) și este generalizabilă la orice platformă robotică care suportă ROS[2] (Robot Operating System). Pentru a demonstra capabilitatea platformei AMIRO, aceasta a fost portată și instalată pe robotul Tiago[3]. S-a realizat și introducerea serviciului robotic (bazat pe platforma AMIRO) în eeris (activitatea A4.1).

Pe baza îmbunătățirilor realizate, s-a realizat versiunea finala a produsului robot asistiv pentru persoane cu nevoi speciale și use-case-ul robot social de interacțiune cu vizitatorii unui laborator.

Îmbunătățirile realizate în 2021 au fost, în principal:

- Extinderea și îmbunătățirea componentei de recunoaștere a activităților utilizatorului din secvențe video (dezvoltată anterior) utilizând o nouă metodă de recunoaștere activităţior bazată pe scheleți și învățare profundă folosind rețele neurale adânci de tip Graph Convolutional NN (GCN), cât și evaluarea comparativă a diferitelor arhitecturi de recunoaștere propuse în proiect. Am demostrat că soluția propusă obține rezultate comparabile cu SoA dar cu o creștere semnificativă a vitezei de inferență. Aceste rezultate au fost incluse în lucrarea "*Comparison between Recurrent Networks and Temporal Convolutional Networks Approaches for Skeleton-based Action Recognition"*, autori M. Nan, M. Trăscău, A.M. Florea, C. Iacob, Sensors, Vol.21 No.6, Jan 2021 (Q1, IF 3.576). Lucrarea a fost ***"highlighted on the journal homepage" and designated among the "Most notable articles (Feb**-May 2021)"*
- S-a extins colectarea și adnotarea setului de date propriu de recunoaștere a activităților umane (Pepper Data Set), colectat și adnotat anterior de colectivul proiectului cu ajutorul robotului Pepper; acesta poate fi pus la dispoziția celor interesanți la cerere. La ora actuală, setul nostru de date conține imagini preluate de la 17 persoane diferite, fiecare acțiune fiind înregistrată de mai multe ori în diferite unghiuri, poziții și la distanțe diferite (pentru a simula diferitele unghiuri din care robotul poate vedea o persoană). Acest lucru a fost făcut pentru a avea varietate în setul de date și a obține rezultate concludente. Există 100 de configurații diferite pentru o persoană, pentru fiecare au fost preluate 5 imagini, astfel încât setul nostru de date conține 17*100*5 imagini RGB-D (fig. 2).
- Portarea și utilizarea platformei AMIRO pe robotul Tiago[4] și realizarea pe acesta a detecției și recunoașterii de persoane, urmărire persoane (anterior realizate în varianta AMIRO de pe Pepper) cât și a facilității de manipulare de obiecte, facilitate care nu există pe robotul Pepper deoarece acesta nu are, prin hardware, capacități manipulatorii (fig.3).

[1] https://www.softbankrobotics.com/emea/en/pepper

[2] https://www.ros.org/

[3] https://pal-robotics.com/robots/tiago/

[4] https://pal-robotics.com/robots/tiago/

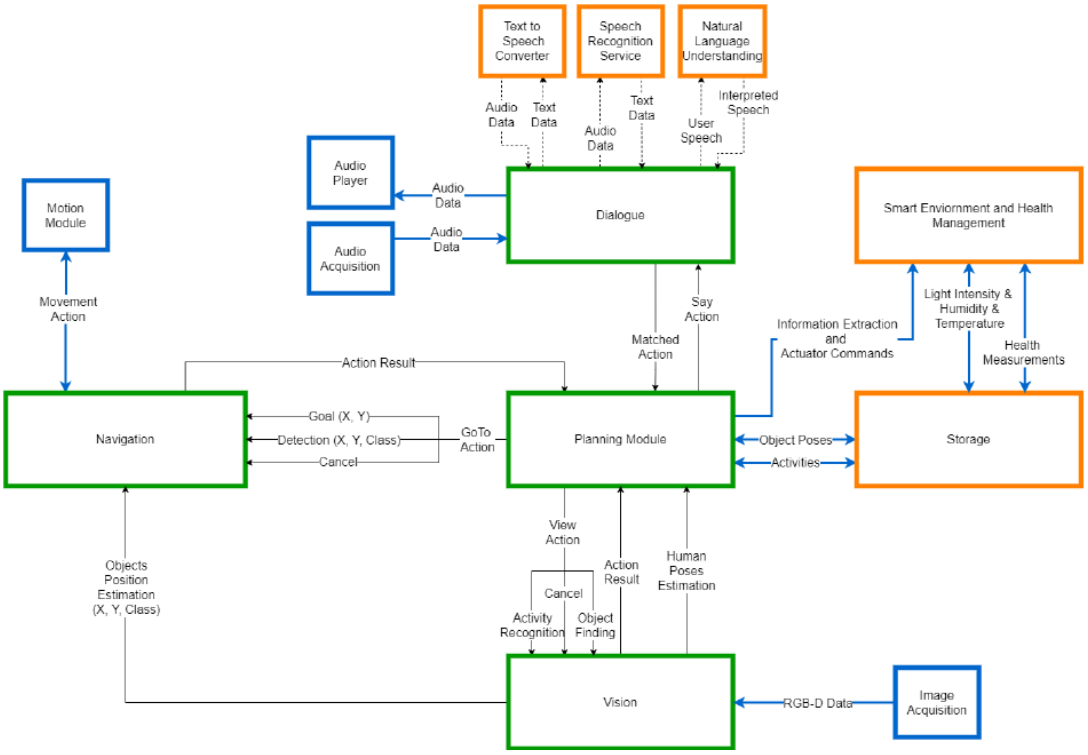

Figura 1. Arhitectura sistemului AMIRO - diagramă bloc. Cu verde sunt marcate componentele cloud-edge, cu portocaliu sunt reprezentate servicii cloud și cu albastru modulele care funcționează pe robot. Săgețile negre reprezintă un apel intern, cele punctate reprezintă un apel la un serviciu cloud și cele albastre reprezintă o comunicare prin ROS.

Figura 2. Colectarea setului de date propriu de recunoaștere a activităților umane

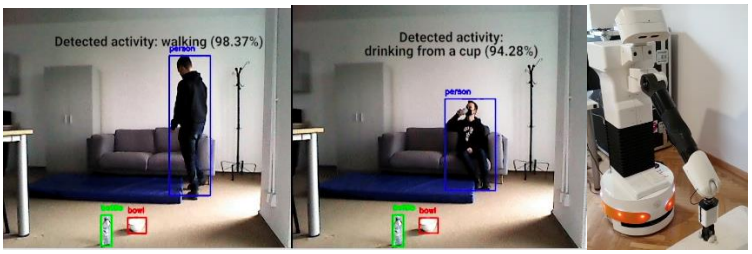

Figura 3. Detecția activităților utlizatorului și manipulare de obiecte cu Tiago

Au fost realizate numeroase teste de performanță ale platformei AMIRO pentru situațiile definite (produs asistiv, use-case vizitare) atât pe setul de date NTU RGB+D[5] cât și pe setul de date proprie colectat. Pentru testare s-au folosit următoarele scenarii:

- (T1) - componenta de recunoaș tere a acțiunii umane a fost antrenată și testată folosind setul de date NTU RGB+D cu 8 acțiuni selectate pentru cele două protocoale propuse de acest set de date.
- (T2) - a fost testată capabilitatea de generalizare a modelului antrenat folosind setul de date NTU RGB+D extins. Astfel, după ce a fost terminat procesul de antrenare folosind cele 8 acțiuni, modelul a fost testat folosind doar exemplele din setul de validare extrase din setul de date realizat de noi în laborator folosind robotul Pepper. Pentru a putea realiza o astfel de testare, a fost nevoie să definim două protocoale similare cu cele propuse de cei care au realizat setul de date NTU RGB+D pentru setul de date realizat de noi : Cross-Subject, și Cross-View.
- (T3) - am pornit de la modelul antrenat folosind setul de date NTU RGB+D și am realizat fine-tuning pe setul de antrenare al setului de date propus de noi.
- (T4) - în acest ultim scenariu, am folosit exclusiv setul de date realizat propus de noi pentru antrenare și testare, pentru a observa calitatea rezușltatelor comparativ cu scenariile prezentate anterior.

Scenariu	Acuratețe	
	CS	CV
Antrenarea pe NTU RGB+D & testarea pe NTU RGB+D – **T1**	78.03%	84.21%
Antrenarea pe NTU RGB+D & testarea pe setul de date realizat folosind robotul Pepper – **T2**	44.81%	43.51%
Antrenarea pe setul de date realizat folosind robotul Pepper & testarea pe setul de date realizat folosind robotul Pepper – **T4**	83.49%	87.02%
Antrenarea pe NTU RGB+D & reantrenarea modelului preantrenat pe setul de date propus & testarea pe setul de date realizat folosind robotul Pepper – **T3**	**91.5%**	**89.75%**

Tabelul 1. Rezultatele experimentale obținute pentru scenariile definite în cazul modelului simplu

Scenariu	Acuratețe	
	CS	CV
Antrenarea pe NTU RGB+D & testarea pe NTU RGB+D – **T1**	75.21%	79.30%
Antrenarea pe NTU RGB+D & testarea pe setul de date realizat folosind robotul Pepper – **T2**	60.37%	45.60%
Antrenarea pe setul de date realizat folosind robotul Pepper & testarea pe setul de date realizat folosind robotul Pepper – **T4**	88.67%	89.12%
Antrenarea pe NTU RGB+D & reantrenarea modelului preantrenat pe setul de date propus & testarea pe setul de date realizat folosind robotul Pepper – **T3**	**92.27%**	**91.63%**

Tabelul 2. Rezultatele experimentale obținute pentru scenariile dfinite în cazul modelului complex

Scenariile diferite au fost propuse pentru a putea valida operația de adaptare a domeniului realizată pentru setul de date realizat folosind robotul Pepper. După cum se poate observa, în cazul ambelor modele, rezultatele sunt vizibil îmbunătățite atât în cazul în care se pornește de la un model care este antrenat pe un set de date mai mare, precum NTU RGB+D, și este specializat pentru o parte din setul de date realizat.

[5] https://github.com/shahroudy/NTURGB-D

În acest fel, proiectul component P1 a realizat o tehnologie nouă de interacțiune robotică (2019) o tehnologie noua prin care s-a dezvoltat platforma robotică AMIRO (20202) și produsul software robot asistiv al persoanelor cu nevoi speciale (2020) care a inclus, suplimnetar față de platforma și produsul dezvoltat in ROBIN, și integrarea unui framework de management al stării de sănătate a utilizatorului (dezvoltat anterior de colectivul de la UPB). Produsul software robot asistiv al persoanelor cu nevoi speciale poate fi controlat de pe diferite dispozitive, de exemplu desktop, tabletă sau telefon mobil. Pe baza platformei AMIRO s-a realizat serviciul robotic care acoperă partea de navigare, vedere artificială a robotului, partea de planificare a acțiunilor robotului și componenta de dialog (2021).

3. Proiectul component ROBIN-Car

Parteneri ai proiectului ROBIN-Car

INSTITUTUL DE MATEMATICĂ "SIMION STOILOW" AL ACADEMIEI ROMANE (CO)
UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI
INSTITUTUL DE CERCETĂRI PENTRU INTELIGENȚĂ ARTIFICIALĂ „MIHAI DRĂGĂNESCU"
UNIVERSITATEA "DUNAREA DE JOS" DIN GALAȚI

Etapa din 2021 s-a concentrate pe testarea dezvoltărilor anterioare pe un autovehicul și realizarea unui prototip de asistare a șoferului în conducere semi-autonomă. Pe baza cercetărilor anterioare, am dezvoltat un simulator de rulare care pemite antrenarea unei rețele neurale adânci end-to-end steering pentru determinarea direcției de urmat a masinii, vitezei pe care aceasta trebuie s-o aiba, daca trebuie sa accelereze sau sa franeze.

Am realizat estimarea adancimii din input monocular (fig 4), în care obiectivul de estimare a adancimii este formulat ca cel de minimizare a erorii fotometrice pentru o reproiectie a unui cadru din trecut (target) in cel curent (source, care e mai recent). Pentru a obtine rezultate bune a fost initial antrenat intai modelul de estimare pe setul de date KITTI, iar apoi s-a efectuat procedura de fine-tuning pe setul de date UPB. Video-urile inregistrate au fost sub-esantionate la 10 cadre pe secunda si au fost mentinute doar acele cadre succesive care au media normei pentru fluxul optic peste 1 (pentru a evita antrenarea pe cadre succesive statice, i.e. unde nu se face nicio miscare).

- *Figura 4: Rezultate ale estimarii adancimii din input monocular. Imagini din setul de date UPB. Randul de sus prezinta imaginea RGB. Randul de jos prezinta imaginea de adancime (disparity map) aferenta.*

Am utilizat în implementarea proprie o arhitectura de tip encoder-decoder pentru a prezice fluxul optic de o maniera end-to-end. Predictiile obtinute prin aceasta metoda sunt ulterior imbunatatite, facand uz de pipeline-ul de antrenare pentru estimarea de adancime, mentionat anterior, pentru a calcula deplasarea la nivel de pixel intre cadrul sursa si cel destinatie. Prin aceste metode se extrage intai un flux rigid (eng. rigid flow field), corespunzand obiectelor care sunt statice in imagine (e.g. sosea, pomi, masini parcate).

Constructia, antrenarea si evaluarea retelei de generare a directiei

Am realizat implementarea unei arhitecturi proprii, de o complexitate mai mica, ce poate fi antrenata mai repede si care va servi drept referinta pentru modelele mai complexe. In cele ce urmeaza descriem arhitectura si modul de antrenare al retelei de referinta pentru predictia directiei.

Setul de date UPB contine 408 de inregistrari video de condus exemplar, acoperind o distanta de 72.7[km], pe o durata de 254 de minute, insumand numarul total de capturi la 458026. Cateva particularitati ale setului de date sunt: prezenta pietonilor, a limitatoarelor de viteza si a barierelor. Dintre cele 408 de inregistrari, 21% au fost efectuate in timpul noptii, corespunzand la 0.9 ore de condus acoperind 13.74[km]. Setul de date cotine 739 treceri prin intersectii, dintre care 172 corspund unuiviraj la dreapta, 164 corspund unui viraj la stanga si 403 mentinerea directieide mers. Viteza medie de condus este de 24.72[km/h] cu o deviatie standard de 1.91[km/h]. Viteza maxima este de 31[km/h]. Setul de date a fost impartit in doua subseturi, 80% pentru antrenare si 20% pentru validare, distribuite astfel incat suprapunerea geografica8 sa fie mini-mala. Inregistrarile video au fost reesantionate la 3 capturi/secunda (FPS). Pentru imaginile originale, cu dimensiunea de 360×640, au fost efectuate urmatoarele etape de procesare: eliminare caroseriei masinii (necesara pentru efectuarea augmentarilor de perspectiva) reducand dimensiunea la 320×640, iar apoi o redimensionare a imaginilor la 256×512.

Arhitectura rețelei
Modelul antrenat urmeaza arhitectura ResNet18 pentru care am inlocuit ultimele doua starturi/nivele (average pooling si linear layer), pentru a se potrivi cu imaginea de intrare si pentru a produce iesirea dorita, reprezentate de o distributie discreta cu 401 valori posibile (bins). Pentru antrenarea modelului am folosit antrenare supervizata pentru a prezice orientarea relativ (rcourse) dupa 0.33[s]. Obiectivul de antrenare (target) urmeaza o distributie Gaussiana avand media centrata in orientarea relativa si deviatia standard de 0.5. Dupa analiza histogramei a orientarii relative dupa 0.33[s], am decis limitarea valorilor de iesire (rcourse) intre $[-20 \circ , +20 \circ]$, justificand astfel discretizare in 401 valori ale iesirii retelei. O precizie de o zecimala s-a dovedit a fi necesare pentru a surprinde mai multe moduri ale distributiei de iesire, indicand astfel rute multiple pe care masina le poate aborda (ex. intersectii). Modelul a fost antrenat pentru 10 epoci, optimizand divergenta Kullback–Leibler intre distributia construita pe baza orientarii relative si distributia prezisa de model. Optimizare s-a relaizat in grupuri de cat 32 (batch size), folosind optimizatorul Adam cu o rata initiala de invatare de 1e-4, redusa dupa 5 epoci cu un factor de 0.1. S-a pastrat modelul care a perfomat cel mai bine pe setul de validare considerand metrica de evaluare divergenta Kullback– Leibler.

In timpul antrenarii au fost aplicate metode clasice de augmentare, reprezentate de schimbari in: luminozitate, contrast, saturatie si nuanta. Setul de date a fost extins prin introducerea agumentarilor perspective 2D, care simuleaza redresarea masinii din situatii de condus eratic. Pentru fiecare imagine de antrenare, cu aceesi probabilitate, s-a aplicat o translatie si/sau o rotatie. Datorita limitarii procedurii de augmentare prespectiva, deformarea obiectelor si lipsa informatiei vizuale, am eliminat din imagine zona situate deasupra liniei de orizont, ceea ce reprezinta 40% din latimea imaginii, si am redus lateralel cu 50%. Experimentele anterioare au deomnstrat ca modelul poate prezice comanda corecta bazandu-se pe zonele cu informatie vizuala lipsa care au forma puternic corelata cu deplasare aplicata, fortandu-ne sa reducem semnificativ campul vizual.

Evaluare
Am propus un nou sistem de evaluare (fig. 5) care reduce seminificativ deformările obiectelorsituate deasupra liniei orizontului și nevoia de a reduce semnificativ câmpul vizual alimaginilor rezultate. Pentru fiecare captură dintr-un video de testare, mai întâi estimămharta de adâncime, aplicăm o transformare perspectivă corespunzătoare deplasării și orientăriiactualizate a mașinii, calculate la pasul anterior folosind modelul Ackermann. Transformareaperspectivă este trimisă împreună cu masca pixelilor valizi modulului de inpaintig care completează informația lipsă care nu era prezentă în campul vizual inițial sauaparută în urma ocluziilor. Imaginea sintetizată este dată ca intrare modulului de prezicere atraiectoriei care furnizează următoarea comandă. Conform comenzii prezise, simulatorul actualizează poziția și orinetarea internă a vehiculului simulat, ceea ce reprezintă sfârșitulunui cliclu. Aceeași procedură este aplicată urmatoarelor capturi.

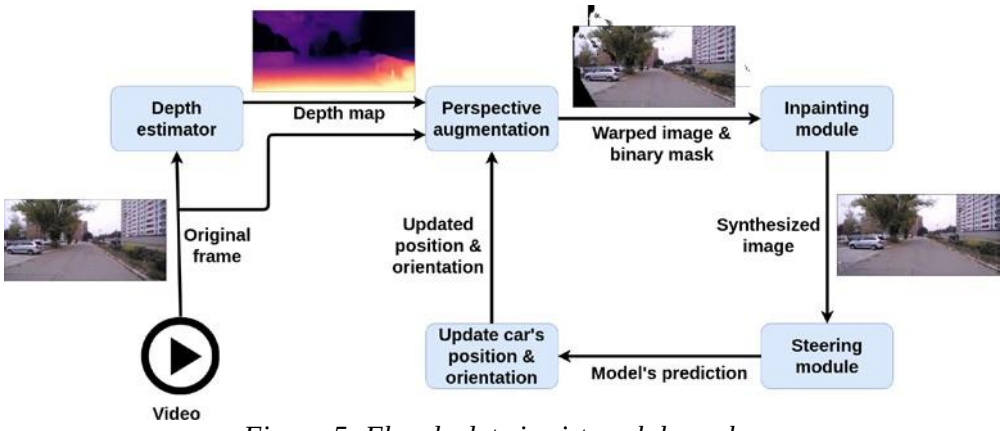

Figura 5: Flux de date in sistemul de evaluare.

Am efectuat o evaluare open-loop și closed-loop a modelelor în simulatorul original (trans- formări 2D) și versiunea îmbunătățită (transformări 3D). Pentru evaluare open-loop, am folosit divergența KL ca metrica între distribuția de referință și distribuția de iesire prezisă de model. Pentru evaluare closed-loop am raportat autonomia vehiculului și numărul de intervenții. Am considerat o intervenție dacă deplasarea dintrevehiculul simulat și cel real în valoare aboslută depașește1.5metri sau dacă diferența deorientare între cele două vehicule în valoare absoultă depașește 0.2radiani. Pentru a testa senzitivitatea simulatorului, am considerat un model de referință (baseline) care prezice întotdeauna o orientare relativă de 0. Perfromanța scazută a modelului dereferintă demonstrează faptul ca simulatorul este capabil sa detecteze miscări fine ale vehicului simulat. Rezultatele simulărilor sunt prezentate în tabelul 3.

Am evaluat rețeaua de prezicere a traiectorie în versiunea îmbunătățită a simulatorului și nu am identificat diferențe semnificative de performanță față de evaluările percedente. Rezultatele sugerează faptul că aplicând regularizare adecvată, modelul de condus poate beneficia de augmentările perspective, chiar dacă imaginile sintetizate sunt puternic distorsionate.

			Open-loop(KL)		Closed-loop(2D)		Closed-loop(3D)	
Model	View	t_{max}	μ	σ	A	NI	A	NI
Baseline	*	-	1.797	2.504	0.45	620	0.45	620
B	Cropped	-	0.792	0.901	0.62	313	0.63	298
BS	Cropped	0.75	0.828	1.017	0.74	181	0.73	190
BS	Cropped	1.00	0.828	0.971	0.74	177	0.74	179
BS	Cropped	1.25	0.837	0.964	0.75	173	0.74	183
B	Full	-	0.792	0.813	-	-	0.60	346
BD	Full	-	0.729	0.888	-	-	0.61	323
BDF	Full	-	0.483	0.616	-	-	0.60	347

Tabelul 3: Rezultatele evaluărilor pentru cele două experimente pe o durată de 3060.16[s].Modelul de referintă (baseline) corespunde modelului care prezice întotdeauna 0◦. Abrevieri:balansarea setului de date (B) în timpul antrenării, harta de adâncime (D) furnizată caintrare, fluxul optic (F) furnizat ca intrare, autonomia (A), numărul de intervenții (NI), nuse aplică (-)

Am arătat că simulatorul propus poate evalua cu succes poltici de condus antrenate pe întreaga rezoluție și care primesc la intrare informație perceptuală adițională sub forma harții de adâncime sau flux optic, inutilizabile în versiunea originală datorită distorsiunilor obiectelor. Analiza curbelor de antrenare pe setul de validare sugerează faptul că setul de date UPB pare a fi nereprezentativ pentru învătarea prezicerii traiectoriei. Având la dispoziție un câmp vizual extins, modelul are capabilitate de a surpinde mai multe moduri în distribuția de ieșire, corespunzând mai multor traiectorii posibile pe care vehiculul le poate urma. Datorită faptului că nu conditionăm ieșire modelului pe o traiectorie predefinită, și datorită faptului că în timpul simulării alegem ce mai probabilă acțiune, este posibil să se aleagă o traiectorie diferită față de cea înregistrată, și astfel un element de șansă poate influența rezultatele evaluării. Nu în ultimul rând, calitatea imaginilor sintetizate poate reprezenta un alt factor care să influenteze performanța modelului, deși în aceste situații, deviația este deja semnificativă și o intervenție umană este iminentă.

În final, evidențiem discrepanțele între evaluare open-loop și closed-loop ale politicii de condus. Similar cu experimentele precedente, observăm că evaluare open-loop supraestimează performanța reală a modelului. Dându-se istoricul actiunilor de referință, o rețea poate extrapola traiectoria vehicului în timpul evaluării open-loop, astfel reușind să reducă eroare predicției. În contrast, în scenariul closed-loop, modelul calculează predicția pe baza observațiilor care depind de acțiunile precedente, astfel reușind să estimeze mult mai bine performanța modelului în lumea reală. Am constatat ca folosirea unei rezolutii mai mari a imaginii de intrare (256x512) conduce la rezultate mai bune a factorului de autonomie si a numarului de interventii.

De asemenea, pentru testarea soluției pe autovehicul am echipat un autoturism Dacia Logan electric cu senzorii și actuatorii necesari, și am realizat sistemele care colecteaza si filtreaza datele de la senzori, precum si cele care transfera decizia asupra directiei traiectoriei de urmat catre actuarea automatizata a coloanei de directie. In particular, am realizat:

- antrenarea si evaluarea retelei de generare a traiectoriei (steering), in vederea adaptarii la capabilitatile noilor senzori de tip camera RGB
- implementarea unui pipeline de colectare, sincronizare si fuziune a datelor provenite de la senzorul inertial, cel GPS si viteza la roata a masinii in vederea corectarii valorii de viteza reale a masinii.

Rezultatul remarcabil a fost reprezentat de primul autoturism autonom care s-a deplasat în campusul UPB fără intervenția șoferului asupra volanului (fig. 6).

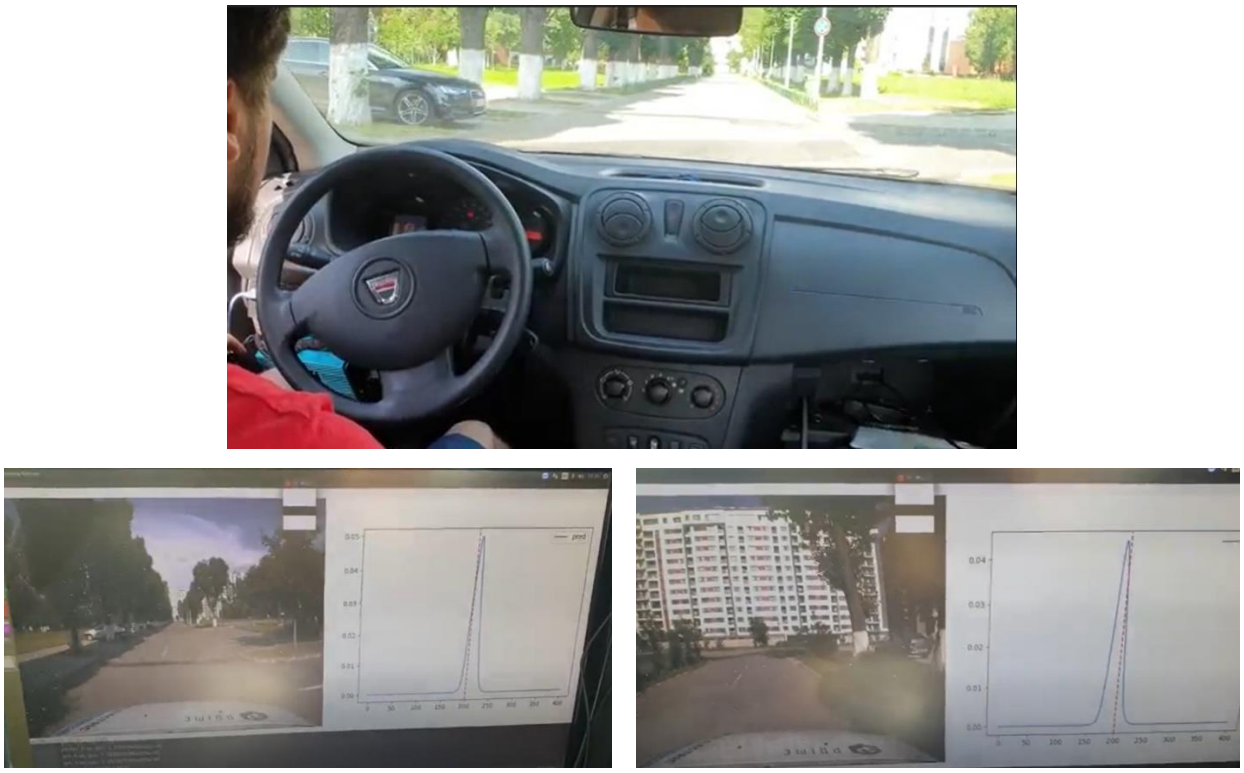

Figura 6. Asistare șofer conducere semi-autonomă. Prima imagine - automobilul menține banda în mod autonom, șoferul nu controlează volanul (șoferul este în auto pentru siguranță și are un comutator de urgență pentru comutarea în mod controlat de fșoer). A 2-a și a 3-a imagine – pe ecran apare în stânga ceea ce vede camera video în timp real iar în dreapta estimarea direcției de mers (cu albastru) față de direcția dorită (cu roșu) pe baza modului antrenat cu rețea neurală în prealabil în simulator și utilizând setul de date UPB augmentat.

Un scurt video de prezentare se poate găsi la adresa:
https://drive.google.com/drive/folders/1puo_naqyv61Rx7jLTeKzeuORE3LPuSww?usp=sharing

De asemenea, partenerul UDJG a propus un ecosistem edge-cloud pentru vehicule autonome, un sistem în care dispozitivele edge (vehiculele) comunică cu servere edge (RSU - road side units), amplasate aproape de vehicule, și servere aflate în alte locații (cloud) ar reduce din presiunea plasată pe vehicule.

4. Proiectul component ROBIN-Context

Parteneri ai proiectului ROBIN-Context

UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI (CO)
UNIVERSITATEA TEHNICĂ DIN CLUJ – NAPOCA

Serviciu web pentru adăugarea de și accesul la informații contextuale ale micro-lumilor

În abordarea bazată pe micro-lumi interacțiunea cu robotul social este încadrată în termenii unui vocabular și al unor funcționalități bine definite. Un exemplu în acest sens îl constituie interacțiunea de asistență/orientare/ghidare într-o clădire inteligentă. Mai exact, un scenariu utilizat este cel de asistență a studenților și vizitatorilor în cadrul clădirii PRECIS, unde au loc activități științifice și educaționale. În cadrul proiectului ROBIN Dialog au fost definite conceptele care constituie micro-lumea interacțiunilor posibile. Acestea fac referire la:

- Evenimente (cursuri, laboratoare, workshop-uri, prezentări științifice)
- Persoane (profesori, asistenți, speakeri)
- Săli (de laborator, de curs)
- Elemente identificând unități de timp și repetivitatea evenimentelor

Folosind aceste elemente se pot compune interacțiuni de exemplu :

Exemplul 1

- Vizitator: Unde se află sala în care se ține laboratorul de robotică?
  - Pepper: Nu știu de niciun laborator de robotică care să se desfășoare astăzi aici.
  - Pepper: Cu cine aveți laboratorul: cu d-na Florea sau cu d-l Georgescu?
    - Vizitator: Nu știu…
      - Pepper: În ce grupă sunteți?
      - Vizitator: În 3C.
      - Pepper: Laboratorul dvs. se desfășoară în sala 412. Vă conduc?
        - Vizitator: Nu, mulțumesc, poți să-mi spui unde e?
        - Pepper: A treia ușă pe dreapta, după lift.

Exemplul 2

- Vizitator: Pepper, am întârziat la laborator!/Laboratorul meu trebuie să înceapă./
  - Pepper: La ce laborator trebuie să participați?
    - Vizitator: Nu știu cum se numește.
    - Pepper: Următoarele laboratoare se desfășoară acum: …
    - Vizitator: A, trebuie să merg la sisteme de operare.
      - Pepper: În sala 123.

În cadrul proiectului ROBIN-Context s-a construit ontologia de domeniu pentru micro-lumea interacțiunilor în clădirea PRECIS, numită PRECIS Schedule Ontology. Conceptele și proprietățile care le leagă sunt prezentate în Figura 7.

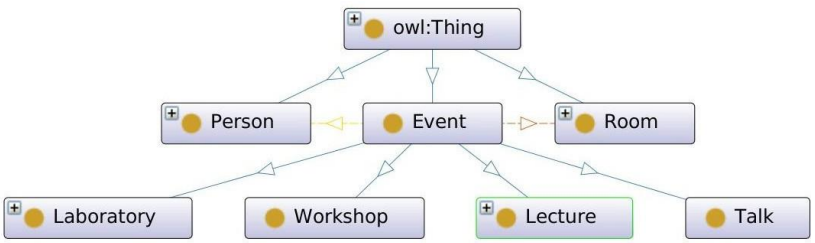

Figura 7. Reprezentare a taxonomiei conceptelor (săgețile albastre) și a proprietăților existente între acestea (event host - săgeata galbenă, event location - săgeata portocalie). Evenimentele se împart mai departe în Laboratoare, Workshop-uri, Cursuri sau Expuneri

Accesarea informației contextuale

Accesul la informația contextuală aferentă micro-lumii de asistență în cadrul clădirii PRECIS se face prin "îmbrăcarea" modelului semantic într-un serviciu web RESTful pentru crearea/interogarea/actualizarea instanțelor de entități și a proprietăților acestora. Dezvoltarea acestui serviciu s-a efectuat folosind tehnologia OBA[6] care permite crearea de specificații de API și template-uri de cod pentru aplicația server și clientcorespunzătoare API-ului pornind de la descrierea ontologică a conceptelor.

Cu alte cuvinte, pornind de la ontologia PRECIS Schedule Ontology, OBA este folosit pentru agenera un API prin care să se facă accesul RESTful la instanțele Evenimente.de Persoane, Săli șiOBA facilitează acest lucru prin conversia unui model semantic definit într-o ontologie, într-o specificație ce utilizează modelul OpenAPI[7]. Fluxul OpenAPI este prezentat in figura 8. Un fișier OWL (conținând ontologia de domeniu contextual PRECIS Schedule) este parsat, facându-se un inventar al entităților (conceptelor) definite în ontologie. Folosindu-se de modelul OpenAPI ontologia este transformată într-o specificație OpenAPI, rezultând un fișier YAML de configurare a fiecărui endpoint REST prin care se face accesul către o entitate a modelului de context (e.g. o instanță de persoană, sală sau eveniment), precum este prezentat în figura 9. În spate, OBA folosește interogări SPARQL pentru a implementa operațiile de create/read/update/delete (CRUD) peste un graf de cunoștințe (knowledge graph) RDF ce conține instanțele definite conform ontologiei de domeniu contextual. În final, prin folosirea specificației OpenAPI create, OBA poate genera cod în diferite limbaje deprogramare (e.g. Python, Java, JavaScript) pentru a implementa interfața client informația de context.

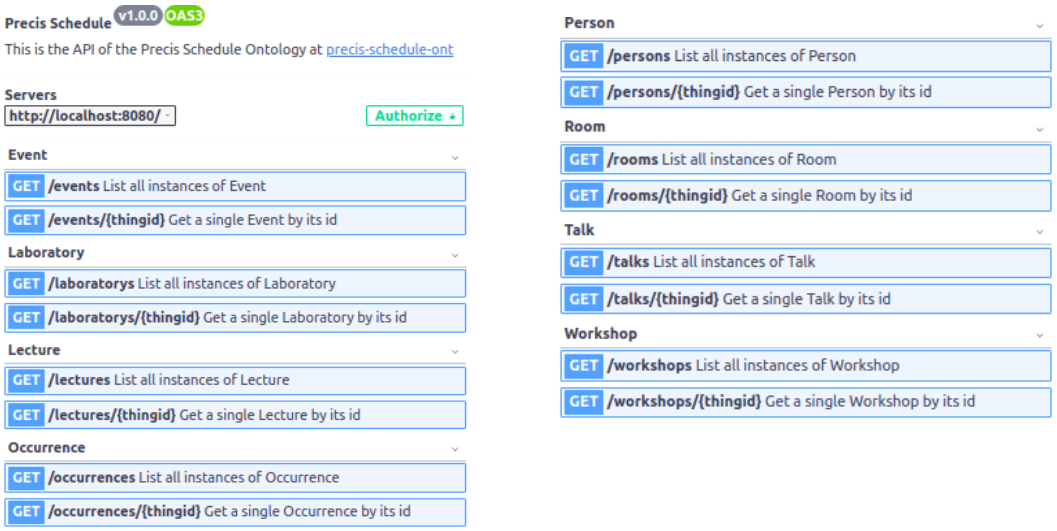

Figura 8. Instanțiere sub forma de specificație OpenAPI a ontologiei de domeniu contextual PRECIS Schedule Ontology

Folosind API-ul pentru informație contextuală rezultat în urma utilizării OBA peste ontologia de domeniu PRECIS Schedule Ontology se poate face cuplarea între managerul de dialog rezultat din ROBIN Dialog și informația de context necesară acesteia. Managerul de dialog pentru micro-lumi creat în proiectul ROBIN-Dialog este capabil să identifice care sunt slot-urile relevante micro-lumii în afirmația sau întrebarea venită de la utilizator. În exemplul 2 utilizatorul asertează că a întârziat la laborator. Modulul ROBIN- Dialog recunoaște că afirmația face referire la un eveniment de tip laborator. Pentru că lipsesc informațiile de natură să identifice despre care laborator este vorba, robotul solicită această informație de la utilizator. În lipsa unui răspuns, managerul de dialog continuăprin listarea tuturor laboratoarelor care rulează acum. Pentru a obține lista acestora, managerul de dialog va folosi interfața client spre API-ul de gestiune a contextului micro-lumii, accesând endpoint-ul aferent evenimentelor de tip laborator. Din lista acestora, va filtra după cele care încadrează ora curentă între

[6] https://oba.readthedocs.io/en/latest/

[7] https://swagger.io/specification/

cele date de proprietățile hasStartime și hasEndTime. Figura 9 ilustrează metoda de integrare între motorul de dialog ROBIN-Dialog și funcționalitatea de slot-filling facilitată de ROBIN-Context.

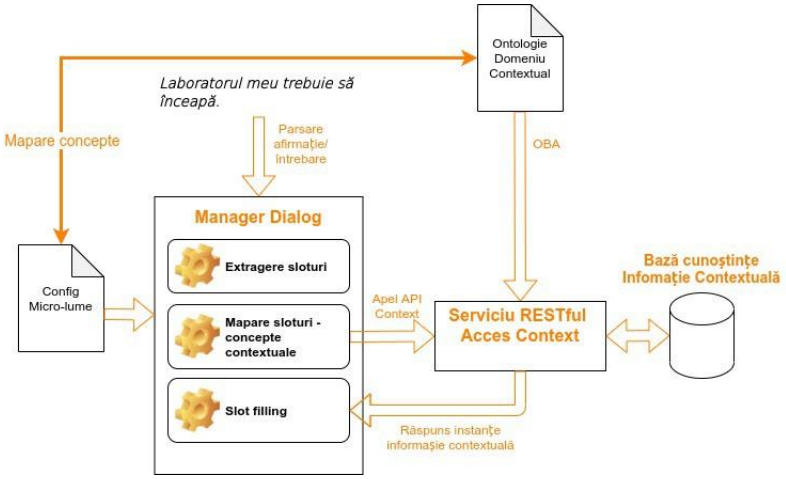

Figura 9. Fluxul de integrare între Managerul de Dialog și Serviciul RESTful de acces al informației contextuale obținut prin folosirea OBA pentru a crea specificația API-ului interfeței client plecând de la ontologia de Domeniu Contextual (PRECIS Schedule Ontology în exemplul dat)

Identificarea punctelor de schimbare în fluxuri de evenimente de la senzori ambientali

In raportul anterior am mentionat ca, in ce priveste adaugarea de rationament probabilistic in platforma ROBIN Context, atentia cade pe procese de clasificare, folosind tehnici de invatare automata, a activitatilor cotidiente (eng. Activities of Daily Living - ADL). Motivul principal este ca acest tip de predictii sunt utile in cadrul proiectului ROBIN Social. In etapa anterioara era subliniat faptul ca detectia activitatilor se face pe baza unor activari de senzori ambientali precum: senzori de miscare, senzori de detectie a inchiderii/deschiderii de usa, senzori de detectie a utilizarii unor robineti sau de folosire a unor obiecte. Astfel, problema principala se rezuma la clasificarea unei serii de activari a acestor senzori intr-una sau mai multe activitatati corespunzatoare acelor activari.

Pentru a efectua clasificarea cu o eficacitate crescută este utilă identificarea punctelor de schimbare (eng. change points) în fluxul de evenimente provenite de la senzori. Aceste puncte de schimbare reflecta faptul ca distribuția din care făceau parte activările anterioare este diferită de cea curentă, indicând astfel posibila trecere de la o activitate la alta. Astfel, în cadrul proiectului ROBIN-Context am realizat metodele de detecție a punctelor de tranziție pentru evenimente de senzori ambientali, utilizând în acest scop seturi de date provenind din proiectul CASAS8. Modalitatea de identificare a punctelor de tranziție între activități propusă în această lucrare este o abordare nesupervizată deoarece se dorește identificarea tranzițiilor cât mai aproape de momentul în care acestea au avut loc. Pentru a realiza acest lucru a fost folosită o abordare bazată pe Kernel-uri Gaussiene pentru a determina similaritatea dintre diferite ferestre consecutive de evenimente.

Au fost considerate primele trei apartamente din setul de date CASAS HH (HH101, HH102 și HH103). Printre activitățile efectuate de către persoaneledin apartament regăsim: Personal Hygiene, Enter Home, Eat, Wash Lunch Dishes, Work, Cook, Groom, Sleep, Bathe, Dress. Orice activare de senzor care nu face parte dintr-o categori anume va fi atribuită categoriei Other Activity. Pentru fiecare set de date au fost calculate metrici de precizie (eng. precision) și regăsire (eng. recall) pentru detectarea corectă a punctelor ce reprezintă o trecere de la o activitate la alta în etichetarea dată în setul de date. De asemenea, a fost luat în considerare un interval de toleranță pentru identificarea punctelor de tranziție. Astfel, 0 - reprezintă potrivirea perfectă, 1 - reprezintă faptul că sunt considerate puncte de tranziție și punctulaflat înaintea unui punct denziție, câtși punctul aflat după acesta și 2 - care respectă regula de la 1, doar că suntconsiderate câte două evenimente. Am calculat precizia, recall-ul și acuratețea la nivel global pentru fiecare set dedate în parte. Pentru aceste statistici s-au eliminat tranzițiile de la Other Actvity către Other Activity deoarece nu erau relevante în contextul actual (o tranziție putea fi identificată corect într-o

[8] D. Cook, A. Crandall, B. Thomas, and N. Krishnan. CASAS: A smart home in a box. IEEE Computer, 46(7):62-69, 2013

secvență de activări de senzori etichetate cu Other Activity, care însă nu au fost etichetate cu activitatea potrivită). Rezultatele obținute sunt trecute în tabelul 4. Se observă că acuratețea și recall-ul au valori destul de ridicate odată cu creșterea intervalului de toleranță, însă precizia este scăzută indiferent de alegerea acestui interval. Deducem astfel că metoda propusă oferă multe puncte care în realitate nu sunt puncte de tranziție, dar identifică corect majoritatea tranzițiilor care chiar au avut loc.

	0			**1**			**2**		
HH	**P**	**R**	**A**	**P**	**R**	**A**	**P**	**R**	**A**
HH101	1	20	80	5	65	81	9	77	82
HH102	1	19	81	4	85	80	8	92	80
HH103	1	24	78	10	76	80	19	86	81

Tabelul 4. Statisticile (P - Precizia, R - Recall-ul si A - Acuratetea) obținute pentru fiecare set de date în parte considerând fiecare interval de toleranță (0, 1 și 2). Valorile au fost calculate exceptând tranzițiile de la Other Activity la Other Activity

În această ultimă etapă din ROBIN-Context s-a dezvoltat serviciul care face integrarea cu dezvoltările din proiectul ROBIN Dialog, s-a continuat dezvoltarea soluțiilor probabilistice pentru detecția activităților pe bază de fluxuri de evenimente provenite de la senzori ambientali prin implementarea și testarea unui algoritm de identificare a punctelor de tranziție, care împreună cu dezvoltările din etapele anterioare au condus la realziarea unei noi tehnologii de prelucrare inetligentă a datelor de context.

5. Proiectul component ROBIN-Dialog

Parteneri ai proiectului ROBIN-Dialog

INSTITUTUL DE CERCETĂRI PENTRU INTELIGENȚĂ ARTIFICIALĂ „MIHAI DRĂGĂNESCU" (CO)
UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI
UNIVERSITATEA TEHNICĂ DIN CLUJ – NAPOCA

Implementarea use case-urilor, oferirea de servicii robotice, validarea și rafinarea soluțiilor, testarea acceptanței și valorificarea ROBIN-Dialog

Managerul de dialog ROBIN-Dialog (prescurtat RDM în cele ce urmează, disponibil în cod open-source la https://github.com/racai-ai/ROBINDialog) implementează trei use case-uri, *la un nivel demonstrativ*:

1. Pepper, ghid de orientare într-o clădire (vezi fișierul de micro-lume precis.mw);
2. Pepper, asistent de vânzări într-un magazin de electronice (vezi sales.mw);
3. Pepper, companion pentru persoanele în vârstă sau cu dizabilități (vezi asistiv.mw).

Serviciile robotice (adică informațiile pentru algoritmul de planificare a robotului) sunt oferite de clasa RDRobotBehaviour, care, împreună cu enumerarea UIntentType, precizează indicatori de comportament pentru robot. Această enumerare trebuie actualizată cu noi tipuri de intenții ale utilizatorului pentru a putea realiza o corespondență dintre un răspuns final al managerului de dialog cu un comportament al robotului. De exemplu, un membru al acestei enumerări este „SAY_SOMETHING” în care indicația pentru Pepper este să verbalizeze răspunsul generat de managerul de dialog. Un alt membru este „TAKE_ME_SOMEWHERE” care instruiește algoritmul de planificare să deplaseze robotul din locul în care se află la poziția pe care managerul de dialog a produs-o. Predicatul cu care managerul de dialog a răspuns întrebării conține această poziție (poate fi o cameră sau sală în care utilizatorul dorește să ajungă).

RDM a fost descris în lucrarea „A Dialog Manager for Micro-Worlds” (Ion et al., 2020) și, de la momentul scrierii până în prezent, a primit următoarele îmbunătățiri:

1. *Referințe ale conceptelor din micro-lume rezolvate prin apeluri către clase Java*. În acest mod, putem răspunde în limba română unor întrebări care necesită un anumit tip de calcul. De

exemplu, la întrebarea „Cât este ceasul?”, RDM trebuie să afle ora curentă din sistem și s-o traducă în fraza corespunzătoare în limba română. Acest calcul este implementat în clasa ro/racai/robin/dialog/generators/TimeNow.java. Pentru a preciza referința substantivului comun „oră” în definiția micro-lumii, vom adăuga liniile

```
REFERENCE oră ro.racai.robin.dialog.generators.TimeNow = G1
TRUE fi G1
```

Clasa ro/racai/robin/dialog/generators/DegreesNow.java ne permite să răspundem întrebării „Câte grade sunt afară?” sau „Ce temperatură e afară?”, utilizând informații despre locul unde se află RDM (după adresa IP) și serviciul web gratuit pus la dispoziție de Agenția Națională de Meteorologie (http://www.meteoromania.ro/).

2. *Abilitatea de a răspunde cu „Da” întrebărilor care sunt specificate complet*, de tipul „Este adevărat că ... ”, „Este astăzi data de ...”, „Se ține cursul X în sala Y?”, etc. Detaliul de implementare este acela că atunci când modulul de analiză a întrebării poate construi un predicat cu argumentele sale care este găsit în baza de cunoștințe, împreună cu argumentele legate, atunci se poate răspunde cu „Da.”.
3. Modulul de „Text-To-Speech (TTS)” a fost înlocuit cu unul mult mai rapid, până când găsim resurse pe care să instalăm modulul TTS descris în lucrare, bazat pe rețele neuronale, care are o intonație mai apropiată de realitate. Am folosit librăria SSLA (Boroș et al., 2018), scrisă în Java, care sintetizează foarte rapid (sub o secundă) fraze de 10-20 de cuvinte.
4. Modulul de „Automatic Speech Recognition” (ASR) a fost înlocuit cu unul mult mai performant (cu o eroare medie de recunoaștere a cuvintelor de trei ori mai mică decât modulul ASR descris în lucrare) și mai rapid, bazat pe rețeaua neuronală complexă Deep Speech 2 (Avram et al., 2020; vezi Secțiunea 3.2.2).

Implementarea sistemului de dialog în limbaj natural pentru micro-lumea unui robot asistiv/teleprezență

Au fost elaborate încă două micro-lumi pentru un robot asistiv în două ipostaze:

1) asistent casnic, pentru care robotul poate susține dialoguri referitoare la interogări/comenzi de genul: aprinde/stinge lumina, crește/scade/setează temperatura în cameră, pornește/oprește muzică/TV, adaugă/întreabă eveniment în calendar. A fost implementat un flux de prelucrare bazat pe RASA (Rasa, 2021) cu un manager de dialog implementat în Prolog. Pentru partea de prelucrare a vocii au fost folosite aplicațiile Google Speech.
2) o persoană cu dizabilități sau în vârstă, pentru care au fost schițate 5 scenarii. Unul dintre aceste scenarii a fost definit formal într-un fișiere de tip „microworld”. Acesta poate fi consultat la adresa src\main\resources\asistiv.mw din GitHub și poate fi utilizat cu managerul de dialog ROBIN-Dialog (RDM pe scurt).

Detecția intenției și a parametrilor acestora - modulul NLU

Am investigat două opțiuni pentru implementarea modulului de detecție a intenției și a parametrilor acesteia: o soluție bazată pe RASA NLU (Rasa, 2021), precum și o soluție proprie, bazată pe arhitecturi de tip Capsule Neural Nets (Zhang et al., 2019). Pentru implementarea ambelor soluții, a fost necesară mai întâi generarea unui set de date care să permită învățarea intenției și a parametrilor asociați acesteia, pentru intențiile considerate de asistentul casnic, în limba română. Am considerat pentru aceasta primele 14 intenții din Tabelul 4 (următoarele 3 au fost adăugate în contextul managerului de dialog). În urma analizei de domeniu, am identificat o serie de provocări ce ar putea apărea în context real într-o asemenea aplicație, și am definit mai multe variante ale setului de antrenare - mai multe scenarii de analiză a performanței. Rezultatele obținute de abordarea bazată pe RASA NLU sunt prezentate în Figura 10. O analiză calitativă a erorilor a indicat faptul că detecția este îngreunată de prezenta sinonimelor/ antonimelor, și a modului în care reprezentările distribuite (eng. „word embeddings”) mapează aceste tipuri de relații: similaritatea dintre sinonime este mai mică decât similaritatea dintre antonime, ceea ce determină o confuzie pronunțată a intențiilor opuse.

Pred/Act	L_ON	L_OFF	L_LO	L_HI	T_SET	T_LO	T_HI	M_ON	M_OFF	M_VOL	TV_ON	TV_OFF	TV_CH
L_ON		2, 3.3	-	-	2	-	-	3,3	-	-	1, 2, 3	-	2, 3.1
L_OFF	1, 2, 3		-	1, 2	3,3	-	2	2	1, 3.1, 3.3	-	2, 3.2	3	2, 3.1
L_LO	2, 3	2, 3.2, 3.3		1, 2, 3	3,1	-	-	-	-	-	-	-	-
L_HI	2, 3.1, 3.3	0, 3	2, 3		3,1	-	-	-	-	3,2	-	-	-
T_SET	1	-	-	-		1, 2, 3.1, 3.2	1, 2, 3	2, 3.1	-	3,3	2, 3.1, 3.2	-	-
T_LO	-	-	-	-	1, 2, 3		1, 2, 3	-	-	-	-	-	-
T_HI	-	-	-	2	1, 2, 3	3.1, 3.2		-	-	1, 3.2	2	-	-
M_ON	-	2, 3.1, 3.2	3,3	-	2, 3.1, 3.3	-	3,2		-	3,3	1	3,2	2, 3
M_OFF	-	1, 3	-	-	-	-	3,3	-		-	-	3.1, 3.2	3,3
M_VOL	-	-	-	-	1, 3.1, 3.3	3,2	3,2	-	-		-	-	3,3
TV_ON	1	-	-	-	2	3,2	-	2	3	3,3		-	1, 2, 3
TV_OFF	1	1, 3	-	-	-	-	-	-	-	1, 3.1, 3.2	-		3
TV_CH	-	-	-	-	-	-	-	-	-	3	-	-	

Figura 10. Rezultatele obținute de modelul RASA NLU – SUPERVISED EMBEDDINGS. Matricea indică scenariul din care încep să apară confuzii între clasele specifice

Prin urmare, am implementat o soluție de preprocesare care să „apropie" sinonimele, și să evidențieze sensul opus al antonimelor (Mrksic et al., 2016; Kim et al., 2016). Tabelul 5 prezintă performanța cantitativă a modelului bazat pe RASA înainte de a efectua această preprocesare, respectiv Tabelul 6 arată performanța modelului în prezența preprocesării.

	0	1	2	3.1	3.2	3.3
Intent F1	0.996	0.589	0.489	0.352	0.472	0.345
Slot F1	0.998	0.885	0.534	0.553	0.501	0.745

Tabelul 5. Performanța modelului RASA NLU fără preprocesare

	0	1	2	3.1	3.2	3.3
Intent F1	0.998	0.708	0.677	0.466	0.585	0.411
Slot F1	0.998	0.885	0.534	0.553	0.501	0.745

Tabelul 6. Performanța modelului RASA NLU cu preprocesare

Se observă o creștere pronunțată a performanței de identificare a intenției corecte, în toate scenariile de analiză. Totodată, Tabelul 7 arată scăderea absolută a confuziilor de tip intenții opuse, în prezența preprocesării.

	0	1	2	3.1	3.2	3.3
FastText	2	67	98	70	66	80
Baseline	1	70	85	59	61	55
Our Solution	1	41	28	17	24	24

Tabelul 7. Numărul de confuzii de intenții opuse, cu diferite modele

Modelul propriu, bazat pe arhitecturi de tip Capsule Neural Networks, a obținut rezultatele prezentate în Tabelul 8. Am comparat performanța cu modelul Wit.ai (Facebook) (Wit.ai, 2021). Pentru aceste modele nu s-a aplicat modulul de preprocesare, dar analiza erorilor a indicat același fenomen de confuzie a intențiilor opuse. În schimb, în urma analizei mecanismului de *self-atention* din aceste modele, s-a observat că - în anumite situații - atenția la detecție nu este, de fapt, pe verb.

	Wit.ai		CapsNetI2S		CapsNetS2I	
Scenario	Intent F1	Slot F1	Intent F1	Slot F1	Intent F1	Slot F1
0	**100**	99.10	99.74	99.30	**100**	**99.88**
1	39.74	**86.97**	47.17	76.88	**54.10**	78.34
2	**38.66**	**76.17**	37.00	49.44	38.33	43.10
3.1	29.48	**73.01**	37.69	42.53	**48.46**	38.25
3.2	29.74	**75.85**	43.33	36.56	**44.10**	33.66
3.3	27.17	**79.18**	**33.41**	49.27	24.94	49.62

Tabelul 8: Comparația performantei modelelor CapsNets cu modelul Wit.ai

Managerul de dialog pentru scenariul de asistent casnic

Un prototip al managerului de dialog a fost implementat în Prolog, în 2 versiuni, exemplificând ultimele 3 intenturi din Tabelul 4 (2 de tip interogare, 1 comandă). Alegerea acestor intenturi a fost făcută întrucât ele poseda o complexitate mai mare în ceea ce privește managerul de dialog.

În ambele versiuni s-a urmărit asigurarea unei varietăți mari de fluxuri alternative de conversație, cum ar fi parametri lipsă și necesitatea solicitării explicite de informație asupra lor, sau furnizarea informației de timp în format relativ și/sau ambiguu, urmat de cereri de clarificare din partea managerului de dialog. Exemplificarea variațiilor poate fi văzută în două videouri disponibile online.

- Dialog Manager Version 1 demo: HomeAssistant_flux_procesare.mp4, cod: disponibil la cerere;
- Dialog Manager Version 2 demo:
  https://www.youtube.com/watch?v=dLzlVVpMKW8&ab_channel=Marcus
  Cod: https://github.com/MarcusGitAccount/home_assistant_ro

Descrierea scenariului pentru asistarea unei persoane vârstnice

Așa cum am arătat deja, pentru a putea susține un dialog coerent și cooperant cu utilizatorul, robotului Pepper trebuie să i se formalizeze conceptele și acțiunile relevante în micro-lumea respectivă. În cele ce urmează este prezentată reprezentarea parțială a micro-lumii asistentului social al unei persoane cu dizabilități sau în vârstă (care poate fi extinsă, în conformitate cu regulile sintactice ale limbajului de definire a micro-lumilor). Dăm aici listingul fișierului `src\main\resources\asistiv.mw` din GitHub. Acest fișier-definiție a unei micro-lumi de robot asistiv dă posibilitatea managerului de dialog RDM să răspundă punctual unor întrebări cum ar fi: „Ce medicament trebuie să iau astăzi?” / „Ce zi este astăzi?” / „Cât e ceasul?” / „Câte grade sunt afară?” / „Trebuie să iau Extraveral astăzi?” / etc.

Implementarea sistemului de dialog în limbaj natural pentru asistența șoferului

Sistemul pentru asistența șoferului poate interacționa atât cu persoanele din mașină prin interacțiune verbală (persoanele din mașină rostesc întrebări iar sistemul răspunde la acestea tot verbal) cât și cu computerul mașinii (trimite comenzi mașinii pentru a fi executate și primește comenzi verbale din partea acesteia – vezi Figura 11).

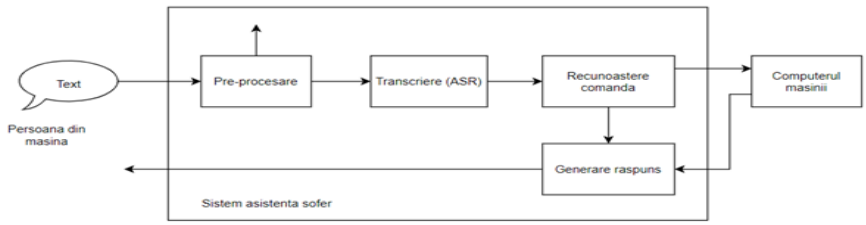

Figura 11. Schema bloc a sistemului de dialog pentru asistența șoferului

Principalele blocuri ale sistemului pentru asistența șoferului sunt:

- **Blocul de preprocesare**: are rolul de a verifica dacă semnalul audio captat este suficient de puternic pentru a putea fi considerat o cerere (evitând astfel activarea sistemului atunci când persoanele din mașină discută între ele, de exemplu). Semnalele audio slabe nu trec de blocul de preprocesare, prelucrarea lor oprindu-se aici. În schimb, semnalele audio puternice trec și sunt filtrate înainte de a fi trimise mai departe. Filtrarea are loc deoarece mediul din mașină poate fi unul zgomotos: discuții pe fundal, muzică pe fundal, etc.
- **Blocul de transcriere**: Același modul de detecție automată a vorbirii (ASR) folosit și pentru ROBIN Dialog.
- **Blocul de recunoaștere a comenzii**: implementat de managerul de dialog ROBIN-Dialog - blocul de recunoaștere identifică ținta comenzii (conceptul) și modul în care se dorește acționat asupra ei (pornire/oprire).
- **Blocul pentru generarea răspunsurilor**: redă unul sau mai multe fișiere audio preînregistrate. Poate fi apelat intern de către blocul de recunoaștere a comenzii sau extern (de către computerul mașinii). În urma primirii unei comenzi, computerul mașinii poate trimite înapoi un răspuns pozitiv („Luminile de avarie au fost pornite!”), un răspuns negativ („Radioul este deja închis.”) sau un răspuns general de eroare (pentru a permite șoferului să verifice problema apărută, de exemplu s-a ars un far). Computerul mașinii poate genera răspunsuri fără să fie nevoie să primească o comandă. De exemplu, dacă modul de vedere artificială identifică pietoni pe partea dreaptă, computerul mașinii poate genera răspunsul compus „pieton”, „pe dreapta”. În mod

similar se pot genera răspunsuri compuse pentru a descrie și alte rezultate ale modului de vedere artificială.

Definitivarea, testarea și validarea modulului configurabil de dialog în limba română

Produs nou dialog în limbaj natural pentru microlumi

Pentru a valida ciclul de dezvoltare a unei micro-lumi, în ce privește acuratețea sistemului ASR, am înregistrat un corpus bimodal specializat pe scenariul asistentului de vânzări într-un magazin de calculatoare. Astfel, pe lângă corpusul de antrenare pe care l-am înregistrat cu interfața specializată, am înregistrat un corpus de test alcătuit din 50 de propoziții în același domeniu al tehnologiilor din industria de profil. Scopul nostru a fost acela de a verifica dacă sistemul ASR care este adaptat acestei micro-lumi, folosind corpusul de antrenare, este mai bun decât sistemul ASR antrenat numai pe corpusul bimodal balansat CoRoLa. Tabelele 9 și 10 prezintă rezultatele acestui experiment.

ASR de referință	**WER (%)**	**CER (%)**
Vorbitor 1	38.71	9.42
Vorbitor 2	59.21	26.28

Tabelul 9: Rezultatele sistemului ASR de referință (antrenare pe CoRoLa) pe corpusul de test

ASR îmbunătățit	**WER (%)**	**CER (%)**
Vorbitor 1	28.37 (-10.34)	9.09 (-0.33)
Vorbitor 2	44.88 (-14.33)	21.83 (-4.45)

Tabelul 10: Rezultatele sistemului ASR îmbunătățit prin antrenarea pe corpusul specializat pe tehnologii de calcul

Se observă imediat că sistemul ASR specializat este mult mai bun decât sistemul ASR general atât la „Word Error Rate” (WER) cât și la „Character Error Rate” (CER). Vorbitorul 1 a înregistrat propoziții în corpusul de antrenare (fiind cunoscut sistemului de ASR) pe când vorbitorul 2 nu a făcut acest lucru și este, deci, necunoscut sistemului de ASR. Această observație explică de ce vorbitorul 1 are rate de eroare WER și CER mai mici decât vorbitorul 2. De asemenea, un efect îmbucurător este că performanța pentru vorbitorul 2 s-a îmbunătățit mai mult decât pentru vorbitorul 1 ceea ce indică faptul că *acest sistem de ASR poate fi adaptat cu succes pentru vorbitori noi*.

Pe lângă validarea adaptării sistemului de ASR într-o micro-lume nouă, am adăugat teste JUnit pentru managerul de dialog ROBIN-Dialog, în fiecare dintre cele trei micro-lumi implementate. Aceste teste se efectuează cu success și sunt disponibile la următoarele link-uri:

- https://github.com/racai-ai/ROBINDialog/blob/master/src/test/java/ro/racai/robin/AsistivTest.java
- https://github.com/racai-ai/ROBINDialog/blob/master/src/test/java/ro/racai/robin/PrecisTest.java

https://github.com/racai-ai/ROBINDialog/blob/master/src/test/java/ro/racai/robin/SalesTest.java

6. Proiectul component ROBIN-Cloud

Parteneri ai proiectului ROBIN-Cloud

UNIVERSITATEA POLITEHNICA DIN BUCUREȘTI (CO)
UNIVERSITATEA TEHNICĂ DIN CLUJ – NAPOCA
UNIVERSITATEA "DUNAREA DE JOS" DIN GALAȚI

S-a definitivat realizarea platforma de planificare a joburilor bazată pe microservicii extrem de scalabilă pentru Cloud Robotics: GIGEL-sched[9]. Entitatea centrală este microserviciul, care este implementat folosind containere Docker. De asemenea, am realizat o platformă de monitorizare web pentru interogarea stării platformei de planificare în timp real pentru a exporta rapoarte. Menționăm câteva contribuțiile realizate: suport pentru infrastructura Cloud utilizată pentru a colecta date de la device-uri IoT, platforma de planificare a joburilor, algoritmi Cloud ML pentru NLP, cercetare și propunere de soluții în contextul mașinilor autonome și roboți colaborativi.

În Figura 1, evidențiem principalele surse de date și componente pentru ROBIN-Cloud. Datele provenite de la sisteme autonome, cum ar fi autoturismele, sunt introduse în ROBIN-Car. Diferite tipuri de roboți sunt conectați la componenta ROBIN-Context. ROBIN-Dialog expune serviciile NLP pentru roboții de asistență. ROBIN-Social presupune că roboții sunt cognitivi pentru a extinde societatea digitală.

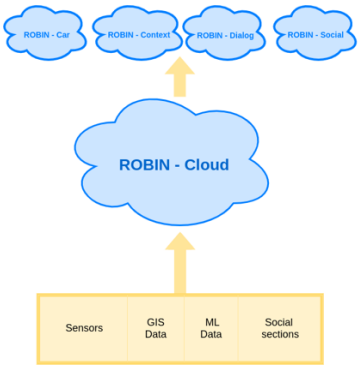

Figura 11. Proiectul ROBIN-Cloud

Într-o lucrare anterioară, am prezentat „My Cloudy Time Machine," un robot de asistență conectat la ROBIN-Cloud. Un utilizator poate face un instantaneu al mediului (de exemplu, flux audio, flux video, date senzori etc.). O capsulă de date[10] este construită cu date colectate și este împinsă în cloud pentru stocare permanentă.

My Cloudy Time Machine

Prezentăm în Figura 12 un model arhitectural pentru dispozitivele conectatela „My Cloudy Time Machine". O structură multi-strat este utilizată pentru a face abstracție peste resursele hardware și software. Entitățile din stratul „Microservices" sunt coordonate de un supervizor. Dacă procesarea locală este aleasă in schimbul procesării în cloud, presupunând că avem mai mulți roboți în rețea, procesele de Supervizon devin echivalente cu nodurile din cloud. Un algoritm descentralizat este folosit pentru a alege un nod de planificare și pentru a planifica procesări locale.

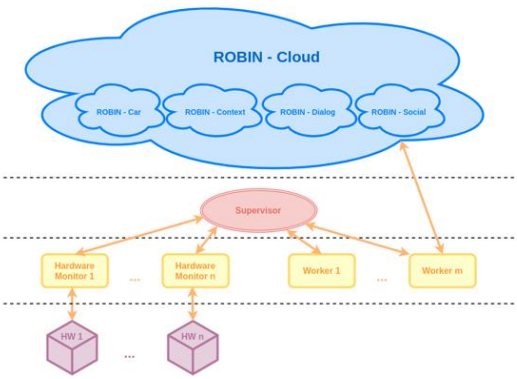

Figura 12. Arhitectura My Cloudy Time Machine

[9] Guided Intelligent GEared Legend - assistive robots used to validate the model

[10] ROBIN-Cloud a definit o abstractizare pentru datele colectate de la device-uri IoT

Platforma scalabila de planificare a task-urilor

Am realizat platforma de planificare a task-urilor GIGEL[11]. În această secțiune, specificăm componentele abstracte utilizate în cadrul GIGEL și modul în care acestea sunt conectate. Vă prezentăm o arhitectură simplificată pentru platforma de planificare în Figura 13.

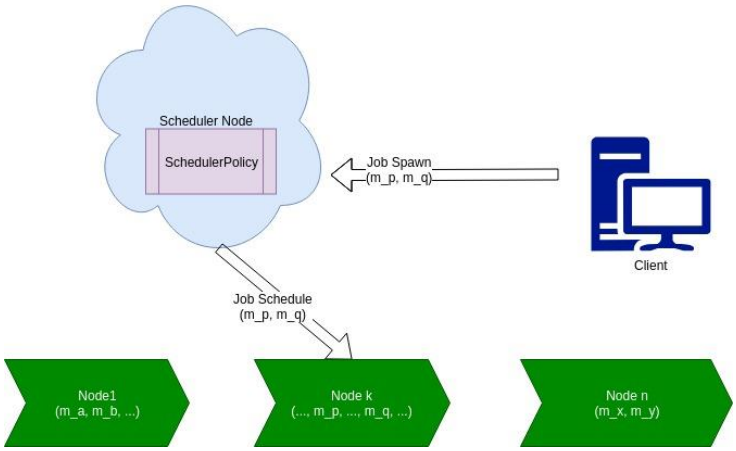

Figura 13. Platforma de planificare

Un client este un dispozitiv utilizator (de ex., PC, server, smartphone) care rulează o aplicație care a integrat API-ul GIGEL pentru trimiterea job-urilor. Orice solicitare a utilizatorului este transformată într-un job care este trimis în Cloud. Când se adaugă un job la coada de așteptare, se adaugă metadatele acestuia (de ex., ID-ul jobului, destinația unde se oferă răspunsul).

Izolarea microserviciilor într-o instanță VM se realizează prin containerizare. Un nod poate avea mai multe microservicii care rulează, fiecare accesibil pe un port diferit. Nodul worker este responsabil pentru gestionarea microserviciilor. O configurație cu microservicii implementate este furnizată la pornirea nodului. Worker-ul monitorizează, repornește și acceseaza microserviciile sale. În acest model, microserviciile dintr-un nod sunt accesibile doar local. Accesul utilizatorului se realizează din executarea job-urilor.

Nodul planificator este capabil să extragă job-uri din coada de așteptare (W), să aplice politica de planificare și, eventual, să introducă lucrări în coada de rulare (R). De asemenea, este responsabil pentru transmiterea efectivă a datelor lucrărilor către nodul țintă. Un comportament definit de implementare este acela că un nod poate fi atât un worker, cât și un planificator. Presupunem că un nod este fie un lucrător, fie un planificator pentru simplitate.

Pot fi utilizate modele personalizate pentru alegerea nodului planificatorului. Cea mai simplă soluție este să ai un nod desemnat anterior. Dezavantajul este că platforma poate deveni inutilizabilă dacă planificatorul se blochează. Un algoritm descentralizat pentru alegerea liderului [12] pentru sistemele distribuite poate fi utilizat pentru a selecta dinamic nodul planificatorului.

Dacă seturile sunt stocate la nivel global și distribuite, sunt implementate metode de acces sincronizate (de exemplu, sincronizare ceas, ceasuri logice, sincronizare blockchain, sincronizare bazată pe token) pentru gestionarea accesului la structurile de date, mai multe noduri de planificare pot fi utilizate în același timp.

Prezentăm procesul de planificare în Figura 14. Planificatorul poate fi asincron sau bazat pe evenimente (de exemplu, atunci când este creat un lot de joburi). În timpul procesului de planificare, elementele din coada de așteptare (W) sunt asociate cu nodurile de la N. Opțional, poate fi specificat un obiectiv (de exemplu, maximizarea randamentului, minimizarea timpului de răspuns, minimizarea consumului de resurse etc.). Un algoritm de planificare este utilizat în etape multiple de la planificare. Planificatorul poate utiliza, de asemenea, o implementare hibridă prin combinarea mai multor politici.

[11] Guided Intelligent GEared Legend - assistive robots used to validate the model

[12] Farhad Soleimanian Gharehchopogh and Hassan Arjang. A survey and taxonomy of leader election algorithms in distributed systems. Indian journal of science and technology, 7:815–830, 2014.

Rezultatul este un plan de planificare care poate fi aplicat direct de planificare sau salvat global (de exemplu, un lucrător primește o notificare și va scoate automat lucrarea asociată din coadă).

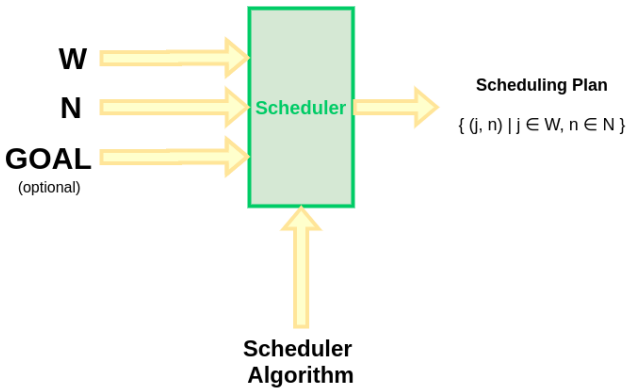

Figura 14. Fluxul de planificare

Hipervizorul este responsabil pentru repornirea VM (nod) în caz de eșec. Toate task-urile neterminate dintr-un nod blocat sunt mutate înapoi în coada de așteptare atunci când planificatorul detectează evenimentul. Un container Docker este pornit pentru fiecare microserviciu care rulează pe un worker. Ca o recuperare de eșec dintr-un container in stare de eroare, un daemon este creat atunci când containerul pornește.

Activitatea de monitorizare colectează date pentru valori cum ar fi încărcarea globală și individuală a procesorului, utilizarea memoriei, numărul de procese, numărul de descriptori de fișiere, utilizarea stocării. Datele sunt trimise într-un backend pentru a fi salvate într-o bază de date SQL. O aplicație Web este conectată la backend pentru a expune statisticii furnizorului Cloud despre platforma de planificare.

Aplicația web GIGEL este, de asemenea, accesibilă clienților cu constrângeri de confidențialitate a locurilor de muncă. Un utilizator poate urmări progresul în timp real pentru joburile trimise. Grafica generată folosind valorile colectate îi ajută pe dezvoltatori să îmbunătățească algoritmii sau euristica utilizată în planificare prin compararea directă a strategiilor disponibile cu aceleași scenarii de testare.

7. Concluzii

Prezentul raport reprezintă descrierea activităților de cerectare și a rezultatelor obținute de consorțiul proiectului în cadrul etapei a 4-a a proiectului ROBIN. Se poate constata faptul că s-au continuat și exploatat cercetările din etapele anterioare în cadrul fiecărui proiect component și s-au stabilit legături între proiecte. În același timp fiecare proiect component a dat naștere la rezultate, publicații, tehnologii originale, utilizabile în diferite alte contexte și aplicații. Este de asemenea de subliniat faptul că cele 5 proiecte componente reprezintă o viziune unitară a conceptului de roboți autonomi, inteligenți, fie că aceștia sunt utilizați pentrua sistenâă socială sau interacțiune cu clienții, fie că sunt utilizați pentru pilotaj automat.

Director de proiect
Prof. Adina Magda Florea