CONSERT

Platformă de management a contextului pentru aplicații de inteligență ambientală

Cod proiect PN-III-P2-2.1-PED-2016-1753
Număr contract 34PED

Etapa II

Validarea CONSERT, in interactiune cu utilizatorii și mediul de business

Cuprins

1. Proiectul CONSERT - Motivație și obiective

Inteligența Ambientală (AmI) este astăzi destul de matură pentru a intra în atenția industriei și a directivelor de susținere la nivel european. Un subdomeniu de bază al AmI este cel al gestiunii contextului și calculul sensibil la context (furnizarea către aplicații și utilizatori a informației potrivite, la momentul potrivit și în modul potrivit). Gestiunea Contextului acoperă multe subiecte de cercetare de la reprezentarea și raționamentul asupra informației și până la arhitecturi de sisteme informatice. O separare a nevoilor și obiectivelor între funcționalitatea de bază a unei aplicații și adaptarea ei contextuală este necesară.

Obiectivul acestui proiect este **crearea unui middleware open source pentru gestiunea contextului** care să furnizeze opțiuni bogate ce înlesnesc dezvoltarea de aplicații sensibile la context. Plecând de la dezvoltări create in cadrul laboratorului AIMAS al UPB, proiectul CONSERT a avut ca obiective: creșterea performanței și modularizarea motorului de inferență pentru a susține aplicații cu diferite nivele de complexitate; crearea unui mediu integrat de dezvoltare (IDE) pentru modelarea contextului spre facilitarea dezvoltării de aplicații contextuale pe baza CONSERT Middleware; dezvoltarea unor opțiuni de implementare (deployment) a unităților de lucru din CONSERT potrivite pentru aplicații din domeniul IoT (folosind OSGi și Docker pentru a aborda schimbarea dinamică a contextului într-o aplicație); dezvoltarea a două aplicații de test: Smart Campus Support pentru monitorizarea activităților studenților și profesorilor într-un campus universitar și Smart Elderly User Support pentru configurarea dinamică adecvată a serviciilor oferite de un mediu de inteligență ambientală care asista utilizatorii in vârsta acasă.

Prezentul raport corespunde etapei de evaluare a middleware CONSERT și la dezvoltarea a două aplicații de integrare și testare a motorului CONSERT în scopul demonstrării capabilităților soluției dezvoltate pentru gestionarea adecvata și dinamică a contextului în aplicațiile de inteligență ambientală. În același timp, s-au realizat o serie de îmbunătățiri ale IDE CONSERT dezvoltat în prima etapă a proiectului.

Secțiunea 2 prezintă evaluarea motorului CONSERT, intr-un mod obiectiv, prin testarea sa pe un set de date folosit pentru recunoașterea activităților zilnice (eng Activities of Daily Living - ADL). In Secțiunea 3 se descrie modul de utilizare a middleware-ului CONSERT pentru Smart Elderly User Support, în particular pentru asistența persoanelor în vârstă prin integrarea middleware CONSERT în soluția de inteligență ambientală CAMI [1] dezvoltată în colaborare de membrii proiectului. Secțiunea 4 concluzionează raportul și prezintă linii de dezvoltări viitoare.

2. Motorul CONSERT – Evaluare pentru o aplicație de detecție a activităților zilnice

Obiectivele inițiale ale proiectului CONSERT prevedeau evaluarea dezvoltărilor din cadrul proiectului pe o aplicație de tip Smart Campus Support, menita monitorizării activităților studenților și profesorilor într-un campus universitar. Cu toate acestea, la momentul elaborării etapelor de evaluare pentru aceasta aplicație, s-a observat ca dezvoltarea unei asemenea aplicații de la zero și testarea acesteia ar costa timp și resurse de infrastructura (senzori in campus și in corpurile clădirilor) care nu au fost disponibile. In plus, o astfel de abordare nu ar fi permis evaluarea intr-un context comparativ (prin raportare la rezultatele obținute de alte soluții pe același tip de aplicație). Ca atare, decizia echipei de proiect a fost aceea de a evalua middleware-ul CONSERT, și motorul de inferența CONSERT in mod particular, pe seama unui set de date existent și foarte cunoscut: CASAS (Center for Advanced Studies in Adaptive Systems)[1] colectat de Washington State University.

In cadrul acestei evaluări, capabilitățile motorului CONSERT (detaliat in raportul Etapei I) sunt folosite pentru a efectua detecția activităților zilnice ale persoanelor in vârsta pe baza activărilor înregistrate de senzori simpli (e.g. de mișcare, de detecție a deschideri/închiderii ușilor). Un beneficiu suplimentar a fost acela ca testarea pe acest set de date a fost informativ in vederea exploatării CONSERT in cadrul soluției CAMI, unde, printre altele, recomandările trimise utilizatorilor sunt declanșate in funcție de activitatea pe care o desfasoara (e.g. mersul la baie de dimineața).

In cele ce urmează este descris setul de date CASAS și modul in care a fost colectat. Apoi este prezentata evaluarea motorului CONSERT pe doua subset-uri de date din cadrul CASAS: ADL Normal și ADL Interwoven.

2.1 Setul de date CASAS și detecția activităților zilnice

CASAS (Center for Advanced Studies in Adaptive Systems) conține diverse seturi de date care au înregistrat activări ale unor senzori instalați in setup-uri de locuințe inteligente in mai multe orașe din lume. Setul de date care a fost folosit in proiectul CONSERT este setul de date înregistrat in orașele Kyoto și Tokyo. Figura 1 prezintă un plan al apartamentului folosit in cadrul experimentelor, precum și amplasarea senzorilor in interiorul apartamentului. Tipurile de senzori folosite sunt următoarele:

- Senzorii de mișcare - devin activi când o persoana trece prin dreptul lor
- senzori de detecție a închiderii/deschiderii unei uși - e.g. dulap cu medicamente, debara
- senzori de detecție a folosirii robinetului de apa sau a aragazului,
- senzori de prezenta a unor obiecte - e.g. oala, pahar, DVD
- senzori de temperatura

[1] http://casas.wsu.edu/datasets/

Având acești senzori instalați, persoanele participante la experimente au fost rugate sa desfășoare o serie de activitati din sfera vieții cotidiene (e.g. a se uita la un DVD, a găti o supa, a scrie o felicitare, a uda florile, a face curat in apartament). Scopul experimentului a fost testarea posibilității de a deduce activitatea efectuata de către utilizatori pe baza activărilor setului de senzori minim intruzivi prezentat anterior.

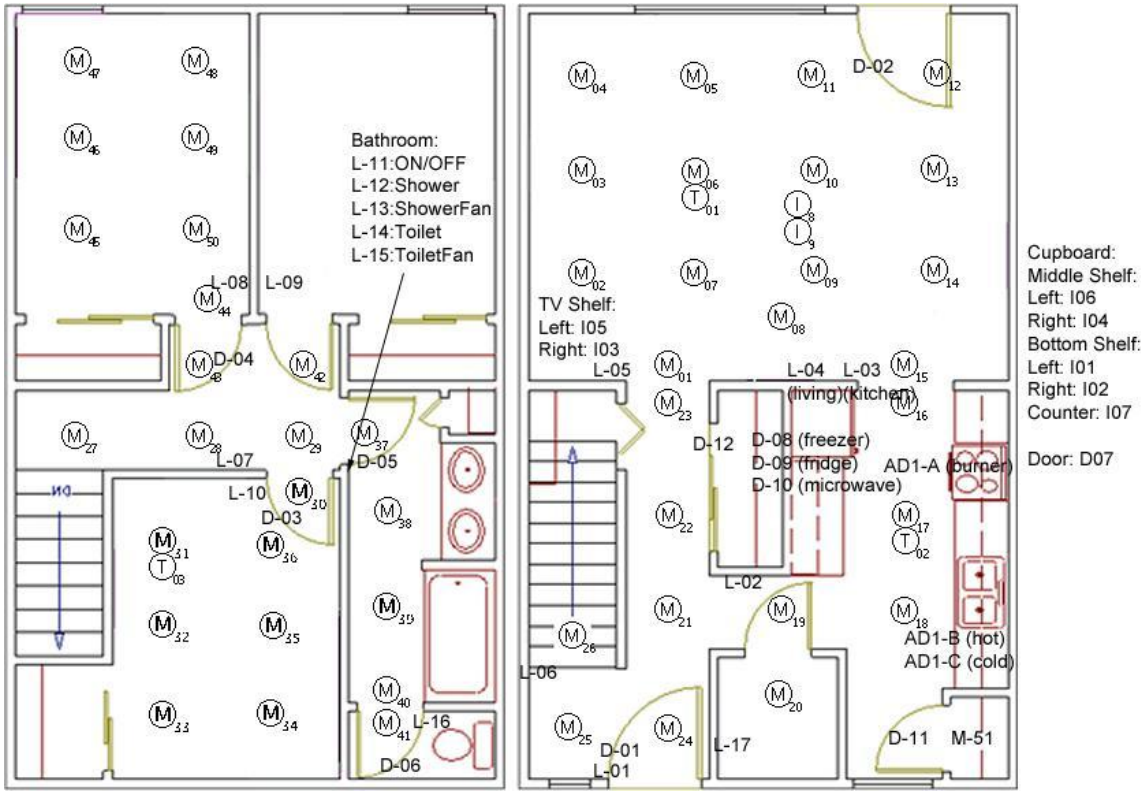

Figura 1. Amplasarea senzorilor in apartamentul inteligent din Kyoto, in cadrul proiectului CASAS[2].

Activările senzorilor au fost așadar adnotate manual, in sensul ca in dreptul fiecărei activări de senzor e trecuta și activitat(ea)/atile corespunzătoare. Pe toata durata experimentelor apartamentul a fost ocupat de o singura persoana, iar in total au participat intre 20 - 25 de persoane la teste, in funcție de tipul acestora.

Experimentele înregistrate au urmat doua scenarii. Intr-o prima instanța persoanele sunt rugate sa execute o secvența predefinita de activitati, fara întreruperea acestora (ADL normal). In al doilea scenariu, activitățile pot fi întrepătrunse - o persoana poate începe sa facă o activitate, apoi o începe pe a doua, o termina pe prima, apoi o termina pe a doua (ADL Interwoven). Aceste scenarii precum și modul lor de evaluare sunt prezentate in secțiunile următoare.

2.2 Evaluarea pe setul de date CASAS ADL Normal

Pentru primul set de date, scopul experimentelor este de a detecta când au loc activitățile, știind ca se executa toate, nu se întrepătrund și sunt parcurse intr-o ordine definita. Activitățile

[2] http://casas.wsu.edu/

propuse sunt - apel telefonic, spălat pe mâini, gătit, mâncat și făcut curat. Numărul de activitati este unul mic, asa ca și rezultatele sunt bune, sistemul detectand aproape perfect activitatile (cf. Tabelul 1). Exista 24 de persoane pentru acest set de date. E important de mentionat ca activitatile implica diverse tipuri de senzori - miscare, utilizare apei, a ustensilelor de gatit, etc, in diverse camere din mediu, cu mici variatii de la persoana la persoana.

Pornind de la acest setup și analizând datele manual, s-a implementat un număr de reguli in CONSERT pentru detectarea activităților. Au fost definite zone funcționale in mediu, a căror importanta spațiala pentru activitati joaca un rol cheie in detectarea lor. De exemplu, daca cineva este in zona mesei cel mai probabil mananca, iar daca e in bucătărie probabil face de mâncare. La fiecare 5 secunde, o regula noua se instantiaza cu locația persoanei. In funcție de locația persoanei și de ceilalți senzori, putem detecta activitățile. Au fost implementate 23 de reguli, 6 pentru locație și restul pentru detecția activităților - cate 2-3 pentru o activitate, pentru a surprinde natura dinamica a acestora (începutul, finalul, etc.). S-a încercat minimizarea numărului de reguli care sa permită modelarea corecta a activităților.

Activitate	Detecții	Delta mediu start (msec)	Delta mediu final (msec)
Curatenie	24/24	-2300	14050
Gătit	24/24	69700	7250
Mâncat	24/24	38700	43200
Apel telefonic	24/24	22100	8050
Spălat mâini	22/24	25350	8250

Tabela 1. Rezultatele evaluării motorului CONSERT pe setul de date CASAS ADL Normal. Sunt prezentate gradul de recall (coloana 1), precum și timpii medii de diferența in detecția începutului și sfârșitului de activitate (coloanele 2 și 3).

Pentru fiecare activitate a fost calculata diferența dintre timpii proprii de detecție și timpii indicați in datele din CASAS, apoi a fost făcuta media acestora pentru a vedea diferența dintre timpii inferați și cei reali (cf. Tabelei 1). Datele oficiale au un oarecare prag de eroare, de exemplu spălatul mâinilor începe uneori inca din sufragerie, de aceea diferențele de timp pot ajunge la nivelul zecilor de secunde in unele cazuri, și de asemenea au fost omise doua activitati, deoarece senzorul de apa pornita lipsește, fara de care nu s-a putut spune cert ca se întâmpla acea activitate (chiar daca persoana respectiva s-a dus in zona chiuvetei, de exemplu). Totuși, in afara de acest caz, celelalte activitati au fost deduse corect.

In privința timpurilor de diferența intre începutul și sfasitul dedus versus cel real, in multe cazuri activitățile sunt marcate ca active (in etichetarea făcuta de utilizator), înainte de a începe

propriu zis, neexistând sloturi in care sa nu fie nici o activitate efectuata. De exemplu tranzițiile intre doua camere (care nu comporta in sine nici o activitate concreta) sunt marcate ca făcând parte dintr-una din cele doua activitati alăturate (e.g. gătit și mâncat). Ca atare o precizie de 100% comparativ cu datele oficiale este greu de obținut, pentru ca de obicei activitățile sunt detectate puțin mai târziu decât sunt ele adnotate oficial, in funcție de când se îndeplinesc anumite condiții din reguli.

2.3 Evaluarea pe setul de date CASAS ADL Interwoven

Pentru setul de date CASAS Interwoven exista mai multe tipuri de activitati (8 activitati), mai mulți senzori, și in plus activitățile pot fi întrepătrunse și executate in orice ordine. Exista câteva persoane care nu au toate activitățile efectuate, dar in general fiecare persoana trebuie sa facă toate cele 8 activitati. Activitățile sunt: - umplerea recipientului de pastile, vizualizarea unui clip de pe un DVD, udat plantele, răspuns la telefon, scrierea unei felicitări pentru o zi de naștere, prepararea supei, curatenie, alegerea imbracamintii.
Exista un număr de 20 de persoane analizate, in general fiecare persoana făcând toate activitățile, cu câteva excepții. Deoarece detecția activităților este acum mult mai grea, fiind necesara clasificarea mai multor instanțe ale aceleași activitati (eventual sparta pe bucati), s-a propus folosirea mai multor metrici pentru a cuantifica rezultatul abordării.

Este important de menționat ca obiectivul acestei evaluări a fost acela de a analiza capacitatea de generalizare a unui sistem de procesare complexa de evenimente (ca cel implementat de motorul CONSERT), de la câteva instanțe particulare la un set întreg de date. Concret, in elaborarea regulilor de detecție, s-a încercat definirea lor pe baza comportamentului a 2-3 persoane, și evaluarea pe restul.

Metoda generica de elaborare a unei reguli a fost aceea de a descrie condițiile *necesare* și *suficiente* pentru detectarea începutului și a sfârșitului acelei activitati, iar apoi de a scrie o serie de condiții necesare pentru a declara ca o instanța a acelei activitati este *inca/din nou* in desfășurare. Au fost calculate mai multe metrici, cum ar fi acuratețe, precizie, rata de intervale care au fost intersectate, senzitivitatea, dar și timpii in care s-au detectat activitati.

Aceste statistici au fost puse intr-un document pentru fiecare persoana in parte, apoi au fost agregate pentru fiecare activitate, și sintetizate in graficul din Figura 2. Rata de suprapunere indica *procentul* de instanțe ale unei activitati care au fost identificate corect de către motorul CONSERT. Precizia arata daca procentul suprapunerii dintre intervalul inferat al unei instanțe de activitate, versus intervalul real, are o valoare mare. Senzitivitatea (eng. recall) arata in schimb cat de mici sunt duratele intervalelor clasificate greșit (e.g. predicția unui tip de activitate atunci când de fapt acest nu era in desfășurare) - cu cat senzitivitatea e mai mare, cu atât intervalele clasificate greșit au o durata mai mica.

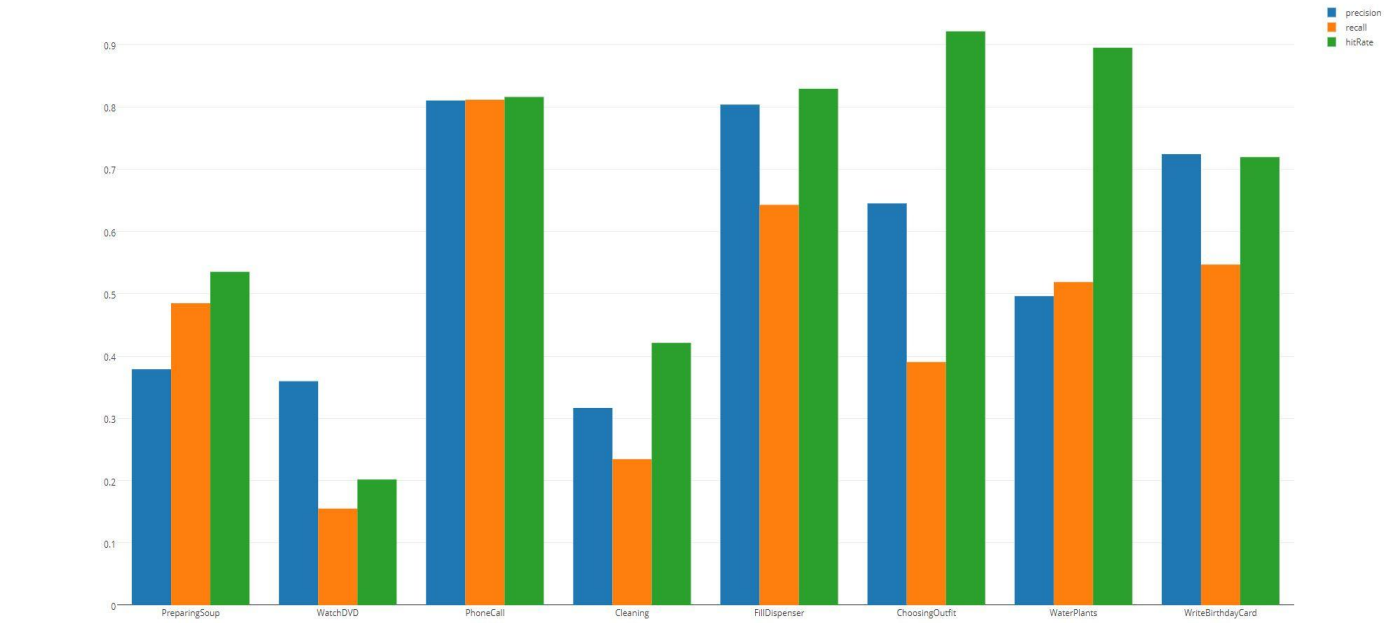

Figura 2. Rezultatele de precizie, senzitivitate și rata de suprapunere agregate per activitate pentru evaluarea efectuata pe setul de date CASAS ADL Interwoven.

In Figura 2 se poate observa ca exceptând activitățile de curatenie și uitat la DVD, activitatea cu prepararea supei are o rata suprapunere de peste 50% iar celelalte au o rata de suprapunere de peste 70%, având și o precizie destul de mare in general. Rezultatele au fost mult influențate de modul in care au fost gândite regulile și de faptul ca e mult mai greu de alcătuit niște reguli care sa tina cont de întreruperi, astfel incat activitățile vor avea durate mai scurte pentru a nu exista multe detecții false.

Modul de imbunatatire a acestor rezultate, tine de efectuarea unor analize statistice apriori asupra datelor pentru a vedea, de exemplu, durata medie a fiecărui tip de activitate, precum și ordinea tipica a desfasurarii lor. Aceste informații pot fi folosite suplimentar in elaborarea regulilor. O alta imbunatatire tine de spargerea activităților principale in subactivitati intermediare, care monitorizează mai îndeaproape fiecare tip de activare de senzor posibila in cazul unei activitati principale. Acestea vor fi explorate in cadrul dezvoltărilor viitoare.

3. Middleware-ul CONSERT – Evaluare în scenarii de asistență a persoanelor în vârsta

Imbatranirea populației este in creștere și cere soluții pentru a ajuta independenta, sănătatea și a unei vieți lipsite de riscuri pentru cei in vârsta și pentru a preveni izolarea lor sociala. Asistarea ambientala a vieții (AAL) folosește informații și tehnologii de comunicare intr-un apartament sau un mediu de lucru pentru a-i face pe cei in vârsta sau pe cei cu nevoi speciale sa ramana activi, conectați la viața sociala și sa trăiască independent utilizând tehnologii și comunicare permanenta și interfețe inteligente cu utilizatorul.

CAMI este un sistem inteligent artificială pentru auto management și o calitate susținuta a vieții in asistarea ambientala a vieții, ce are o arhitectura modulara in cloud ce integrează caracteristici AAL puternice, cum ar fi - monitorizarea sănătății de acasă, exerciții fizice supervizate, detecția căderii, și o interfața multimodala ce asigura un acces ușor la functionalitatile diferite ale sistemului. Conceptul general și serviciile oferite sunt prezentate in Figura 3.

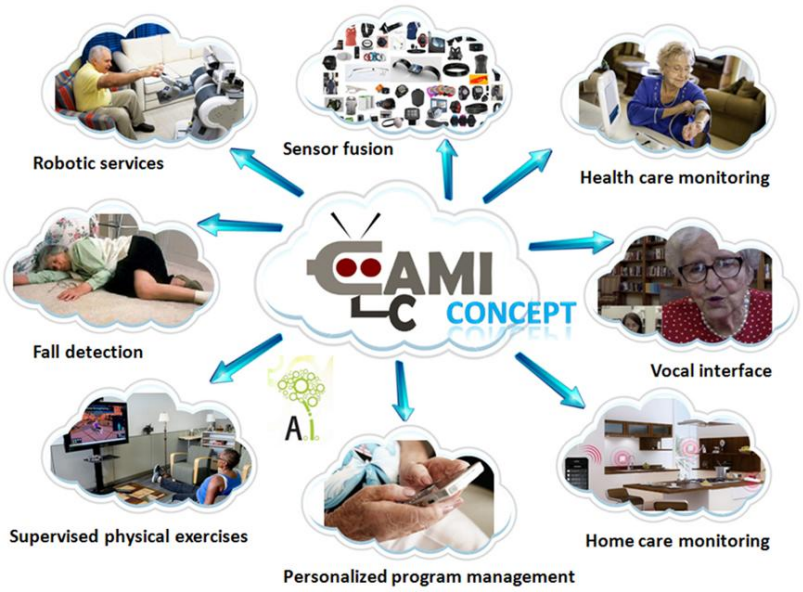

Figura 3. CAMI: concept și servicii oferite

Conceptul și functionalitatile sunt realizate printr-o arhitectura flexibila și modulara, integrând diverse tehnologii, platforme, și tehnici de inteligenta artificiala. Aceste functionalitati includ monitorizarea parametrilor de sănătate, monitorizarea mediului, stimularea activității fizica a individului, detecția căderii, planificarea activității zilnice, emiterea de sfaturi și aduceri aminte, comunicarea cu îngrijitorii și interacțiunea cu sistemul printr-o platforma robotica.

3.1 Arhitectura sistemului CAMI de suport pentru persoanele în vârstă

Soluția CAMI este realizată ca un sistem de tip micro-serviciu multi-modal, unde componentele individuale și serviciile pot decreta propriul lor ciclu de viața, in același timp răspunzând la evenimente și cereri de servicii de la alte module. Figura 4 prezintă o diagrama bloc ilustrând separarea sistemului CAMI in componentele sale principale logice - Unitatea de Sensori, Portalul CAMI, CAMI Cloud și CAMI Interfata cu Utilizatorul Multi Modala.

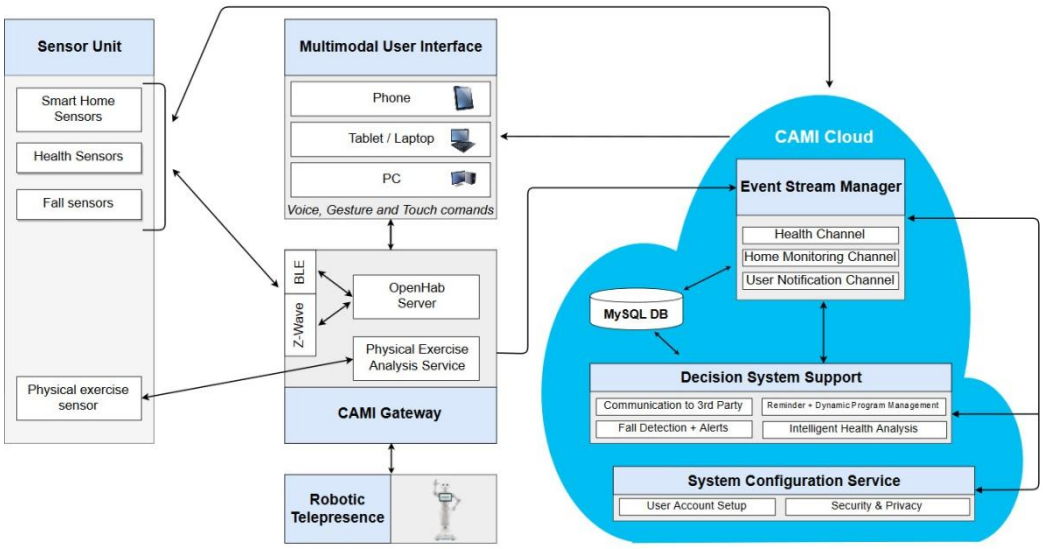

Figura 4. Diagrama bloc a sistemului CAMI

CAMI Cloud e compus dintr-un set de micro servicii ce definesc funcționalitatea sistemului CAMI. Pe langa infrastructura (de exemplu, Managerul fluxului de evenimente) ce permite micro serviciilor sa interacționeze cu celelalte, CAMI Cloud găzduiește servicii ce permit configurarea conturilor utilizatorilor și o analiza a masuratorilor și evenimentelor primite de portalul CAMI. Serviciul de suport pentru decizii (DSS) este responsabil de a emite notificări sau aduceri aminte pentru utilizator, având in vedere informația observata. E de asemenea serviciul in care middleware-ul CONSERT este folosit ca sa implementeze funcționalitatea.

3.2 Utilizarea CONSERT Middleware in arhitectura CAMI

Sistemul de suport pentru decizii CAMI oferă funcționalitatea de a se ocupa cu comunicarea cu dispozitive terțe, cum ar fi sisteme de monitorizare a sanatatii, analiza sanatatii, managerul de notificări, managementul dinamic al activității saptamanale, ca și al alertelor in cazul in care se detectează căderea. Ne vom concentra pe doua aspecte, și anume analiza sanatatii și managementul notificărilor și al memento-urilor.

3.2.1 Analiza parametrilor de sănătate

CAMI monitorizează continuu parametrii de sănătate cum ar fi greutatea, presiunea arteriala, pulsul, numărul de pași și somnul. Depinzând de nevoile persoanelor in vârsta, senzori care masoara nivelul de oxigen al sângelui sau nivelul glucozei pot fi integrați ușor in sistem.

Unul din taskurile importante ale CAMI este sa emită notificări periodice atât persoanelor in vârsta, cat și purtătorilor de grija, când parametrii monitorizați deviază substanțial de la norma uzuala.

Sistemul de suport pentru decizii CAMI va trimite alerte cu notificări in următoarele situații:

- Valorile tensiunii sistolice sau diastolice sunt mult sub sau peste normele uzuale (de exemplu, tensiune peste 15/9 sau sub 10/5)
- Valorile pulsului sunt peste parametrii normali (de exemplu, puls peste 100), deși persoana in cauza nu face exerciții
- Valorile pulsului in timpul zilei sunt foarte scăzute, deși utilizatorul nu doarme (puls sub 50, de exemplu)
- Utilizatorul a făcut mai puțin de 6000 de pași de la începutul zilei
- Utilizatorul are 2 cântăriri zilnice consecutive care diferă cu peste 2kg

3.2.2 Managementul notificărilor și al aducerilor aminte

S-au menționat deja câteva cazuri in care utilizatorul primește notificări bazate pe devierea masuratorilor tipice ale sanatatii. Pe langa aceste evenimente, sistemul CAMI trimite aduceri aminte pentru consumul planificat de medicamente sau pentru măsurătorile obligatorii de sănătate. Totuși, aceste masuratori nu sunt definite pe baza unui moment al zilei fix, dar mai degrabă relativ la alte activitati (trezit, mâncat). De exemplu, intr-un scenariu considerat, sistemul determina ca utilizatorul s-a trezit, bazat pe activarea senzorilor de mișcare aflați in baie. Sistemul așteaptă 10 minte pentru ca monitorizarea obligatorie a greutății și presiunii arteriale sa aibă loc. Altfel spus, CAMI considera ca utilizatorul isi va aminti singur sa facă aceste masuratori, fara sa trebuiască sa ii fie amintit exact după activarea senzorilor. Daca nu a fost efectuata nici o masuratoare in 10 minute, CAMI trimite o aducere aminte pentru măsurătorile necesare și așteaptă primirea valorilor pentru alte 10 minute.

Daca măsurătorile nu sunt primite nici a doua oara, o notificare va fi trimisa purtătorilor de grija. Orice alta masuratoare după acest timp va fi înregistrata, dar un și validata. Pentru acest scenariu simplu, devine evident ca managerul de notificări trebuie sa raționeze pentru ordinea și distanta in timp a evenimentelor, de altfel și pentru lipsa evenimentelor pentru o perioada de timp.
Sistemul trebuie in plus sa raționeze peste datele colectate din diverse surse, cum ar fi dispozitivele de măsurare a sanatatii și senzorii de monitorizare a mediului, ca sa punem situațiile in context.
Aceste tipuri de raționament și funcționalitate sunt oferite de către CONSERT.

3.2.3 Modelarea informațiilor de context în CAMI

S-a menționat faptul că Managerul fluxului de activitati e implementat ca o instanță de tip RabbitMQ, in care toate măsurătorile, monitorizarea de acasă și eventuri de tip feedback al utilizatorului (reamintiri sau instiintari) sunt inserate. Pentru început, aserțiunile relevante de tip ContextAssertions și ContextEntities sunt modelate folosind meta-modelul CONSERT. Tabelele 2 și 3 arata un exemplu de aserțiuni și entitati modelate, și de asemenea descrierea entităților asociate.

ContextAssertion	**Roles (ContextEntities)**	**ContextAnnotations**
bp_measurement	Person BPValues	Timestamp Source device Source gateway
weight_measurement	Person Numeric literal (weight value)	Timestamp Source device URI Source gateway URI
motion	String literal (room name)	Timestamp Source sensor URI Source gateway URI
exercise_started	Person String literal (exercise name)	Timestamp
exercise_ended	Person String literal (exercise name)	Timestamp
send_notification	Person Notification	Timestamp
notification_ack	Person Notification	Timestamp

Tabela 2. Exemplu de ContextAssertions folosite in CAMI. Entitatile care joaca un rol in asertiune și adnotarile necesare sunt indicate.

ContextEntity	**EntityDescription**	
Person	hasID hasName hasProfileLang hasTimezone	String literal (URI) String literal String literal (enumeration) String literal
Notification	hasID hasTitle hasContent	String literal (URI) String literal String literal

Tabela 3. Lista cu exemple de ContextEntities ce au descrieri aditionale.

3.2.4 Implementarea Middleware-ului CONSERT în CAMI DSS

In sistemul de suport pentru decizii CAMI folosim o singura unitate de management, ce are următoarea compoziție:

- 2 agenți CtxSensor: unul care are grija de informațiile legate de măsurătorile de sănătate, și unul care are grija de monitorizarea casei și informațiilor interacțiunii cu utilizatorul
- Un CtxCoordinator și un CtxQueryHandler ce au grija de o instanța de CONSERT Engine
- O instanța de CtxUser ce subscrie pentru toate instanțele create de notificări/ aduceri aminte pe care sistemul trebuie sa le trimită utilizatorului.

Figura 5 arata cum Middleware-ul CONSERT e instantiat in sistemul de decizii CAMI. Cei doi menționați agenti CtxSensor sunt cuplați prin adaptoare la canalele corespondente in managerul fluxului de evenimente. Adaptoarele implementează un client de RabbitMQ ce subscrie la coada necesara evenimentului (de exemplu Adaptorul pentru masuratori de sănătate subscrie la canalul de sănătate). Adaptorul primește mesajul cu masuratoare, formatat in concordanta cu API-ul de inserție din CAMI, și se translatează in asertii și entitati sub forma arătata in tabelele de mai sus.

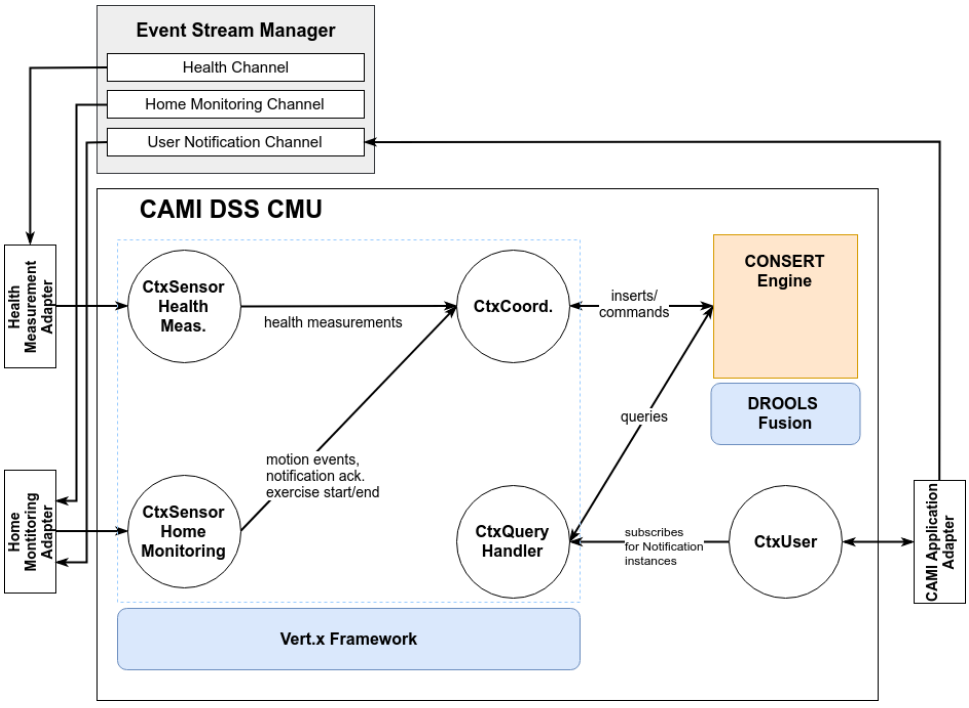

Figura 5. Instantierea Middleware-ului CONSERT in serviciul de suport pentru decizii din sistemul CAMI

In scenariile prezentate la început, am explicat ca datele de ieșire ale analizei de sănătate și al managerului de notificări sunt, in sine, notificări (de exemplu alerte pentru citiri anormale de sănătate, reamintiri pentru a luat măsurătorile de sănătate). De aceea Adapatorul de aplicație CAMI ce e legat de instanța de CtxUser instruiește agentul sa subscrie pentru instanțe ale unor

aserțiuni derivate de engine-ul CONSERT. Pentru fiecare din aceste aserțiuni, adaptorul le convertește intr-un JSON și îl publica pe canalul de notificări al utilizatorului din managerul fluxului de evenimente. De aici e preluat de un CAMI Cloud Service ce trimite notificări la telefoanele utilizatorilor prin PushBots API.

3.2.5 Deducția informației de context

Deducția informației de context pornește de la descrierea tipului de reguli ce sunt utilizate pentru analiza masuratorilor medicale și gestiunea notificărilor. Inferențele in Middleware-ul CONSERT sunt efectuate de motorul CONSERT, beneficiind de dezvoltările aduse acestuia din urma, descriere in raportul pentru etapa I a proiectului.

In cele ce urmează este descris un subset din regulile de procesare complexa a evenimentelor care implementează funcționalitatea prezentata in scenariile de la începutul acestei secțiuni.

S-a menționat ca in sistemul CAMI pot fi captate evenimente declanșate de senzori de mișcare, precum Fibaro Motion Sensor[3]. Cu toate acestea, in scenariul prezentat se doreste trimiterea de notificari pentru efectuarea unei masuratori de greutate dimineata, atunci cand utilizatorul este detectat ca fiind in baie. Se doreste ca regula sa fie activata atunci cand sistemul este sigur ca utilizatorul este in baie și ca a ramas acolo pentru o perioada de timp.

```
rule "Remain in the Bathroom"
when
        $loc : PersonLocation(p: person, room : loc == "Bathroom",
                locAnn : annotations)
        not( exists PersonLocation(person == p, loc != room,
                this annOverlappedBy[0s, 5s] $loc || $loc annIncludes this))
        not( exists Motion(status == "ON", this annHappensAfter[0s, 5s] $loc))
then
        long ts = eventTracker.getCurrentTime();
        DefaultAnnotationData ann = new DefaultAnnotationData(ts);
        PersonLocation sameLoc = new PersonLocation("Bathroom", ann);
        eventTracker.insertAtomicEvent(sameLoc);
end
```

Figura 6. Regula de inferenta contextuala care se activeaza atunci cand o persoana se afla in baie și ramane acolo.

In Figura 6 este afișata o regula care detectează momentul in care o persoana ramane intr-o anumita camera. Regula spune ca daca o persoana se afla in bucătărie și nu exista nici o suprapunere cu o alta locație pentru aceeași persoana și niciun alt senzor de mișcare nu este declanșat in următoarele 5 secunde, atunci durata validității pentru locația curenta a persoanei poate fi extinsa.

[3] https://www.fibaro.com/en/products/motion-sensor/

De observat este folosirea operatorului *not*, care in acest caz permite exprimarea unei condiții privind absenta altor evenimente intr-o perioada de timp, înainte sau după un eveniment existent.

```
rule "Early Morning Weight Measurement"
when
        $loc : PersonLocation(p: person, room : loc == "Bathroom",
                locAnn : annotations, locAnn.startTime > 6AM, locAnn.endTime <
11AM,
                locAnn.duration >= 1m)
        not( exists PersonLocation(person == p, loc == room,
                this annHappensBefore[0s, 5h] $loc))
        not( exists WeightMeasurement(person == p, this annHappensAfter[0s,
10m] $loc)
                || $loc annIncludes this)
        not( exists SendNotification(person == p, notif : notification,
                notif.type == “weight_measurement”)))
then
        // send the reminder for early morning weight measurement
        long ts = eventTracker.getCurrentTime();
        DefaultAnnotationData ann = new DefaultAnnotationData(ts);
        Notification n = new Notification(p, “weight_measurement”);
        SendNotification sendNotif = new SendNotification(p, n, ann);
        eventTracker.insertAtomicEvent(sendNotif);
end
```

Figura 7. Regula pentru trimiterea unei notificari de efectuare a unei masuratori de greutate dimineata, ca urmare a primei vizite la baie.

Daca se poate determina când o persoana se afla in baie, regula din Figura 7 arata condițiile in care este trimisa notificarea de aducere aminte pentru efectuarea unei masuratori de greutate dimineața. Primele doua condiții de tip *PersonLocation* asigura ca este prima oara in dimineața curenta in care utilizatorul intra in baie, un semn ca acesta s-a trezit. Următoarea condiție așteaptă după o masuratoare de greutate preț de 10 minute, considerând ca utilizatorul isi poate aduce aminte singur de acest lucru. Ultima condiție verifica daca sistemul nu a trimis deja o astfel de notificare.

In cele din urma, Figura 8 arata modul in care motorul CONSERT verifica după descreșteri anormale intre doua masuratori consecutive ale greutății.

```
rule "Abnormal Weight Decrease"
when
        $meas1 : WeightMeasurement(p: person, v1 : value, ann1 : annotations)
        $meas2 : WeightMeasurement(person == p, v2 : value, v2 - v1 > 2,
                this annHappensBefore $meas1))
        not( exists WeightMeasurement(person == p, this annHappensAfter
$meas2,
                this annHappensBefore $meas1))
then
        // send a notification for abnormal weight decrease
        long ts = eventTracker.getCurrentTime();
        DefaultAnnotationData ann = new DefaultAnnotationData(ts);
        Notification n = new Notification(p, “weight_decrease”);
        SendNotification sendNotif = new SendNotification(p, n, ann);
        eventTracker.insertAtomicEvent(sendNotif);
end
```

Figura 8. Regula pentru trimiterea unei notificari in urma unei descresteri anormale a greutatii intre doua masuratori consecutive (in dimineti diferite).

Primele doua condiții din Figura 8 identifica masuratorile de greutate pentru o persoana, in timp ce a treia asigura ca cele doua sunt consecutive (i.e. nu exista nici o alta masuratoare de greutate intre ele).

Beneficiul general dat de utilizarea unei instanțe a middleware-ului CONSERT pentru gestiunea unei parți a responsabilităților serviciului CAMI DSS este nivelul de control pe care CONSERT îl aduce in cadrul procesului de dezvoltare și mentenanta a sistemului. Meta-modelul CONSERT se potrivește bine raportat la nevoile de reprezentare existente in informația gestionata de CAMI DSS, iar natura declarativa a regulilor din motorul CONSERT asigura un bun nivel de înțelegere (comprehensibilitate) pentru funcționalitatea dorita a sistemului. In acest fel, procesul de debugging și repararea a erorilor este ușurat.

4. Concluzii

În cea de-a doua etapa a proiectului s-a realizat evaluarea middleware-ului CONSERT prin integrarea motorului îmbunătățit CONSERT în două aplicații și creșterea performanțelor IDE. Evaluarea s-a efectuat prin intermediul a doua scenarii. Primul a privit testarea capabilităților de modelare și inferența ale motorului CONSERT pe seama setului de date CASAS, care presupune detecția activităților umane zilnice, utilizând activări de senzori minim intruzivi. Evaluarea pe acest set de date a urmărit îndeosebi validarea functionalitatii și a facilităților motorului CONSERT dezvoltate in prima etapa de proiect.

Rezultatele arata ca posibilitatea folosirii de operatori temporali, precum și cea a unor operatori de negație (i.e. condiții asupra lipsei unor evenimente) ajuta mult in flexibilitatea și puterea de deducție a unui sistem de procesare a evenimentelor. Procesul de extensie implicita a validității temporale a aserțiunilor contextuale, prezentat in raportul primei etape, este cel care permite transformarea evenimentelor atomice din setul de date CASAS, in situații concrete cu durata de timp, pentru care pot fi exprimate condiții relevante.

Rezultatele obținute pe setul de date CASAS Interwoven arata ca exista loc de imbunatatire ale acurateței detecției de activitati, dar metodele de imbunatatire in acest caz sunt clare și vor fi subiectul dezvoltărilor din viitorul apropiat.

Cel de-al doilea scenariu a privit evaluarea unui deployment întreg al unei instanțe din CONSERT Middleware (agenți + motor de inferența) in cadrul unui sistemului CAMI dezvoltat de membrii proiectului în colaborare în cadrul unui consorțiu cu parteneri europeni. Existența unui set de date extins colectat de partenerii din cadrul consorțiului a permis testarea direct pe date provenite de la utilizatori. In plus, deoarece soluția CAMI este gândită să evolueze spre un produs comercial, s-a putut face evaluarea în cadrul unui sistem care ține cont de cerințele mediului de business. Descrierea din Secțiunea 3 arata ca middleware-ul CONSERT este o soluție foarte potrivita pentru cerințele funcționale ale sistemului de suport al deciziilor (DSS) din cadrul CAMI. Mai mult decât atât, funcționalitatea din CAMI, la care middleware-ul CONSERT a contribuit, a fost supusa și unui studiu utilizator a cărui detalii pot fi analizate in [2].

Dezvoltările ulterioare încheierii proiectului CONSERT privesc următoarele:

- Imbunatatirea evaluării motorului CONSERT pe setul de date CASAS și evaluarea pe alte seturi de date din domeniul Inteligentei Ambientale și a detecției de activitati
- Construirea unei infrastructuri de senzori minim intruzivi in laboratorul AI-MAS și testarea unei implementări distribuite a CONSERT in cadrul acestuia
- Utilizarea middleware-ului CONSERT in scenarii de asistenta robotica, prin asigurarea capabilităților de inferența contextuala ale unui robot umanoid, folosit in aplicații de asistenta personala sau in cadrul unor expoziții

Referințe

[1] Ashalatha Kunnappilly, Alexandru Sorici, Imad Alex Awada, Irina Mocanu, Cristina Seceleanu, and Adina Madga Florea. "A Novel Integrated Architecture for Ambient Assisted Living Systems." In *IEEE 41st Annual Computer Software and Applications Conference* (COMPSAC), pp. 465-472. IEEE, 2017.

[2] Imad Alex Awada, Oana Cramariuc, Irina Mocanu, Cristina Seceleanu, Ashalatha Kunnappilly and Adina Magda Florea, “An End-User Perspective on the CAMI Ambient and Assisted Living Project”, In the *12th Annual International Technology, Education and Development Conference* (INTED), Valencia, Spain, pp. 6776-6785, DOI: 10.21125/INTED.2018.1596, ISSN: 2340-1079, 2018.

[3] Mihai Trăscău, Alexandru Sorici, Adina Florea. “Detecting Activities of Daily Living Using the CONSERT Engine”. In 9th International Symposium on Ambient Intelligence (ISAmI) 20-22 iunie 2018, Toledo, Spain. In curs de publicare in *Advances in Intelligent Systems and Computing Series*, Springer

[4] Imad Alex Awada, Irina Mocanu, Alexandru Sorici, Adina Florea. “An Integrated System for Improved Assitsted Living of Elderly People”. Book chapter in *Recent Advances in Intelligent Assistive Technologies: Paradigms and Applications*, eds. H.N. Costin, B. Schuller, A.M. Florea, Springer, (publicare in 2019).

[5] CONSERT Middleware https://github.com/ami-lab/CONSERT-Middleware/

[6] CONSERT IDE https://github.com/ami-lab/CONSERT-IDE/