

Raport cercetare

Specificațiile unui sistem fondat pe grafuri de context în medii dinamice de Inteligență Ambientală

Adrian Marius Dobrescu, anul 1, ISI

Coordonator – Șl. Dr. Ing. Andrei Olaru

Abstract:

În Inteligența Ambientală, orice scenariu trebuie raportat la agenți deoarece ei sunt entitățile care fac posibilă realizarea unui mediu electronic permanent activ dar transparent pentru om. Un agent conține informații despre mediu, în funcție de care poate lua decizii și efectua acțiuni. Contextul, pe baza căruia agenții funcționează, în acest studiu, este reprezentat ca un graf orientat. Astfel, vom ridica problema gestionării grafurilor și interacțiunea cu alți agenți, prin schimbarea de informație reprezentată ca grafuri, într-un scenariu în timp real.

1. Introducere:

Inteligența ambientală – sau Aml (Ambient Intelligence) este un domeniu care prezintă mult interes în multe ramuri ale direcțiilor de cercetare actuale, agregând capacitățile și beneficiile care pot fi introduse în activitățile zilnice ale oamenilor. Proiectele de inteligență ambientală folosesc foarte des noțiunea de “context”, deoarece o situație în care un sistem trebuie să ia o decizie pentru a face o acțiune este definită de un context. O bună reprezentare pentru descrierea contextului și pe care o vom încorpora în acest studiu, este un graf orientat.

În prezent, sistemele multi-agent întâlnite în diverse scenarii de Inteligență Ambientală sunt particulare mediului respectiv, fiind greu de adaptat și reconfigurat pentru situații diferite. Problema care se încearcă a fi rezolvată este aceea de a construi un sistem de agenți, adaptabil, care să se poată reconfigura la schimbări ale mediului și ale contextului scenariilor, care intervin de-a lungul timpului. O soluție pentru construirea unui astfel de sistem trebuie să aibă în vedere mai multe aspecte. Reprezentarea contextului, suficient de ușor manipulabilă de agenți cognitivi, care pot folosi extensiv informațiile pentru construirea de decizii. De asemenea, trebuie avută în vedere comunicarea între agenți, modul cum informația este diseminată de către grupurile de agenți. Rezolvarea validității temporale a informației este un aspect important, la fel ca și identificarea priorităților pentru agenți.

Pentru definirea unui scenariu, construit pe baza agenților au fost ilustrate numeroase definiții și arhitecturi, care au încercat să fie cât mai generale pentru tipuri diverse de situații. În Inteligența Ambientală, un sistem construit pe baza agenților, este fondat în cea mai mare parte

pe agenți reactivi (senzori, cronometre, afișaje), cu unele centre cognitive care sunt capabile de a efectua acțiuni asupra mediului în diverse condiții. În “Minsky’s Society of the Mind Theory” [1] este o ipoteză care sugerează construirea unui sistem inteligent ca o societate de agenți fără raționament dar capabili să interacționeze între ei, fiecare având competențele sale specifice. De exemplu, o societate de agenți care este capabilă să construiască un turn, va încorpora agenți pentru identificarea materialelor, ridicarea și mutarea lor[2]. Ideea constă în cooperarea locală a agenților astfel încât societatea ca un întreg să funcționeze corespunzător. Într-o astfel de arhitectură, după cum este prezentată de R. Brooks [3], ca arhitectură de module subsumate, agenții pot fi adăugați, eliminați și modificați fără a influența alți agenți. În același timp, modularitatea, distributivitatea, flexibilitatea și robustețea constituie o abordare atractivă. Felul în care agenții realizează diverse acțiuni rămâne o problemă deschisă, la fel și identificarea factorilor pentru care agenții pot coopera.

P. Maes [2] precizează două ipoteze ale unui astfel de sistem:

- Acțiunile raționale ale sistemului global pot decurge prin lăsarea libertății agenților de a se activa și inhiba în modul corect.
- Nu sunt necesari agenți ‘birocrati’, a căror competență este doar de a controla, activa, inhiba alți agenți.

Selectarea acțiunilor este efectuată conform unei teorii calitative [4], cu legi bazate pe mai mulți parametri de tipul: viteză, optimalitate, orientare spre obiectiv și oportunism.

În studii anterioare [5, 6, 7], am considerat o funcționare a agenților care are la bază grafuri de context, care au ca obiective principale, uniformitatea protocoalelor și a reprezentării cunoștințelor. De asemenea, contextul reprezentat ca graf trebuie să poată fi utilizat și de agenți reactivi. În acest sistem cele două aspecte ale sistemului, identificate de P. Maes și prezentate mai sus trebuie să rămână valide. Orice agent, deși are un scop total diferit de ceilalți, trebuie să poată reprezenta datele în același fel și de a comunica informația pe care el o deține despre sistem sub un protocol comun. Astfel, fiecare agent poate lua orice rol, la momente diferite de timp, atât de complex pe cât îi permite performanța dispozitivului pe care el rulează. Dacă ne propunem o metodă comună de a reprezenta orice tip de cunoștințe, care să îmbine sub același comportament, atât agenți cognitivi cât și reactivi, atunci acest sistem de grafuri trebuie să funcționeze ca și cum ar avea un design specific agenților reactivi. Modul de funcționare al acestui nivel din arhitectura agentului nu va fi complet reactiv, chiar dacă acesta va rula pe o mașină simplă (ex. un senzor de temperatură). Orice agent trebuie să fie capabil de a-și schimba interesul dinamic, sau augmenta vechile capacități cu altele noi, chiar dacă mașina pe care rulează va îndeplini aceeași funcție permanent. Acest nivel din arhitectura unui agent, specific grafurilor și interacțiunii între agenți folosind grafuri, trebuie să fie realizat independent de activitatea temporară a agentului în cadrul dispozitivului pe care rulează.

Astfel putem atinge acea viziune, de a avea un mediu computațional care să poată funcționa singur, adaptiv, în funcție de situații, scenarii și abilități ale dispozitivelor fizice, fără ca noi să fim nevoiți să intervenim explicit în manipularea agenților. Putem astfel avea, o infrastructură care se poate modela intereselor noastre (ex: un ceas pe care putem citi știri din sport sau un cititor de carduri care să ne informeze de programul zilnic). Deși pentru

dispozitivele dedicate, nu pare o idee prea utilă, pentru dispozitive multifuncționale, precum telefoanele mobile este o posibilitate foarte interesantă și utilă dar în care nu s-a investit suficient de mult efort. Pentru a realiza un astfel de sistem universal este necesară doar posibilitatea de a interacționa cu mediul în orice fel, fiecare agent fiind capabil de a asimila orice context.

În continuare, vom analiza aspectul organizării grupurilor de agenți și modul cum pot coopera. Vom vedea de ce potrivirea de grafuri a fost aleasă ca soluție pentru luarea de acțiuni de către agenți și cum poate fi tratată potrivirea unor șabloane cu aspecte temporale. Deoarece problema potrivirii grafurilor este NP-completă, vom vedea cum se poate optimiza pentru o funcționare îndelungată în timp real.

2. Coordonarea și cooperarea agenților.

În cadrul sistemului de reprezentare a contextului ca grafuri, cooperarea are o particularitate interesantă. În literatura de specialitate, cooperarea agenților este văzută ca o metodă de coordonare în rezolvarea uneia sau mai multor probleme. JyiShane Liu [8] prezintă o metodologie, numită Partiție de constrângeri și Reacție de coordonare (CP & CR – *Constraint Partition and Coordinated Reaction*), unde o soluție a unei probleme provine din evoluția procesului computațional a unui grup de agenți reactivi diverși care interacționează. L. Gasser [12] prezintă un studiu „Cooperative Distributed Problem Solving” (CDPS), care este extins de K. Decker în [10] și [11], a cărui strategie de bază este Divide et. Impera. Astfel, problemele sunt descompuse în subprobleme independente care pot fi repartizate ca task-uri la agenți diferiți. În general, studiile efectuate în Inteligența Ambientală despre rezolvarea distribuită de probleme diferă de cele din Inteligența Artificială care s-au orientat înspre sisteme multi-agent compuse din indivizi sofisticăți [9], numiți agenți cognitivi.

Coordonarea trebuie să dețină la bază un proces care să facă posibilă o convergență rapidă în controlul interacțiunilor [8]. Acesta poate fi alcătuit pe baza a trei principii:

- Perturbare minimă - Când un agent rezolvă conflicte, interacțiunile trebuie să fie inițiate doar atunci când e necesar și astfel încât să reducă șansele de a cauza alți agenți preocupați să inițieze alte interacțiuni.
- Insule de încredere – Consensul este atins de un proces de evoluție, coerent, bazat pe deciziile de grup, bazate pe insule de încredere și modificându-le doar când coerența grupului este percepută ca fiind nefezabilă sub presupunerile curente.
- Prevenirea ciclurilor – Comportamentul care conduce la bucle, precum schimbările oscilatorii ale valorilor de către o submulțime a agenților trebuie prevenite.

În cadrul studiului nostru, cooperarea între agenți trebuie văzută ca o funcționalitate auxiliară, care le permite agenților să devină mai ‘informați’ și să își îmbogățească contextul dinamic; nu ca pe un scop colectiv. Deoarece fiecare agent funcționează independent de grup pentru a furniza sau îndeplini propriile obiective, cooperarea cu grupul este un efect al diseminării informației de context pe care agentul încearcă să o identifice în mediu pentru un interes propriu. De asemenea, fiecare agent trebuie să își facă cunoscut contextul și să îl poată pune la dispoziția altor agenți, fără a încerca însă să ajute în îndeplinirea intereselor altor agenți. Astfel, fiecare agent are un scop pur individual iar prin rutarea contextului și a intereselor altor agenți, el ajunge să influențeze activitatea altor agenți, fără să fie conștient de

acest lucru. Putem vedea cooperarea între agenți mai mult ca pe o colaborare cauzală și nu ca pe o coordonare pentru rezolvarea unei probleme așa cum este specific sistemelor multi-agent.

Cele trei principii identificate de J. Liu [8] și prezentate mai sus, în cadrul sistemului abordat în lucrarea curentă, se pot traduce astfel:

- Perturbare minimă – Dacă un agent este într-un proces de potrivire a unui șablon cu un graf de context [5, 6], atunci alt agent nou introdus în sistem, cu multe informații de context noi și cu un grad de încredere mai ridicat, nu trebuie să îi modifice considerabil considerabil graful acestuia.
- Insule de încredere – se vor transforma în zone de interese. Un grup de agenți dintr-un laborator de chimie, deși vor fi diferiți, ei vor avea interese asemănătoare. În schimb, interesele lor vor fi diferite de mulțimea agenților din parcare alăturată laboratorului. Doi agenți cu interese asemănătoare care își pot declanșa unul altuia diverse acțiuni ca urmare a întâmplării unor situații trebuie să poată comunica rapid și să rămână flexibili și agili la schimbarea de context, ca să poată lua acțiuni rapide. De cealaltă parte, însă, dacă doi agenți au interese diferite și nici nu pot comunica direct, informația între ei va trebui să se propage diferit, astfel încât să nu inunde toată rețeaua cu schimbări de context destinate unor agenți neinteresați.
- Prevenirea ciclurilor – Este incomod dacă se petrece însă nu trebuie detectată. Dacă mediul este oscilatoriu, atunci contextul agenților trebuie să se adapteze și să revină la o situație inițială, de un număr indefinit de ori, atât timp cât agenții rămân în conformitate cu contextul. Aceste situații se pot rezolva cu șabloane de context care au fost potrivite în trecut iar informația rămâne în grafuri care au fost valabile anterior, au dispărut ca urmare a schimbărilor de context dar nu au un timp de expirare.

D. Servat [13] sintetizează principiile unui agent reactiv într-un sistem PI (Pervasive Intelligence) cu următoarele principii, la care vom arăta în ce mod sunt rezolvate în problema noastră de agenți cognitivi care pot lua decizii asupra contextului:

- Designul se bazează adesea pe metafore, multe dintre ele fiind bine adaptate la sisteme care constau în agenți eterogeni.
 - o Un interes exprimat de o persoană în limbaj natural poate fi deseori metaforic și interpretabil. De aceea potrivirea în grafuri se face cu ajutorul expresiilor regulate, care conferă unui șablon care se dorește a fi identificat, o expresivitate sporită. Am analizat introducerea expresiilor regulate pentru potrivirea muchiilor și este o soluție ce nu aduce un impact mare de performanță [17].
- Agenții reactivi nu sunt neapărat înzestrați cu un comportament rațional și deseori se bazează pe arhitecturi individuale care îi coordonează.
- Se folosesc populații masive de agenți care suferă de o comunicare slabă și de interacțiuni probabilistice.
 - o Motivația fezabilității folosirii unui număr mare de agenți se datorează faptului că interacțiunile probabilistice se realizează doar la început, când agenții nu cunosc interesele altor agenți și nu s-au clusterizat în grupuri.
 - o În lucrarea lui Lu Linyuan [25] s-a experimentat pe mai multe seturi de date, modul cum se împrăștie informația într-o rețea de agenți. Concluziile finale arată că o rețea Small-World cu modificări aleatorii de topologie provoacă o propagare

mai bună decât o rețea de agenți regulată. Așadar, o poziționare aleatorie în mediu, pentru agenți, nu trebuie să fie problematică pentru răspândirea contextului.

- O metodă pe care o vom implementa în sistemul curent [26], care a apărut recent, studiază o comunicare eficientă într-un graf cu o mare slabă-conductanță (Large weak conductance) (Ex, grupuri de clici, conectate prin punți critice), specifice modului cum sunt grupați agenții în problema studiată de noi.
- Se bazează pe organizare individuală și nu pe scheme organizaționale
- Mediul în care agenții acționează este considerat dinamic și nu se axează pe nicio reprezentare simbolică.
 - De aceea se folosesc grafuri de context pentru reprezentarea cunoștințelor, deoarece acestea oferă multă flexibilitate, putând astfel reprezenta orice.
 - Problemele întâmpinate prin abordarea unei strategii particulare de reprezentare (exemple: logică modală, logică cu predicate, ontologii) pot fi soluționate printr-o reprezentare valabilă universal.
- Validarea sistemelor reactive este una empirică.
 - Se încearcă potrivirea șabloanelor până când aceasta reușește.
- Agenții reactivi încorporează metode evoluționiste de tehnici de învățare.
 - În cazul nostru, agenții vor învăța noi concepte sau relații cu care își vor augmenta contextul. De asemenea pot învăța șabloane noi. Pot învăța contextele agenților cu interese comune.

3. Potrivirea grafurilor ca sistem de inteligență omniprezentă

Sistemele multi-agent s-au dezvoltat mult în ultimii ani, fiind acum utilizate pe scară largă. În Inteligența Ambientală, însă, nu există încă implementări practice la scară largă, care să lanseze mii de agenți într-un mediu, cu un scenariu care să beneficieze din plin de posibilitățile oferite de un sistem multi-agent (MAS) controlabil și inteligent. În domenii precum economia și biologia, MAS par capabile să răspundă la problemele pe care le ridică [13]. Pentru a putea aborda un scenariu complex de inteligență ambientală cu date eterogene și context variabil, designul unui sistem multi-agent va trebui să considere și alte surse de inspirație, în afara copierii modelelor economice sau sociale în care grupul de agenți încearcă să maximizeze câștiguri, să voteze un lider, să funcționeze ierarhic. O direcție interesantă a PI constă în crearea unor 'ecosisteme' de agenți fizici eterogeni, „open-minded”, dinamici în grupuri masive care interacționează, organizați după principii chimice, fizice sau biologice[13]. În lucrarea respectivă, sunt precizate studii de computație amorfă, inspirată de procesele stochastice din natură. Acestea reprezintă un punct de interes în cercetarea actuală. O. Simonim [22] prezintă doi algoritmi de rezolvare colectivă a problemelor de către un grup de agenți: Eco-rezoluție, respectiv algoritmi bazați pe răspândirea feromonilor. În aceeași lucrare sunt investigate două modele comportamentale ale agenților și anume:

- Model "Satisfacție – altruism" în care agenții nu au intenții și la un macro-nivel, doar activitățile colective ale grupului pot fi luate în considerare.

- Model reactiv bazat pe interacțiune fizică (senzori, aparate de monitorizare, etc..)

Mark Weiser a popularizat prin anii 1990 termenul de “Ubiquitous computing” [14, 15], care a intrat prima dată ca o viziune în domeniul calculatoarelor în 1960, inițiat de Licklider [16]. Viziunea lui Weiser de atunci este una căreia nu îi putem aduce contra-argumente deoarece în prezent, numărul de microprocesoare încorporate, vândute zilnic pe piață, este mult mai mare decât numărul de calculatoare produse. Totuși, percepția unui mediu de inteligență ambientală, format din “tab-uri”, “pad-uri” și “board-uri” [14] încă nu reflectă situația actuală pe care o avem în casele noastre, la tot pasul. Oamenii pot lucra cu greu la două lucruri diferite în același timp și poate că prezența a zeci de afișaje în jurul unui om este inutilă deoarece acestea nu vor putea fi urmărite.

Se întrezărește astfel, nevoia unui mediu inteligent, care iese și mai puțin în evidență. Anterior, am propus un scenariu în care o persoană în vârstă, Emily, este notificată de o alarmă sau de o voce electronică care o atenționează că este în fața ușii de la ieșire și a uitat să ia cheile și portofelul [17]. Influența mediului asupra unei persoane trebuie să se manifeste atunci când chiar este nevoie. În rest, va trebui să existe un proces ascuns în care sistemul inteligent funcționează iar acest proces poate fi influențat sau nu de către om, în cadrul unei submulțimi de dispozitive.

Aceste dispozitive vor fi de două tipuri [13]: purtate de utilizator (unelte, telefoane, haine, ochelari [18]) și statice, integrate peste tot în jurul nostru. Observăm tendințele actuale, de a avea obiecte inteligente care se pot purta (ex: Google Glass, pedometrul, aplicații mobile de monitorizare).

Principalele limitări în Inteligența Ambientală omniprezentă actuală sunt: populație mică de agenți [19], comunicare standardizată între sisteme diferite, care să fie tolerantă la defecte și posibilă în medii în care variază timpul sau granularitatea lui; Kqml [20] încă nu rezolvă aceste probleme; arhitecturi omogene (ex: BDI [21]), în care se încorporează greu considerente eterogene. Mediile, în prezent, sunt închise și statice iar schemele de funcționare sunt deterministice.

Am standardizat un model de încorporare a timpului în potrivirea de grafuri, care va fi prezentat în capitolul următor. Din păcate, celelalte limitări încă rămân din constrângeri de resurse fizice. Problema eterogenității a fost și ea rezolvată prin folosirea acestui tip particular de grafuri pentru reprezentarea contextului. Problema portabilității aplicației care descrie acest nivel din arhitectura unui agent încă este una în studiu. Cu noile cercetări de translație a codului compilat în limbaj cu instrucțiuni LLVM, portabil în prezent pe arhitecturi Intel x86/x64, Itanium, Arm și AMD sau cu adoptarea universală a HTML 5 ca design grafic pentru dispozitivele mobile, avem puncte de plecare în realizarea unei portabilități așteptate pe viitor între dispozitive.

4. Potrivirea de grafuri cu restricții temporale.

Problema timpului este una dificil de tratat în scenariile în timp real, în care un context expiră după o perioadă sau unele relații sunt valabile doar într-un interval din viitor sau chiar din trecut.

Pentru a vizualiza nevoia introducerii de potriviri temporale, observăm în figura de mai jos trei șabloane, reprezentate ca grafuri orientate.

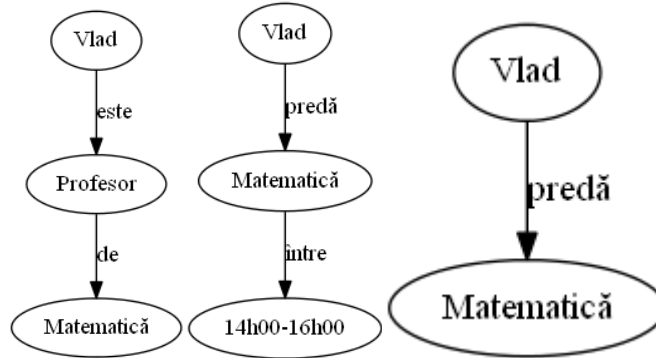


Fig. 1 – Exemple de șabloane cu validitate temporală diferită

Prin asimilarea celor două șabloane în contextul agentului, va rezulta următorul graf de context:

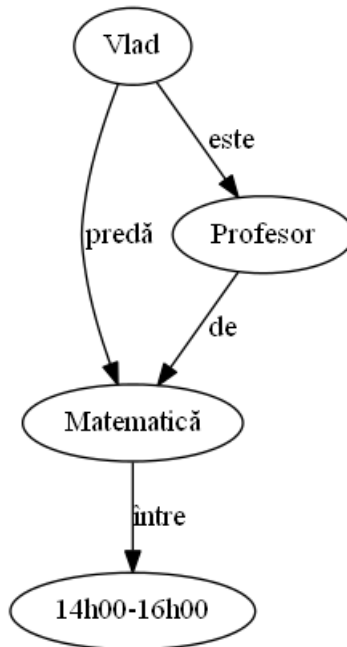


Fig. 2 Exemplu de graf de context care conține informații temporale diferite

Sub-șablonul “Vlad predă matematică între 14h00 – 16h00” poate fi șters la sfârșitul zilei, deoarece doar astăzi, cursul respectiv se ține în intervalul orar respectiv. Dacă am șterge toată

această ramură din graf, va dispărea relația “Vlad -> predă -> Matematică”, care provine din mai multe șabloane, într-unul din ele având o durată de valabilitate permanentă.

Astfel, am convenit folosirea următoarei strategii:

- Fiecare muchie va avea două amprente de timp
 - o Timp de expirare al relației, t ;
 - o Interval închis $[t_{min}, t_{max}]$ în care relația respectivă se întâmplă efectiv în realitate.
(Exemplu: pentru relația “Matematică -> între -> 14h00 – 16h00”, t poate fi considerat $t = 17h00$, când se decide că muchia respectivă nu mai e de actualitate. În schimb ea e valabilă toată ziua, până la timpul expirării. $[t_{min}, t_{max}]$ însă va fi egal cu intervalul $[14h00 – 16h00]$, deoarece atunci se petrece evenimentul în realitate).
- Un graf va avea aceleași două amprente de timp, iar $t_{graf} = \min_{i < card(E)} \{ t_i \}$. Iar
$$[t_{min-graph}, t_{max-graph}] = \bigcap_{i \in |E|} [t_{min-i}, t_{max-i}]$$
- Fiecare muchie dintr-un graf șablon, va avea ca timp de expirare, respectiv interval de valabilitate, aceeași valoare ca valoarea grafului, care se calculează după ajustarea grafului.
- În grafurile contextuale, muchiile au coadă de timpi de expirare și un singur interval de validitate. Dacă o muchie este asimilată în grafurile de context cu mai multe intervale de validitate, atunci înseamnă că sunt două muchii distincte între aceleași două noduri dar care fac referire la resurse diferite și vor fi adăugate ca muchii paralele dar distincte. Coada de amprente de timp de expirare va reține într-o mulțime ordonată timpii în care muchia respectivă expiră pentru șabloane diferite. Când se face reajustarea temporală a grafului, se vor elimina din coada de timpi, toate valorile mai mici decât timpul prezent. Dacă această coadă de timpi devine vidă, înseamnă că muchie nu mai e valabilă pentru niciun șablon și aceasta este eliminată.
- Dacă se dorește potrivirea unui șablon cu un interval de validitate restrâns cu un graf de context, atunci se va construi un nou graf cu relațiile din context, valide între limitele respectie și se va face o potrivire normală, fără să se mai țină seama de timpi.
- Grafurile șablon va conține cele două informații de timp, într-un hipernod cu format special sau va specifica timpii, ca proprietăți ai fiecărei muchii, sau ca o proprietate globală a întregului graf.
- Când un agent își va încheia activitatea el va serializa grafurile într-o memorie persistentă, de unde timpii vor fi citați la repornire ca atribute ale muchiilor, în format .dot.
- Fiecare agent va reține intern, toți timpii, aliniați cu fusul orar GMT, cu granularitate de milisecunde și nu ține cont de ajustările periodice care se aplică asupra datei universale, deoarece datele sunt convertite în timpi numerici, global valizi.
- Dacă există diferență de timpi între agenți sau ei vor cronometra timpul diferit, ei trebuie să aibă o modalitate de conversie a timpului local într-un format standard,

aliniat GMT și se va ține cont de diferența de timp între cei doi agenți, recalculându-se astfel amprentele de timp specifice fiecărei muchii, respectiv întregului graf.

5. Îmbunătățirea performanțelor potrivirilor în timp real

Pentru un agent, domeniul de interes este reprezentat de grafurile șablon pe care le conține la un moment de timp și acestea se pot înmulți ca număr în urma interacțiunilor cu mediul și cu alți agenți din mediu. Unele potriviri pot fi nesatisfăcătoare datorită faptului că agentul nu cunoaște informații suficiente în domeniul respectiv, care să fie reprezentate în graful contextual. Astfel, un procent semnificativ de informație necunoscută va rămâne încă într-un stadiu necunoscut și se vor reîncerca noi potriviri la noi schimbări de context.

Graful de context se poate modifica foarte mult și foarte repede, în urma modificării validității temporale și în urma modificărilor provenite din reactualizarea informației în urma interacțiunii cu alți agenți. Astfel, o cantitate foarte mare de informație trebuie procesată redundant, efectuând potriviri asupra grafului contextual.

Din acest motiv, este nevoie de o metodă eficientă de a putea furniza rapid potriviri și rezultate pe baza potrivirilor anterioare, deoarece potrivirea grafurilor contextuale are o complexitate exponențială și un agent care nu poate furniza rezultate direct, poate fi neutilizabil.

O idee ar fi să analizăm șabloanele potrivite anterior și să identificăm modificările care apar în cadrul contextului și impactul pe care îl au asupra șabloanelor deja identificate. Însă, verificarea izomorfismului cu toate șabloanele poate fi foarte costisitor. Trebuie găsită o metodă care să poată efectua identifica izomorfisme, considerând toată mulțimea de grafuri, într-o singură traversare. O abordare condusă de M. Weber [23], care este un studiu de optimizare asupra unui algoritm descris de Messmer și Bunke [24], consideră un algoritm care efectuează identificarea izomorfismului împotriva unei biblioteci de grafuri.

Definiție: Fie G un graf. Atunci $A(G)$ este mulțimea tuturor permutărilor matricei de adiacență a lui G .

$$A(G) = \{ M_P \mid M_P = PMP^T \text{ unde } P \text{ este o matrice de permutare } n \times n \}$$

Pentru a formaliza izomorfismul subgrafurilor, putem defini izomorfismul după cum urmează:

Definiție: Fie G_1 și G_2 două grafuri și M_1 și M_2 matricele lor de adiacență corespunzătoare. G_1 și G_2 sunt izomorfe dacă există o matrice de permutare P , astfel încât:

$$M_2 = PM_1P^T$$

Astfel, matricea de permutare P , poate fi considerată ca o funcție bijectivă f care mapează nodurile lui G_1 la G_2 și invers. Astfel, identificarea unui izomorfism între G_1 și G_2 este echivalent cu găsirea unei permutări a matricei P pentru care definiția de mai sus este adevărată.

Definiție: Dându-se două subgrafuri G_1 și G_2 , există un izomorfism de la G_1 la G_2 dacă există un subgraf S inclus în G_2 astfel încât G_1 și S sunt izomorfe. Pentru a exemplifica algoritmul lui Messmer și Bunke, se consideră următorul graf:

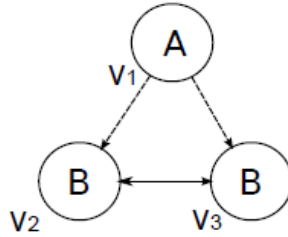


Fig. 3 Exemplu de graf ilustrativ

Permutările matricii de adiacență sunt următoarele:

	V1	V2	V3
V1	A	1	1
V2	0	B	2
V3	0	2	B

M₁

	V1	V3	V2
V1	A	1	1
V3	0	B	2
V2	0	2	B

M₂

	V2	V1	V3
V2	B	0	2
V1	1	A	1
V3	2	0	B

M₃

	V2	V3	V1
V2	B	2	0
V3	2	B	0
V1	1	1	A

M₄

	V3	V1	V2
V3	B	0	2
V1	1	A	1
V2	2	0	B

M₅

	V3	V2	V1
V3	B	2	0
V2	2	B	0
V1	1	1	A

M₆

Fig 4. Permutările matricii de adiacență

Scopul algoritmului este de a construi un arbore de decizie corespunzător tuturor grafurilor din colecția în care se dorește identificarea unui izomorfism. Elementele din noduri sunt stocate ca:

$a_i = (m_{1i}; m_{2i}; \dots; m_{ii}; m_{i(i-1)}; m_{i1})$, adică din rândul și coloana i , care se adaugă submatricii precedente din nodul părinte.

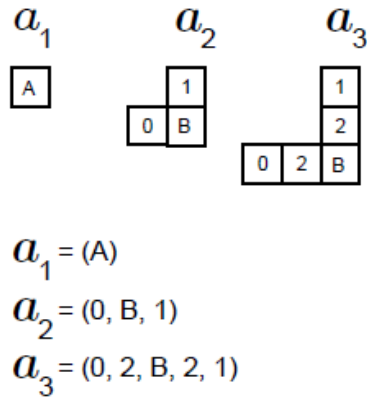


Fig. 5 Pașii iterativi de potrivire a matricelor

Arborele de decizie construit pe baza permutărilor matricelor de adiacență este următorul:

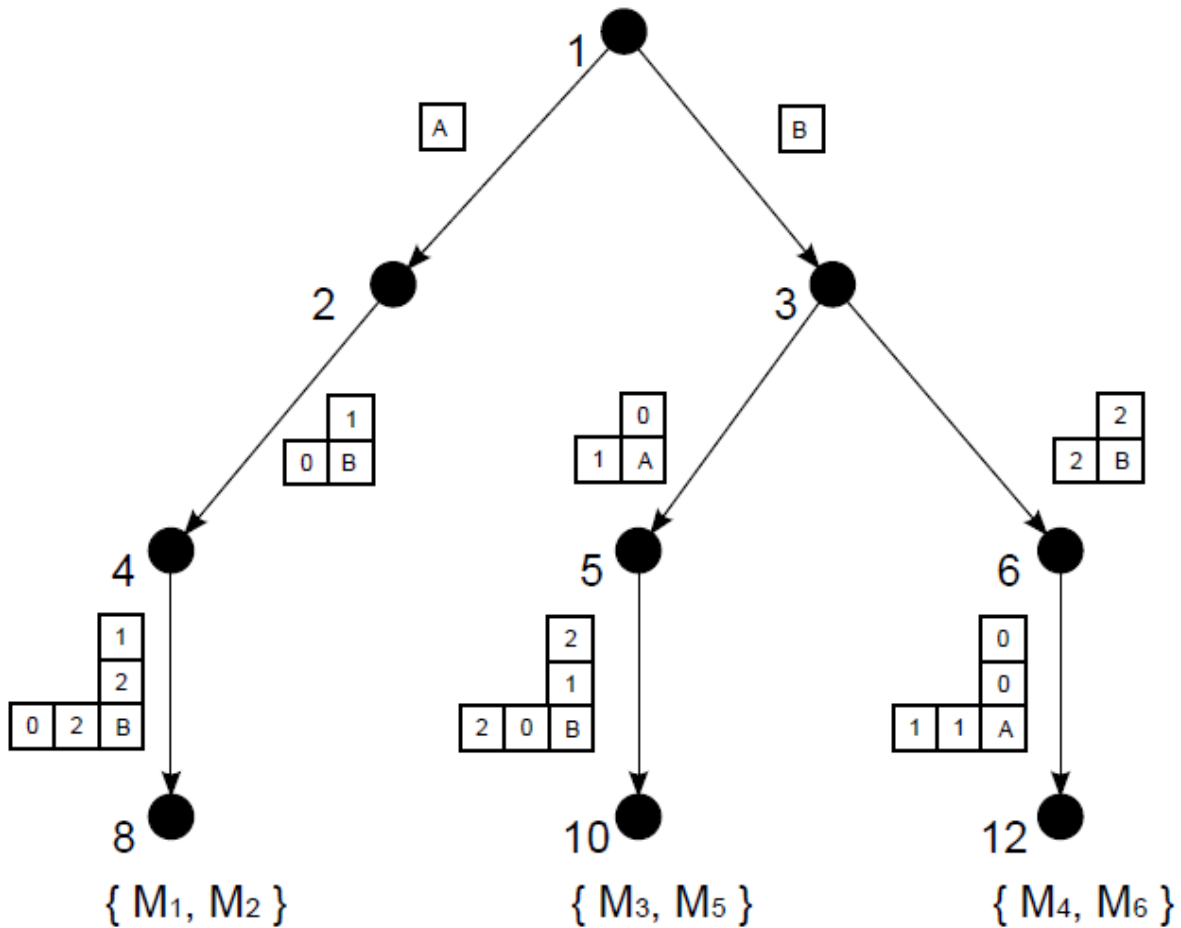


Fig. 6 – Exemplu de arbore de decizie

Algoritmul se poate optimiza, cum este descris de M. Weber [23], astfel încât să nu se mai considere toate permutările ci doar o parte. Identificarea ramurilor la traversarea arborelui pentru aflarea izomorfismului se poate face într-o complexitate amortizată la $O(1)$, ținând cont

de faptul că nodurile pot fi stocate într-o tabelă hash, sortate după un criteriu de numerotare a etichetelor.

Acest algoritm poate fi util pentru potrivirea unui șablon nou asemănător cu șabloane anterioare, deja identificate. Dezavantajul acestei abordări constă în faptul că algoritmul inițial de reconstrucție al arborelui de decizie este foarte costisitor și trebuie reconstruit la fiecare schimbare de context, deoarece mulțimea de șabloane este posibil să nu mai fie valabilă, cel puțin pentru unele dintre șabloane.

6. Concluzii

În urma analizei ipotezelor pe care trebuie să le îndeplinească un sistem dinamic de Inteligență Ambientală, putem spune că abordarea propusă poate fi implementată și testată într-un mediu real.

Introducerea factorului temporal în potrivirea grafurilor aduce mult mai mult control și utilitate funcționării unui agent. Este important ca agenții să păstreze un protocol comun de reprezentare a timpului sau să ofere o modalitate de transformare într-un format universal, aliniat la fusul orar GMT.

Un sistem bazat pe agenți cognitivi care pot interpreta grafurile de context și pot efectua acțiuni asupra mediului în urma șabloanelor potrivite poate aduce multă valoare sistemului ambiental inteligent. Fiecare agent va avea propriul graf de context și mai multe șabloane pe baza cărora el poate lua acțiuni fără să fie influențat direct de alt agent, însă agenții pot asimila informație bazată pe contextul altor agenți și pe interesele proprii.

Transformarea șabloanelor în arbore de decizie este un proces cu complexitate exponențială, însă identificarea unei potriviri precedente va fi calculată de un algoritm tractabil care va explora arborele creat. Atât timp cât schimbările în mediu nu sunt foarte mari și foarte dese, această soluție poate fi foarte eficientă, deoarece evită efectuarea unui izomorfism maximal între graful șablon și graful de context.

7. Dezvoltări ulterioare

Pentru a avea un algoritm bun de potrivire a grafurilor mai sunt necesare experimente prin care să se analizeze performanța unui comportament în timp real, pentru agenți individuali și timpul consumat în diverse situații de schimbare dinamică a contextului și a șabloanelor. Pentru o imagine bună asupra întregului sistem avem nevoie de instrumente grafice prin care se poate manipula scenariul în timp real și se pot observa rezultatele.

Alte îmbunătățiri interesante care se pot aduce implementării sunt:

- Identificarea preferențială de muchii sau noduri dintr-un graf șablon. Anumite elemente pot avea prioritate mai mare și se doresc să fie identificate cu o preferință mai mare sau să existe obligatoriu în izomorfism.
- Studiarea de metode portabile de adaptare a algoritmilor folosiți pe dispozitive specifice domeniului Inteligenței Ambientale.
- Modul cum agenții vor folosi și vor percepe cunoștințele incerte și cum anumite grade de incertitudine pot conduce la decizii diferite.

BIBLIOGRAFIE

1. Marvin Minsky – The society of mind, 1988
2. The dynamics of Action Selection, Pattie Maes, AI-LAB, V.U.B, Pleinlaan 2, B-1050 Brussels
3. Brooks R. (1985) A Robust Layered Control System for a Mobile Robot. IEEE Journal of Robotics and Automation. Volume RA-2, Number 1.
4. Maes P. (1989). A Qualitative Theory of the Dynamics of Action Selection. AI-memo 88-28, AI-LAB.
5. Andrei Olaru - A Context-Aware Multi-Agent System for Aml Environments, București, 15 dec. 2011.
6. Andrei Olaru - Olaru, A., Florea, A. M., and El Fallah Seghrouchni, A. (2011). Graphs and patterns for context-awareness. In Novais, P., Preuveneers, D., and Corchado, J. M., editors, Proceedings of International Symposium on Ambient Intelligence, University of Salamanca (Spain), 6-8th April, volume 92 of Advances in Intelligent and Soft Computing, pages 165-172. Springer. ISBN 978-3-642-19936-3, ISSN 1867-5662.
7. Adrian Dobrescu – Potrivirea în grafuri pentru recunoașterea contextului, Lucrare de diplomă, 2012.
8. JyiShane Liu – Collective Problem Solving through Coordination in a Society of Reactive Agents, CMU-RI-TR-94-23. The Robotics Institute Carnegie Mellon University, Pittsburgh, PA 1521
9. B. Chaib-Draa, B. Moulin, R. Mandiau and P. Millot – Trends in Distributed Artificial Intelligence. Artificial Intelligence Review, 6:35 – 66, 1992.
10. K. S. Decker, E. H. Durfee, and V. R. Lesser. Evaluating research in cooperative distributed problem solving. In Distributed Artificial Intelligence, Vol. 2, pages 485-519. Pitman, San Francisco, CA, 1989.
11. Keith S. Decker. Distributed problem-solving techniques: A survey. *IEEE Transactions on Systems, Man, and Cybernetics*, 17(5):729-739, 1987.
12. Les Gasser and Randall W. Hill, Jr. Engineering coordinated problem solvers. *Annual Review of Computer Science*, 4:203-253, 1990.
13. David Servat, Alexis Drogoul - Combining amorphous computing and reactive agent-based systems: a paradigm for pervasive intelligence. MIRIAD.OASSIS/LIP6. Universite de Paris.
14. M. Weiser. The computer for the twenty-first century. *Scientific American*, 1991.
15. M. Weiser. Some computer science issues in ubiquitous computing. *Commun. ACM*, 1993.
16. J. Licklider. Man-computer symbiosis. *IRE, Transactions on Human Factors in Electronics*, 1960.
17. A. Dobrescu, A. Olaru – Graph Matching for Context Recognition, CSCS 19, 2013
18. Micro optical glasses overview, 2001. <http://www.microopticalcorp.com/>
19. Kristina Lerman. Dynamics of information spread on networks. USC Information Science Institute.
20. T. Finin. Kqml as an agent communication language. In *Proc. of the 3rd Int. Conf. on Information and Knowledge Management*. ACM Press, 1994.

21. A. Rao and M. Georgeff. Bdi agents from theory to practice. Technical report, Technical Note 56, AAIL, 1995.
22. Olivier Simonim, Franck Gechter – An Environment-based Methodology to Design Reactive Multi-agent Systems for Problem Solving, Laboratoire SeT (Systemes et Transports) Universite de Technologie de Belfort-Montbeliard, 90010 Belfort cedex, France
23. Markus Weber, Marcus Liwicki, Andreas Dengel, - Faster Subgraph Isomorphism detection by Well-Founded Total Order Indexing
24. B. Messmer, H. Bunke, A decision tree approach to graph and subgraph isomorphism detection, Pattern Recognition 32 (1999) 1979–1998.
25. Linyuan Lu, Duan-Bing Chen, and Tao Zhou - Small world yields the most effective information spreading
26. Keren Censor-Hillel, Hadas Shachnai - Fast Information Spreading in Graphs with Large Weak Conductance