Învățare şi Adaptabilitate în Modelul de Obiecte Active

Adrian Groza

Universitatea Tehnica Cluj-Napoca
E-mail: Adrian.Groza@cs.utcluj.ro

Dorian Gorgan

Universitatea Tehnica Cluj-Napoca
E-mail: Dorian.Gorgan@cs.utcluj.ro

Rezumat. Lucrarea tratează în cadrul modelului de obiecte active posibilitățile de construire de interfețe inteligente şi adaptive. În prima parte se propune un model optim de învațare în AOM (Active Objects Model) a cunoştințelor care se propagă între diferite entități. Se analizează trei strategii de propagare a cunoştințelor între entități şi se propune o metodă prin care interfețele utilizator pot învața şi se pot adapta comportamentului utilizatorului. Metoda are la baza legea Fitt. Învățarea este sinonimă cu posibilitatea de modificare a dimensiunii țintelor într-un spațiu abstract din AOM. O entitate des accesată îşi va mări dimensiunea în spațiul abstract, rezultând, conform legii Fitt, un timp de accesare mai mic în acest spațiu. Viteza cursorului se modifică în spațiul real astfel încât timpul de acces la meniuri să fie cel calculat anterior. Traiectoria analizată este în prealabil trecută printr-un filtru Kalman.

Cuvinte cheie: modelul de abiecte active, legea Fitt, fuzzy logic.

1. INTRODUCERE

Lucrarea are la bază modelul de obiecte active (AOM) dezvoltat la Universitatea Tehnică din Cluj-Napoca şi ca variantă extinsă, incluzând fire de execuție multiple, la Rutherford Appleton Laboratory, UK. AOM a apărut din necesitatea dezvoltatorilor de aplicații interactive de a putea manipula direct atât structura, cât si evoluția modelului implementat. Modelul constă dintr-un set de agenți activi a căror comportament este dictat de reguli atașate. Structura şi funcționalitatea modelului se bazează pe un set de principii, dintre care: 1) model flexibil care permite programarea vizuală şi definirea comportamentului agenților prin manipulare directă; 2) entități care suportă evoluție activă şi dinamică; 3) cooperare între agenți pentru creşterea adaptabilității; 4) modelare consistentă a unor noțiuni: comportament, traiectorie, reguli, condiții, acțiuni[1]. Cunoştințele agenților sunt

definite prin atribute fuzzy. Aceste cunoştinţe pot fi actualizate în urma unor procese de învaţare supervizată sau nu. Agenţii pot interoga entităţile vecine pentru a-şi decide comportamentul

Pe baza acestor noţiuni tratate pe larg în [1] şi [2] ne propunem să demonstrăm aplicabilitatea imediată a modelului de obiecte active în dezvoltarea de interfeţe adaptive. În continuare, lucrarea este structurată astfel: În secţiunea 2 descriem diverse strategii de propagare a cunoştinţelor între entităţi, insistând asupra propagării memetice. În secţiunea 3 am adaptat modelul de învăţare în timp minim la AOM. În sectiunea 4 propunem o metodă de optimizare a interacţiunii om calculator folosind AOM si legea Fitt, iar în final concluzii şi dezvoltări ulterioare.

2 PROPAGAREA CUNOŞTINŢELOR ÎN AOM

La nivelul limbajului, actualizarea cunoştinţelor este realizată fie prin "mecanismul de legare", fie printr-o metodă explicită de acces la comportamentul agentului: prin acţiunile: *get, set, assign, delete.* La rândul sau, mecanismul de legare este de două tipuri: *legare_de_la* şi *legare_la.* Legarea se aplică şi pentru atributele fuzzy, realizându-se în prealabil o mapare în cazul în care tipul atributelor diferă[1]. Prin folosirea unei astfel de tehnici se reduce încărcarea datorată comunicării dintre obiectele care se apelează şi se uşurează dezvoltarea aplicaţiilor modulare.

La nivel conceptual, actualizarea cunoştinţelor se realizează prin propagare. Fie $X = (x_1 ..... x_n)$ o mulţime de elemente ordonate. Considerăm că fiecare x_i este un meniu, relaţia de ordine fiind cea de la stânga la dreapta. Fie cunoştinţa *K="activitate meniu"* şi valoarea fuzzy μ_{xi} reprezentând gradul de adevăr al propoziţiei: „Agentul *A* ştie că meniul x_i este cel mai activ meniu”. În AOM fiecare element din *X* este legat de entităţile vecine prin cunoştinta *K* care este distribuită între entităţi. Prin definirea unei operaţii de propagare peste *X*, aceasta devine o mulţime relaţională. În actuala definiţie AOM, aceste specificaţii sunt determinate de funcţia $w(D_w, r)$ unde: D_w: maximul cantităţii de cunoştinţe pe care le poate acumula un agent, iar *r* un şir de ponderi care defineşte rata de propogare. În funcţie de caracteristicile şirului *r* pot aparea următoarele situaţii:

2.1 Propagare uniform descrescătoare

În acest caz *r* este *uniform descrescător.* În consecinţă, *propagarea depinde de distanţa dintre elemente:*cunoştinţele se propagă descrescător spre vecinii mai îndepărtaţi. Cunoştinţa va fi distribuită local. În exemplul considerat, această abordare presupune că meniurile îşi cunosc ponderea relativă de activare doar într-

o vecinatate. Astfel, influența învățării se resimte doar când mouse-ul ajunge în respectiva regiune.

2.2 Propagare constantă

Dacă *r este constant, propagarea nu depinde de distanța dintre elemente:* cunoştinţele se propagă integral spre toate elementele înrudite. Fiecare element din mulțimea *X* îşi cunoaşte ponderea în ansamblul total de elemente înrudite. Ca urmare, influența învățării va fi resimțită indiferent de regiunea activă la un moment dat.

2.3 Propagare memetica

În continuare vom discuta un model memetic de distribuție a cunoştinţelor şi modul cum se poate adopta acesta în AOM. Termenul "meme" a fost introdus de Dawkins in "The Selfish Gene" (1976). În continuare, vom referi acest termen prin "*mimă*". Mima are aceeaşi funcție în propagarea informației sau a cunoştinţelor cu cea pe care genă o are în propagarea informației biologice. Astfel, propagarea şi distribuția cunoştinţelor se va realiza printr-o componentă evoluționistă. Învățarea va fi sinonimă cu evoluția (în sens Darwinian). Este practic o generalizare a învățării actuale în AOM a unei cunoştinte prin întărire sau repetare. Mima este cel mai mic şablon de informație care poate fi memorat de către o entitate AOM şi care poate fi copiat în memoria altei entități. Aceasta include: obiceiul unui utilizator, îndemânarea acestuia, timpul petrecut, etc. Pentru a fi replicată, o mimă trebuie să treacă prin patru stări succesive[3]:
1)Asimilarea: o mimă trebuie să fie capabilă să "infecteze" o entitate AOM. În model, propagarea se face pe criteriul vecinătății: o cunoştință se poate propaga doar entităților vecine cu care se află în relație. Vecinătățile initiale sunt definite la crearea aplicației. Ele pot fi deformate în spațiul abstract (secțiunea 5), putând fi definite prin mai multe criterii (metrici) decât cele spațiale. Problema constă în modul în care o entitate decide dacă informația receptată este utilă sau nu. Pentru a fi asimilată o mimă trebuie să fie *observată, înțeleasă* şi *acceptată.* Observarea presupune receptarea de la o entitate vecină. Pentru a fi înțeleasă, o mimă trebuie să poată fi reprezentată în "sistemul cognitiv" al entității receptoare. Cunoştinţele anterioare pe care o entitate le deține nu trebuie să fie diametral opuse noii informații primite. Într-un astfel de caz, şansele de acceptare sunt mult mai mici. În consecință, funcția *Dw* depinde de noua cunoştință *K* şi de cunoştinţele curente *KB* (knowledge base): $D_w=D_w(K, KB)$.
2) Retenția: reprezintă durata în care o mimă este reținută în memoria entității. Cu cât stă mai mult timp, informația respectivă are şanse să se distribuie entităților vecine şi să evolueze spre o cunoaştere cât mai exactă a domeniului. În exemplul

considerat, cu cât un agent îşi menţine mai mult propriile informaţii (şi respectiv le trimite entităţilor vecine) cu atât şansele de învăţare corectă a preferinţelor utilizatorului cresc. Dacă aceste preferinţe evoluează în timp, sistemul trebuie să permită un mecanism prin care vechea hartă cognitivă se estompează. O cunoştinţă care nu se mai poate lega de cunoştinţele curente va fi ştearsă din memorie.
3) Exprimarea: reprezintă încapsularea ideii (mimei, cunoştintei, informaţiei) într-un mesaj care poate fi transmis diverselor entităţi. Mima nu se propagă neaparat în forma sa iniţială (prin clonare), ci este în general alterată de entitatea emiţătoare conform propriei reprezentări a mediului.
4) Transmisia: se poate realiza în AOM prin mecanismul de legare sau prin *call(* apel).
Folosirea unei astfel de propagări simulează mai real modul în care cunoştinţe distribuite între entităţile AOM contribuie în luarea unor decizii individuale.

3 OPTIMIZAREA ÎNVĂŢĂRII ÎN AOM

3.1 Operatorii procesului de învăţare

În specificaţiile actuale AOM, învăţarea este modelată de următorii parametri: coeficientul de asimilare Δw (capacitatea de asimilare a unei cunoştinte de către o entitate), rata de propagare r şi coeficientul de solicitare q (indică numărul de accesari a unei entităţi) [1]. Alegerea acestora trebuie făcută astfel încât să asigure:
a) valori pozitive pentru coeficientul de solicitare în orice situaţie. Se elimină astfel condiţia $q=max(0, q-1)$ din [1]. Atingerea valorii 0 a coeficientului de solicitare este echivalentă cu pierderea unor informaţii din istoria învăţării. Practic, uitarea unei cunoştinţe este echivalentă cu neînvăţarea ei vreodată, ceea ce nu este valabil la agenţii umani. Un individ ştie aprope întotdeauna dacă a uitat o informaţie sau nu a învăţat-o.
b) monotonia procesului de învăţare, în sensul că, o accesare în plus a unei entităţi să genereze întotdeauna mai multă informaţie decât este pierdută prin uitarea care are loc simultan cu această accesare.
Condiţiile se vor impune în funcţie de numărul de entităţi grupate într-un domeniu de învăţare. În cazul învăţarii unui meniu, entităţile care alcătuiesc respectivul meniu vor dicta valoarea coeficientului de uitare.

4.2 Strategie optimă de învăţare

În AOM actual, rata de propagare depinde doar de distanta m dintre elemente: $\Delta r=1/(m+1)$. În lumea reală propagarea cunoştinţelor se face în timp. Comunicarea

între agenţii umani, nu se realizează neaparat sincron, existând o întârziere de propagare în lanţul de interacţiune dintre agenţi. Dacă învăţarea o privim din această perspectivă, distanţa dintre entitaţi include factorul timp: cu cât distanţa este mai mare, cu atât timpul necesar unei cunoştinte să atingă o entitate vecină este mai mare. De asemenea, într-un mediu dinamic, cunoştintele îşi pierd validitatea în timp.

Problema care se pune este de a calcula timpul minim de care un agent are nevoie pentru însuşirea unei cunoştinţe, într-un anumit grad specificat, având în vedere ca ea este propagată de la agenţii vecini şi că entităţile uită în timp cunoştintele acumulate. Considerăm următoarea situaţie: agentul *A* are nevoie să înveţe o cunoştinţă în proporţie de *c=40%.* Învăţarea se realizează prin *n=10* sesiuni de lucru cu utilizatorul. Acesta nu interacţionează direct cu entitatea activă AOM considerată, ci cu agenţi vecini de la care cunoştinţa se propagă. Cantitatea de informaţie acumulată în urma interacţiunii curente este proporţională cu *Δw*(coeficientul de asimilare) şi cu rata de propagare. La aceasta se adaugă cantitatea de informaţie acumulată în sesiunile anterioare, căreia i se aplica coeficientul de uitare *f*. Dupa sesiunea *i+1,* cantitatea din cunoştinţă deţinută va fi:

$$c_{i+1} = (1-f)\cdot c_i + \Delta w \cdot r \cdot \sqrt{d_{i+1}} \qquad (1)$$

unde d_{i+1} este timpul pe care agentul îl dedică învăţării. Radicalul modelază faptul că acumularea informaţiei nu este o funcţie liniară din timpul petrecut de utilizator, deoarece acesta are tendinţa să repete aceleaşi actiuni într-o sesiune de lucru. Până la un moment dat, acest comportament este util în cadrul învăţării prin repetare. Dupa ce cunoştinţa a fost suficient acumulată, apare procesul de saturare, în care sistemul nu mai manifesta aceeaşi sensibilitate la acţiunile utilizatorului. Produsul *Δw*r* defineşte eficienţa *e* a agentului. Practic, randamentul timpului dedicat învăţării depinde de trăsăturile individule ale agentului: de coeficientul de acumulare şi de distanta de la care se propaga informaţia.

Notăm prin c_0 cantitatea din cunoştinta *K* pe care agentul o deţine iniţial. Soluţia ecuaţiei recurente (1) este data de:

$$c_n = a^n \cdot (c_0 + e \cdot \sum_{i=1}^{n} a^{-i} \cdot \sqrt{d_i}) \qquad (2)$$

unde *a=1-f.* Punem condiţia finală $c_n = c$ (3) ca după *n* sesiuni să se fi învăţat procentul dorit. Notând prin *D* mulţimea sistemelor de forma $d=(d_1,d_2,\ldots\ldots d_n)$ pentru care ecuaţiile 1-3 sunt adevarate, elementele din *D* vor forma mulţimea *strategiilor admisibile,* care garantează obţinerea în final a scopului propus. Problema este de a găsi o strategie pentru care timpul total petrecut în procesul de învăţare să fie minim. O astfel de soluţie formează *strategia optimă* [4].

Considerăm următorul exemplu. Se doreşte însuşirea unei cunoştinte la valoarea $c_n=400.$ Procentul cunoscut din ea la momentul începerii învăţării este

c_0*=100,* numărul de interacţiuni cu utilizatorul *n=10*, coeficientul de uitare *f=0.15*, *Δw=1* şi distanţa până la agentul de la care se propagă cunoştinţa *m=4.*

Fără restricţie		Cu restricţie	
Ses Min (d_i^*)	Cunoştina (c_i)	Minute (d_i^*)	Cunoştinţa (c_i)
0	100		100
1 1.03	110	2.92	127
2 1.43	123	4.05	158
3 1.98	140	5.61	194
4 2.74	160	7.76	234
5 3.79	185	9	274
6 5.25	214	9	308
7 7.27	250	9	337
8 10.06	291	9	361
9 13.93	341	9	382
10 19.28	400	9	400
66.81		74.35	

Figura 2: Planificarea învăţării în fiecare sesiune de interacţiune cu utilizatorului pentru însuşirea cunoştintei la valoarea c_n=400 a) fără limită de timp. b) cu limită de timp de 9 minute pe sesiune.

Cu cât numărul de sesiuni este mai mare, cu atât timpul alocat diferă de la o sesiune la alta. Pe măsură ce *f* creşte, timpul alocat pentru sesiunile finale este din ce în ce mai mare. Poate apărea situaţia în care utilizatorul nu foloseşte un timp atat de îndelungat d_i^* sistemul. Pentru evitarea ei, se impune o limită de timp pe sesiune. Aceasta ar trebui să fie mai mică decât timpul mediu al unei sesiuni de lucru cu un prag suficient de mare pentru a creşte siguranţa acumularii cunoştintei dorite. După rezolvarea ecuaţiilor [5] se obţin rezultatele din fig. 2. Limită superioară se alege în funcţie de utilizator, învăţarea fiind adaptabilă din acest punct de vedere. Dacă diversele cunoştinte au ponderi diferite, unei cunoştinte considerate relevante i se va atribui un timp limită mai mare. Într-o astfel de interpretare, în care agentul învaţă simultan diverse cunoştinţe care se propagă de la mai multe surse, este important ca el să-şi optimizeze timpul dedicat învăţării. În caz contrar, procesorul (în AOM există 4 procesoare) dedicat acestei activităţi va fi supraaglomerat.

5 PROIECTAREA TRAIECTORIEI ÎN SPAŢIUL ABSTRACT

Fiecare entitate conţine o informaţie topologică care îi permite localizarea în model. Entităţile sunt procesate în spaţiul bidimensional abstract. Este necesară

definirea unei funcţii care face translatarea între cele două reprezentări.

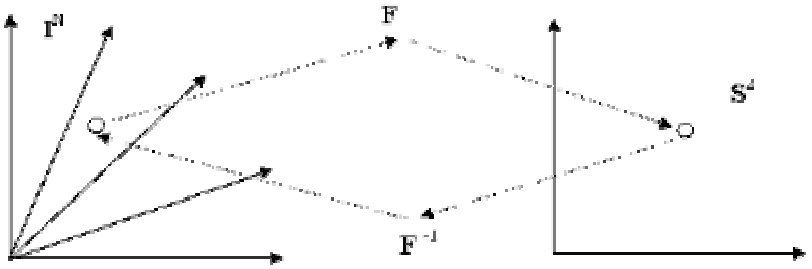

Figura 3: Proiecţia în spaţiul abstract

5.1 Aplicarea Legii Fitt în spaţiul abstract

Legea Fitt afirmă că timpul necesar pentru atingerea unei ţinte este o funcţie de distanţa până la ţintă şi de dimensiunile ei. Consecinţe imediate ale acestei legi sunt: butoanele mai mari se acceseaza mai rapid; cele mai rapizi regiuni ale ecranului sunt cele patru colţuri; un singur rând de butoane înghesuite este de multe ori mai rapid decât două rânduri de butoane dispersate. Legea Fitt are următoarea formă:

$$T = a + b \cdot \log_2(\frac{A}{W} + c) \qquad (13)$$

unde $\log_2(\frac{A}{W} + c)$ se numeşte *factor de dificultate,* iar *a,b,c* sunt constante care se determină experimental. Cu cât distanţa *A* este mai mare şi dimensiunea *W* mai mică, cu atât timpul *T* de atingere a ţintei este mai mare. Ideal ar fi ca dimensiunea butoanelor sau distanţa până la ele să se poată regla în funcţie de obişnuinţele utilizatorului: un buton des folosit va fi "translatat" în apropierea zonei de interes maxim pentru utilizator. Analog, un meniu des utilizat va fi "mărit" în detrimentul altora mai rar accesate. Aplicarea directă a unei astfel de strategii ar avea dezavantaje de ordin estetic sau de control. Practic, interfaţa rezultată se schimbă de la o interacţiune la alta ceea ce va genera disconfort utilizatorului.

O soluţie care elimină dezavantajele menţionate poate fi descrisă prin AOM. Modelul de obiecte active permite reprezentarea atributelor entităţilor într-un spaţiu abstract. Dacă aceste operaţii de optimizare a distanţelor şi dimensiunilor entitaţilor au loc în spaţiul abstract, timpul de acces la ele va fi calculat în acel spaţiu conform legii Fitt. Aplicaţia va prezenta însă utilizatorului aceeaşi interfaţă cu care acesta s-a obişnuit. Învăţarea se realizează în spaţiul abstract. Un algoritm imediat este următorul: la fiecare accesare a unui meniu acesta îşi măreşte dimensiunea cu *p* unităţi. Dacă interfaţa conţine *N* entităţi grafice, acestea îşi vor micşora dimensiunea în spaţiul abstract cu *p/N* unităţi. Timpul necesar accesării unei entităţi

se calculează pe aceste date. Acelaşi timp ar trebui obținut şi în spațiul real (ecranul utilizator). În scenariul considerat, entitățile ramân fixe în ambele spații, deci distanța *W* e constantă. În consecință, se va modifica doar entitatea activă care afişeaza mouse-ul pe ecran. În sintaxa AOM se reprezintă astfel:

set agent(Mouse).attribute(speed), expression(realspeed + x)

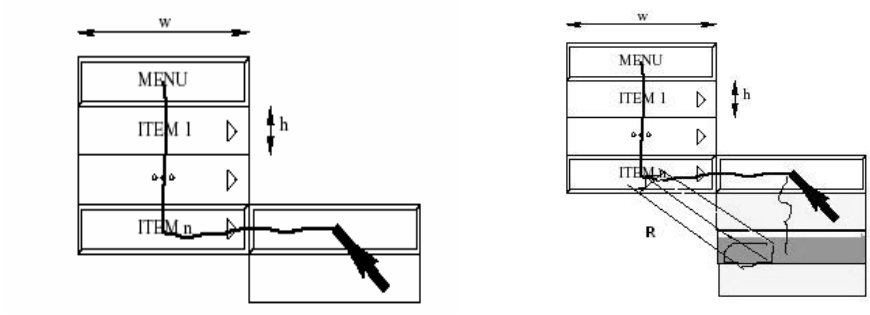

Figura 4: a) viteza navigării prin meniuri este depinde de *w* şi *h*; b) învățarea permite optimizarea traiectoriei. Dacă utilizatorul a accesat des un anumit meniu (portocaliu), el va putea accesa acest meniu şi dacă urmează o traiectorie diagonală. Orice traiectorie din regiunea *R* va corespunde meniului portocaliu.

Astfel, viteza reală de deplasare a mouse-ului este mărită cu valoarea *x* pentru a se obține timpul calculat în spațiul abstract.

5.2 Adaptabilitate prin reguli fuzzy

Entitățile AOM acționează pe baza unui set de reguli. O regulă reprezintă unitatea minimă de comportament care poate fi asociată într-o poziție explicită. Când un obiect ajunge într-o astfel de poziție, el execută regulile asociate respectivei poziții. O astfel de regulă poate fi adaptată logicii fuzzy.

Dacă vitezaMouse este mare se aplică cunoştințele învațate în proporție mare.
Dacă vitezaMouse este mica se aplică cunoştințele învațate în proporție redusă.
Dacă direcțiaMouse indică exact o entitate învățată se activează acea entitate când distanța este medie.
Dacă direcțiaMouse indica aproximativ o entitate învațată se activează acea entitate când distanța este mică.

Figura 5: Exemple de reguli fuzzy aplicate pe cunoştințele învățate despre utilizator

O viteză mare şi o traiectorie precisă indică faptul că utilizatorul ştie exact ce entitate vrea să acceseze. Cu cât aceste atribute sunt mai accentuate cu atât probabilitatea să refere o țintă învățată anterior de system este mai mare.

Pentru o mai precisă estimare a acestor parametri de intrare (viteză, traiectorie) se utilizeaza un filtru Kalman. Mişcarea mouse-ului poate genera o traiectorie uşor tremurată. Pentru a elimina aceste „zgomote" are loc o etapă de preprocesare. înainte de aplicarea regulilor fuzzy. În abordarea clasică a filtrului Kalman[5] erorile survin datorită aparatelor de măsură (în cazul nostru mouse, creion optic). În interpretarea lucrării de fața, eroarea reprezintă distanța dintre intenția utilizatorului şi ceea ce efectiv a transmis dispozitivului. Zgomotele datorate tremurării mâinii sau unei schimbări bruşte şi de scurtă durată a traiectoriei vor avea un efect mai mic asupra reacției interfeței. Se observă ca o astfel de interpretare păstrează caracterul gaussian al erorii, condiție necesară pentru funcționarea filtrului Kalman.

În fig. 4b se ilustrează comportamentul diferit al meniurilor în funcție de cunoştințele acumulate. Dacă meniurile nu ar fi fost învățate, o traiectorie diagonală care iese din meniul ITEMn va genera închiderea meniului MENU. În caz contrar sistemul anticipă că utilizatorul doreşte de fapt accesarea meniului portocaliu. Dacă meniul portocaliu este des accesat în comparație cu vecinii săi, regiunea R se măreşte. Orice traiectorie diagonală va avea şanse mari de a se finaliza în meniul corespunzător. Precizia utilizatorului nu trebuie să fie foarte ridicată, sistemul anticipându-i intenția. Dacă meniurile au ponderi relativ apropiate de accesare, regiunea R se micşorează. În consecință, precizia traiectoriei are o influență mai mare în luarea deciziei care meniu să se activeze. Aplicând filtrul Kalman, decizia dacă o traiectorie aparține sau nu regiunii R este mai certă.

5.3 Exemplu numeric

Se doreşte optimizarea timpului de acces la 3 meniuri frecventate de utilizator. În funcție de numărul de accesări pe care utilizatorul le realizează, sistemul reglează viteza cursorului conform calculelor din spațiul abstract. Am considerat că distanța de la pointer la cele 3 meniuri este egala cu 11. Utilizatorul accesează meniurile de 15 ori cu o distribuție de probabilități. Pe baza dimensiunii meniului din spațiul abstract, se calculează cu legea Fitt timpul necesar accesării meniului.Pentru garantarea acestuia sistemul regleaza viteza cursorului în spațiul real. Reprezentarea grafică a variației timpului este dată în fig 6.

Abordarea de mai sus lucrează doar pe parametrul w din ecuația (13). Modificarea dimensiunilor unei entități în spațiul abstract duce la modificarea distanței dintre pointer şi acea entitate, aspect care trebuie considerat la aplicarea legii Fitt.

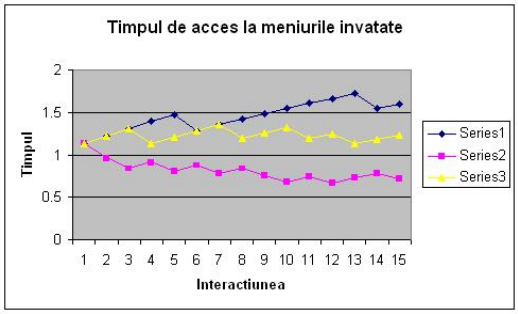

Figura 6 **:** Pentru meniul cel mai des accesat în perioada de învaţare (culoare roşie) se va obţine un timp din ce în ce mai mic de accesare.

6. CONCLUZII

Lucrarea evidenţiază avantajele spaţiului abstract AOM în care sistemul îşi construieşte o reprezentare a cunoştintelor învăţate. O astfel de abordare permite execuţia modelului în acest spaţiu, utilizatorul interacţionând cu aceeaşi interfaţa, dar cu comportament diferit. Adaptabilitatea se construieşte pe baza unor reguli fuzzy pe care AOM le poate defini. Datele de intrare pentru aceste reguli sunt în prealabil filtrate cu o tehnică Kalman. Pe baza legii Fitt, am definit învăţarea prin prisma posibilităţii de modificare a dimensiunii meniurilor în spaţiul abstract.

Necesitatea raţionamentului fuzzy se datorează faptului că simultan, *atât utilizatorul învaţă să folosească sistemul, cât şi sistemul învaţă să se adapteze la utilizator.* Prin legea lui Fitt sistemul învaţă să se adapteze utilizatorului. Prin viteza cu care utilizatorul dirijează mouse-ul se poate cuantifica obişnuinţa utilizatorului cu sistemul. De asemenea, este descrisă o tehnică de optimizare a învăţării pe baza căreia se pot regla parametrii rata de propagare Δr, coeficientul de asimilare Δw sau coeficientul de uitare f. Propagarea cunoştinţelor învăţate şi distribuţia acestora în AOM sunt discutate din prisma strategiilor de propagare.

Referinţe

[1] Gorgan, D., Duce, D.A.: Fuzzy Learning in Multi-Agents Based Interactive Systems, Rutherford Appleton Laboratory, Research Report (reduced variant of the 43 pages paper), February, 1997, pp.1-21.

[2] Gorgan, D., Duce, D.A: Fuzzy Learning in Multi-Agents Based Interactive Systems Rutherford Appleton Laboratory, Research Report, January, 1997, pp.1-43.

[3] Heylighen Francis: What makes a meme successful? Selection criteria for cultural evolution. În Proc. 15th Int. Congress on Cybernetics.

[4] Muntean, I, Vornicescu N.: Strategii de învăţare în timp minim. În Lucrările Seminarului Didactica Matematicii, vol. 8, 1992, pag. 93-112.

[5] Welch, G., Bishop, G. (2002). An introduction to Kalman filter. Technical report, Departament of Computer Science, University of North Carolina at Chapel Hill.