

ASPECTE PRIVIND POTENȚIALUL DE APPLICABILITATE AL MODELELOR COGNITIVE UTILIZATE ÎN INTERACȚIUNEA OM-CALCULATOR

dr. ec. Costin Pribeanu

Institutul Național de Cercetare-Dezvoltare în Informatică - București

Rezumat: Metodele utilizate în prezent în proiectarea interfețelor om-calculator au la bază modele cognitive, care au fost elaborate pentru a reprezenta aspecte esențiale ale procesului de interacțiune, dar care au evoluat în timp, în contextul unor aplicații diverse, din dorința de a fi făcute mai utile în activitatea de dezvoltare. În cadrul acestui articol, se face o discuție asupra potențialului de aplicabilitate a modelelor cognitive actuale, luând în considerație: nivelurile de abstractizare, caracteristicile notației formale și aspectele cognitive modelate.

Cuvinte cheie: interacțiune om-calculator, modele cognitive.

1. Introducere

Evaluarea structurilor de interacțiune om-calculator pe criterii de ergonomie cognitivă necesită studiul factorului uman. O abordare sistematică a bazei cognitive a procesului de interacțiune, trebuie să ia în considerație atât procesele cognitive interne, care au loc în timpul utilizării calculatorului, cât și cunoștințele necesare în acest scop.

În prezent, metodele utilizate în proiectarea interfețelor om-calculator au la bază modelele cognitive, care au fost elaborate pentru a reprezenta aspecte esențiale ale procesului de interacțiune, dar care au evoluat în timp, ca urmare a unor contribuții diverse, din dorința de a fi făcute mai utile în procesul de dezvoltare.

În acest articol, se face o discuție asupra potențialului de aplicabilitate al modelelor cognitive actuale, din punctul de vedere al celui care dezvoltă software. Din acest motiv, spre deosebire de ale lucrări similare, [1], [7], [17], această abordare este orientată pe identificarea elementelor relevante din trei perspective: gradul de acoperire al domeniului studiat, caracteristicile modelării și aspectele cognitive modelate.

În secțiunea 2, se trec succint în revistă principalele modele cognitive, utilizate în interacțiunea om-calculator. În secțiunea 3, se prezintă câteva abordări privind clasificarea și evaluarea potențialului de aplicabilitate al modelelor cognitive. Pe această bază, în secțiunea 4, se argumentează că, pentru a integra modelele cognitive actuale într-un instrumentar coerent, este necesară o analiză mai detaliată a acestora. Restul acestui articol este destinat discutării potențialului de aplicabilitate și concluziilor.

2. Modele cognitive în interacțiunea om-calculator

Unul dintre primele modele cognitive elaborate este MHP (Model Human-Processor), dezvoltat de Card, Moran și Newell (1983) ca bază cognitivă pentru modelul GOMS. Obiectivul principal a fost anticiparea timpilor de execuție, necesari îndeplinirii

unor activități. Avantajul modelului constă în capacitatea de a furniza o arhitectură de procesare, în care poate fi reprezentată o gamă largă de fenomene psihologice. Totodată, se furnizează un limbaj comun în care pot fi translatate activități diverse.

Barnard (1987) a dezvoltat un cadru de lucru mai elaborat pentru modelarea resurselor cognitive ale utilizatorului. Acest cadru de lucru, denumit teoria subsistemelor cognitive în interacțiune, ICS (Interaction Cognitive Subsystems), cuprinde un număr de subsisteme de procesare, care sunt legate într-o arhitectură cognitivă. În această arhitectură cognitivă se pornește de la ideea că acțiunea, percepția și cunoașterea umană pot fi analizate în termeni de module de procesare și fluxuri de reprezentări mentale.

Un model programabil al utilizatorului, PUM (Programmable User Model), a fost elaborat de Young, Green și Simon [19] și este destinat proiectantului pentru a permite o evaluare anticipată a utilizabilității și a atrage atenția asupra unor aspecte specifice. PUM constituie o arhitectură cognitivă, definită printr-un set de restricții asupra reprezentării și procesării cunoștințelor. Ideea de bază este aceea de a trata proiectarea, nu numai din punctul de vedere al proiectului software, dar și al proiectului unui utilizator. În acest sens, PUM reprezintă un instrument de construire a unui model cognitiv al utilizatorului.

Pe baza cunoștințelor privind subsistemul locomotor, Card Moran și Newell [3] au dezvoltat modelul KLM (Keystroke Level Model), având ca scop anticiparea performanțelor umane în execuție. Acest model de estimare a timpului de execuție este aplicabil în interiorul activităților elementare, la nivelul unei interacțiuni constând într-o secvență simplă de acțiuni, care poate dura maximum 20 sec. În prealabil, activitatea este descompusă în subactivități, care sunt apoi translatate în acțiuni fizice.

Tehnica de descoperire GOMS (Goals, Operators, Methods, Selection) a fost elaborată tot de către Card, Moran și Newell [3] și are la bază ideea că utilizatorul acționează în mod rațional pentru a-și îndeplini sarcina de lucru, astfel încât comportamentul său poate fi anticipat, determinând obiectivele, metodele, operatorii și restricțiile sarcinii. Pentru a utiliza cadrul de lucru GOMS în analiza și evaluarea proiectului unui sistem, se presupune că proiectantul va fi capabil să analizeze modul cum utilizatorul va aborda activitatea cu un anumit sistem.

Teoria complexității cognitive, CCT (Cognitive Complexity Theory), elaborată de Kieras și Polson [9], se bazează tot pe o analiză de tip GOMS și este destinată anticipării dificultăților pe care le întâmpină un utilizator în învățarea modului de operare a unui sistem. CCT este un model care încearcă să integreze caracteristicile dispozitivului, cunoștințele privind activitatea, comportamentul utilizatorului și cerințele de procesare cognitivă în scopul de a furniza un cadru de lucru pentru identificarea și cuantificarea unor surse diferite de complexitate cognitivă cu care se confruntă un utilizator care dorește să învețe să opereze cu o interfață.

Metoda NGOMSL (Natural GOMS Language) a fost elaborată de Kieras [9] ca o variantă a tehnicii de descompunere GOMS, conectată la o arhitectură cognitivă, pentru a fi mai ușor de utilizat de către analist, ca o etapă preliminară în aplicarea analizei complexității cognitive. Fiecare instrucțiune NGOMSL corespunde unei reguli de producție din CCT. Ulterior, metoda a fost rafinată și îmbunătățită de Kieras [11] astfel încât să furnizeze o serie de facilități cum sunt: tehnica de descompunere ierarhică GOMS, analiza efortului mental, estimarea timpului de execuție și estimarea ușurinței în învățare.

Limbajul de descriere a acțiunilor asociate sarcinii de lucru, referit și sub acronimul TAL (Task Action Language), a fost propus de Reisner [16]. Acesta are ca scop anticiparea erorilor și a dificultăților de învățare, fără a fi necesară o monitorizare a utilizatorilor. Reisner a introdus ideea că o formă de notație sau o gramatică poate fi utilizată în evaluarea interfețelor cu ajutorul unor metrici care să măsoare diferitele aspecte ale utilizabilității unui proiect. Utilizarea unei gramatici permite compararea unor alternative în termeni de simplitate și consistență.

Gramatica sarcină-acțiune, referită cu acronimul TAG (Task Action Grammar) a fost elaborată de Payne și Green [15] ca o formalizare a limbajelor de interacțiune, destinată pentru modelarea reprezentării mentale a interfeței și pentru specificarea formală a limbajului. Modelul TAG se aseamănă cu modelul TAL prin aceea că produce un limbaj pentru specificarea acțiunilor asociate unei sarcini de lucru. Spre deosebire de acesta, în TAG se acordă mai multă atenție bazei cognitive a gramaticii de interacțiune furnizând un formalism cu care să se poată modela reprezentarea mentală a unui limbaj de interacțiune, așa cum este perceput de utilizator.

O formă extinsă a acestui model este ETAG (Extended TAG), definită de Tauber (1989), care constituie un model cognitiv al utilizatorului și, în același timp, un model conceptual al interfeței. O reprezentare ETAG cuprinde trei părți. Prima parte este descrierea semanticii sistemului care este asociată sarcinii de lucru și se numește mașina virtuală a utilizatorului UVM (User's Virtual Machine). Analiza se poate baza pe o bază canonică CB (Canonical Basis) a conceptelor și relațiilor între acestea. A doua parte a reprezentării ETAG este dicționarul sarcinilor

elementare DBT (Dictionary of Basic Tasks).

A treia parte a descrierii ETAG o constituie regulile de producție, care descriu producțiile de formulări corecte în termenii nivelului de input fizic (acționare taste, mouse), pornind de la descrierea conceptuală a comenzii. Regulile se desfășoară de-a lungul a patru niveluri: specificație, referință, lexical și fizic.

3. Abordări în clasificarea modelelor cognitive

În prezent, există multe clasificări ale modelelor cognitive, care folosesc termenul de model cognitiv în diferite accepțiuni. În sens restrâns, termenul de model cognitiv se referă la modele ale utilizatorului. Într-un sens mai larg, toate modelele de interacțiune sunt modele cognitive, având în vedere că se bazează, în mod implicit sau explicit, pe un model al utilizatorului.

Dix et al [6] propun o clasificare a modelelor după scopul modelării, distingând modele evaluative, care arată, postfactum, în ce măsură un proiect dat îndeplinește anumite cerințe, și modele generative, care contribuie la procesul de proiectare propriu-zis, ghidând proiectantul în luarea celor mai bune decizii; într-o anumită măsură toate modelele pot fi utilizate în acest scop.

Carroll & Reitman [4] identifică trei tipuri de baze de modele cognitive, care caracterizează cunoștințele unui utilizator despre aplicație:

- o simplă secvență de acțiuni, care sunt adecvate unei situații;
- metode care cuprind cunoștințe structurate în obiective generale, subobiective asociate acestora și secvențe de operatori;
- modele privind modul cum lucrează sistemul, ce comportament are în anumite situații, care sunt procesele interne ale acestuia.

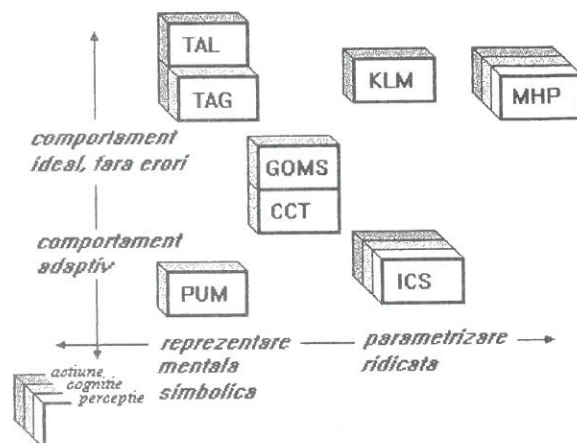


Figura 1. Spațiul modelului (după Simon, 1988)

În [7], se disting mai multe niveluri de abstractizare a interacțiunii: activitate, obiectiv,

semantic, sintactic, lexical și fizic. Analiza făcută de Haan et al pornește de la procesul de delegare a activităților către calculator, identificând următoarele elemente:

- activități externe, care trebuie exprimate în termeni de activități interne, delegate calculatorului;
- operații care trebuie executate de către utilizator, atât la nivel fizic, cât și mental;
- cunoștințe ale utilizatorului, referitor la operațiile și obiectivele sistemului;
- sistemul, ca instrument (sau ansamblu de instrumente) de delegare a activității.

O clasificare a modelelor cognitive, din punctul de vedere al cunoștințelor deținute de operatorul uman este dată de Simon [17], care definește modele de competență, referitoare la comportamentul așteptat din partea unui utilizator ideal, și modele de execuție, care descriu trăsături ale comportamentului legat de executarea unor rutine în contextul unor aplicații. Modelele de execuție descriu atât secvențele necesare de comportament, dar și ceea ce trebuie să știe utilizatorul și modul cum utilizează aceste cunoștințe în îndeplinirea activității.

În scopul unei orientări mai bune în plan teoretic, Simon propune o abordare orientată pe compromisurile făcute de cel care a elaborat modelul, cu scopul de a se concentra asupra unor aspecte prioritare. În acest fel, apare clară funcția principală a fiecărui model și tipul de anticipare comportamentală pe care îl facilitează.

Perspectiva este făcută într-un spațiu tridimensional, în care se pot analiza comparativ principalele stiluri de modelare cognitivă. În plan frontal (axa orizontală și verticală), sunt referite reprezentarea mentală (tipul de procesare mentală modelat și natura informației manipulate) și tipul de comportament. În planul lateral (în adâncime), este referită resursa internă, care corespunde subsistemelor perceptual, cognitiv și motor. Spațiul analizei propus de Simon este prezentat în figura 1.

4. Aplicabilitatea modelelor actuale

Analiza aplicabilității făcută de Simon [17] permite o clasificare generală a modelelor cognitive, care stau la baza diferitelor metode de analiză și de proiectare. Pentru a integra aceste modele într-un cadru de lucru metodologic coerent, orientat pe cerințele de analiză și de proiectare a dialogului om-calculator, este nevoie de o clasificare prealabilă a metodelor, care să urmeze ciclul de viață al unui sistem interactiv. De asemenea, este necesară cuprinderea în această discuție și a unor metode recente de analiză și de proiectare a interacțiunii om-calculator, care se bazează pe extensii ale modelelor cognitive, analizate de Simon.

Întrucât multe abordări recente în analiza sarcinii sunt orientate fie pe cunoștințe, fie pe procesele interne, utilizate în execuția unei sarcini de lucru, nu mai există o delimitare clară între analiză și modelare

în HCI. Mai mult chiar, natura cognitivă a sarcinii de lucru pe calculator face necesară includerea, într-o clasificare generală a modelelor cognitive a tehnicilor și metodelor de analiză a sarcinii. De altfel, este ideea care se desprinde dintr-o serie de abordări privind structura procesului de interacțiune [7], [13], [14].

Din acest punct de vedere, într-o clasificare generală a tehnicilor și a metodelor de analiză și de modelare a interacțiunii, se pot distinge următoarele categorii:

- modele cognitive ale utilizatorului, orientate pe mecanismele interne de procesare, care se declanșează atunci când se execută o sarcină de lucru, adică procesele implicate, conexiunile dintre acestea și parametri de execuție;
- tehnici și metode de analiza sarcinii de lucru, inclusiv pentru translatarea între sarcini externe (definite în termenii activității utilizatorului) și sarcini interne (definite în termeni de comenzi adresate calculatorului);
- metode de specificare a limbajului de interacțiune, prin care utilizatorul comunică cu calculatorul.

În prima categorie intră arhitecturile cognitive, care adoptă un model de procesare a informației umane. Deși arhitecturile cognitive furnizează o perspectivă utilă asupra funcțiilor cognitive ale omului, datorită caracterului foarte specializat (orientat pe psihologie), constituie instrumente dificil de utilizat în procesul de proiectare al unei interfețe om-calculator. Modelele din familia GOMS se vor încadra în analiza sarcinii de lucru, iar metodele orientate pe gramatici ale limbajului de interacțiune (TAL, TAG, ETAG) se vor încadra în ultima categorie.

Această clasificare are avantajul de a integra mai bine instrumentarul folosit în proiectarea structurilor de interacțiune om-calculator în ciclul de viață al unui produs, definit de ingineria software. Pentru definirea unei metodologii cu caracter practic, este necesară o analiză detaliată a potențialului de aplicabilitate al fiecărui model cognitiv, care stă la baza unei tehnici / metode de analiză sau proiectare. În acest scop, este utilă o revedere a caracteristicilor pe care le au acestea din trei perspective:

1. nivelurile de abstractizare utilizate, care determină gradul de acoperire al domeniului;
2. caracteristicile notației formale a limbajului de interacțiune, care determină posibilitățile de reprezentare a structurilor de interacțiune;
3. aspectele cognitive modelate, care determină aplicabilitatea la modelarea competenței respectiv a execuției.

Se observă că evaluarea propusă de Simon [17] se referă numai la aspectele cognitive modelate, analizate din trei puncte de vedere.

5. Gradul de acoperire al domeniului

Din punctul de vedere al acoperirii domeniului HCI, în accepțiunea restrânsă, orientată pe procesul de interacțiune, interesează două aspecte: analiza sarcinii și proiectarea limbajului de interacțiune. În mod adițional (complementar) poate fi considerată evaluarea, având în vedere accentul pus pe elementele cognitive.

În cadrul ciclului de specificare comenzi - interpretare rezultate, Nielsen [14] identifică șase niveluri de abstractizare a interacțiunii: activitate, obiectiv, semantic, sintactic, lexical și fizic. O analiză asemănătoare a abordărilor în modelarea interacțiunii om-calculator este făcută de Haan et al [7] și se bazează pe procesul de delegare a activităților către calculator, identificând patru categorii de obiective ale metodelor: analiza nivelului sarcină, analiza cunoștințelor utilizatorului, anticiparea execuției și reprezentarea interacțiunii om-calculator în scopul proiectării.

Deși conceptul de nivel de abstractizare corespunde în linii mari cu un anumit tip de analiză, având anumite metode, aceasta nu înseamnă că modelele existente și metodologiile care sunt construite deasupra acestora acoperă în mod satisfăcător domeniul. De exemplu, gramatica limbajului de comandă CLG, elaborată de Moran [13], identifică șase niveluri (activitate, semantic, sintactic, interacțiune, spațial și dispozitiv), structurate în cadrul a trei componente (conceptual, comunicare și fizic), dar nu le definește decât pe primele patru dintre acestea.

Un alt model de reprezentare a interacțiunii în scopul proiectării este ETAG, derivat deopotrivă din TAG, în ceea ce privește gramatica adoptată, care este o gramatică a seturilor, și din CLG, în ceea ce privește nivelurile de interacțiune pe care le acoperă, pornind de la nivelul semantic până la nivelul lexical. Ceea ce este remarcabil în ETAG, este introducerea unui nivel conceptual, care ancorează metoda în categoria reprezentărilor destinate să servească practica proiectării, dar cu un plus de rigurozitate față de CLG. În acest sens, prevederea (opțională) a unei baze canonice, văzută ca o "ontologie" a universului utilizatorului, conferă ETAG valențe analitice și fundamentează, totodată, modelul mental al utilizatorului pe care îl creează limbajul sarcină-acțiune.

Prin definirea conceptelor specifice unei aplicații: entități, attribute, relații și evenimente, ETAG oferă posibilitatea preluării, într-o formă utilă etapelor ulterioare de proiectare, a rezultatelor unei analize bazate pe cunoștințe, într-un mod asemănător metodei TKS a lui Johnson [8]. Ceea ce face ca metoda ETAG să fie preferabilă, este orientarea clară pe limbajul de interacțiune, specificat atât la nivel conceptual (în UVM) și semantic (în DBT), cât și la nivelul sintactic-lexical (prin regulile de producție). TKS are dezavantajul unei detalieri excesive a structurii de

cunoștințe, într-un format prea puțin familiar și, în final, nepractic pentru proiectant, ceea ce face ca rezultatul analizei să fie o structură de cunoștințe asociată cu o sarcină, și nu o specificație pentru proiectant.

Se cuvine, însă, a fi menționat faptul că furnizarea unui recipient pentru rezultatele analizei sarcinii nu este suficientă. Nici ETAG și nici CLG, (cu atât mai puțin TAG) nu oferă un instrument pentru efectuarea analizei. Este o trăsătură generală a modelelor lingvistice, care sunt definite pe scheletul teoretic al unei gramatici sarcină-acțiune. Ceea ce lipsește acestor metode este procedura după care se execută comenzile. ETAG și CLG, ca de altfel și TAG, pornesc de la nivelul comenzilor.

O analiză a sarcinii de lucru presupune o descompunere mai amplă, care să permită definirea structurii activității și a contextului în care se încadrează aceasta, într-o perspectivă dinamică, care să evidențieze succesiunea sarcinilor de lucru, evenimentele și/sau stările sistemului (organizațional) care determină lansarea unei sarcini. Pentru aceasta, este necesară o orientare complementară, care să evidențieze modul în care este executată sarcina.

Cerințele de analiză cognitivă ale sarcinii de lucru pe care le ridică interacțiunea om-calculator fac necesară utilizarea unei metodologii de analiză, capabilă să realizeze un asemenea tip de descompunere, pornind de la nivelul conceptual până la nivelul cognitiv. Se poate considera că metoda NGOMSL, care este cea mai recentă variantă GOMS, acoperă această parte a analizei sarcinii. Orientarea pe obiective și metode permite o definire convenabilă a unor sarcini oricât de complicate, astfel încât se pot reprezenta diferite niveluri de abstractizare. În acest sens, este posibilă o structurare în sarcini primare / sarcini secundare, atât în sensul definit de Haan et al [7] cât și în cel definit de Knowles [12].

Practic, descompunerea de acest tip lasă maximum de libertate analistului care poate decide ce anume să facă vizibil în orice metodă GOMS. În acest fel, se pot defini niveluri de analiză în raport cu cerințele concrete ale unui proiect. Pe de altă parte, GOMS oferă și un instrument de proiectare a scenariilor de interacțiune, prin care se pot analiza soluțiile optime de rezolvare a unor procese de interacțiune complexe.

Analiza gradului de acoperire a etapelor de analiză și proiectare a structurilor de interacțiune om-calculator de către metodele actuale furnizează un prim argument pentru a demonstra că niciuna dintre metodele actuale nu acoperă complet domeniul. Totodată, se observă o complementaritate între două metodologii: NGOMSL și ETAG, bazate, fiecare, pe alt tip de notație și având ca direcții prioritare de investigație competența utilizatorului care operează cu un anumit tip de interfață, respectiv execuția propriu-zisă a unei sarcini de lucru.

6. Caracteristicile reprezentării

Portabilitatea rezultatelor etapei de analiză a

sarcinii este o condiție esențială pentru HCI, având în vedere orientarea pe procesul de interacțiune. Din acest punct de vedere, două tipuri de reprezentare prezintă avantaje certe:

1. GOMS, orientată pe obiective, operatori, metode și selecții;
2. ETAG, o formă extinsă a TAG, orientată pe structura sintactică a sarcinii de lucru.

În primul rând, notația GOMS este compatibilă cu modelul KLM, de estimare a timpului de execuție. Această notație permite nu numai o analiză completă, până la nivelul operatorilor fizici și cognitivi, dar și posibilitatea unei evaluări a unor scenarii de interacțiune alternative. În al doilea rând, orientarea procedurală a GOMS permite o analiză în dinamică a procesului de interacțiune, văzută ca un program de execuție pentru utilizator. Varianta NGOMSL cuprinde structurile de control ale unui limbaj de programare: secvențială, alternativă și repetitivă, cu ajutorul cărora se poate reprezenta un scenariu de interacțiune.

Un exemplu de metodă care a eșuat datorită incompatibilității reprezentării este CCT. Principalul inconvenient l-a constituit notația bazată pe reguli de producție, greoaie pentru un proiectant, care a făcut ca metoda să fie dificil de utilizat în practică. Practic, CCT presupunea trei reprezentări diferite: GOMS pentru analiza sarcinii de lucru, rețele de tranziții de stări pentru dispozitiv și reguli de producție pentru procesul mental. În prezent, NGOMSL a integrat filozofia CCT, definind o corespondență unu-la-unu între instrucțiunile din cadrul procedurilor și regulile de producție, care caracterizează o arhitectură cognitivă a utilizatorului.

Al doilea tip de reprezentare este ETAG, care preia din TAG ideea de gramatică a trăsăturilor. Spre deosebire de GOMS, ETAG nu are nici un fel de valențe procedurale. Descompunerea se face de la nivelul semantic la nivelul lexical și evidențiază structura limbajului. În primul rând, ETAG extinde notația TAG permițând reprezentarea unor aspecte lexicale specifice unui stil de interacțiune. În al doilea rând, ETAG introduce nivelurile activitate și conceptual, care realizează legătura cu analiza sarcinii, utilizând o notație care permite reprezentarea entităților și a relațiilor dintre acestea.

De fapt, deși Payne și Green critică notația BNF, se poate considera că TAG este o evoluție a acestei notații, prin introducerea unor elemente de calificare a termenilor din BNF, astfel încât producțiile să poată reprezenta seturi. Totodată, TAG aduce un plus de formalizare, care previne omiterea arbitrară a unor valori ale trăsăturilor semantice din universul de cunoștințe ale utilizatorului. În mod similar, ETAG este o evoluție a TAG, atât în direcția competenței utilizatorului, întrucât aduce în plus concepte, atribute, relații și evenimente asociate sarcinii, cât și în direcția formalizării, introducând nivelurile referință și lexical.

Pentru o analiză mai detaliată, este utilă compararea reprezentărilor limbajului de interacțiune

cu ajutorul acestor trei notații. De exemplu, pentru o interfață grafică, reprezentarea funcțiilor de scalare și de mutare a unui obiect grafic utilizând notația BNF se face astfel:

Notația BNF

```
scal-ob ::= sel-fun + sel-ob + spec-scala
sel-fun ::= KEY "S" + KEY "Enter"
sel-cu-mouse ::= DEPLASEAZAMOUSE + CLICK
sel-ob ::= sel-cu-mouse
spec-scala ::= KEY-factor-scală

mută-ob ::= sel-fun + sel-ob + spec-fp + spec-tp
sel-fun ::= KEY "M" + KEY "Enter"
sel-cu-mouse ::= DEPLASEAZAMOUSE + CLICK
sel-ob ::= sel-cu-mouse
spec-fp ::= sel-cu-mouse
spec-tp ::= sel-cu-mouse
```

În această descriere, sunt reprezentate numai acțiunile utilizatorului. Aplicarea metricii propuse de Reisner [16] arată că, pentru prima metodă, sunt necesare 5 reguli și 5 terminale, iar pentru cea de a doua - 4 reguli (cu formă diferită) și 4 terminale. Pentru ambele funcții sunt necesare 5 reguli și 6 terminale, ceea ce denotă o consistență a celor două funcții în raport cu metodele de interacțiune folosite.

Trebuie menționat însă că notația BNF nu reflectă în mod direct legătura dintre cele două funcții care folosesc aceleași tehnici de interacțiune: selecția funcției prin accelerator sau cu ajutorul mouse-ului, selecția obiectului prin interceptie grafică și selecția punctelor de referință, respectiv translație, cu ajutorul mouse-ului. Această deficiență se datorează simplității notației, care nu permite reprezentarea cunoștințelor pe care le are utilizatorul despre interfață decât la un nivel foarte general. Din acest motiv, o evaluare a ușurinței în învățare a metodei pe baza numărului de reguli, a lungimii neterminalelor și a numărului de terminale este relevantă numai la un nivel general (în exemplul considerat, la nivelul ambelor funcții).

Același exemplu, utilizând notația TAG, se reprezintă astfel:

Dicționarul sarcinilor

```
scal-ob {funcție=S}
mută-ob {funcție=M}
```

Schemele regulilor

```
sarcină[funcție] → nume-fun[funcție] + sel-ob + spec-fp[funcție] + spec-tp[funcție] + spec-sc[funcție]
```

1. nume-fun[funcție=S] → "S" + "Enter"
2. nume-fun[funcție=M] → "M" + "Enter"
3. sel-cu-mouse → DEPLASEAZA-MOUSE + CLICK
4. sel-ob → sel-cu-mouse
5. spec-fp[funcție=S] → NULL
6. spec-fp[funcție=M] → sel-cu-mouse
7. spec-tp[funcție=S] → NULL
8. spec-tp[funcție=M] → sel-cu-mouse
9. spec-sc[funcție=S] → "factor scală"
10. spec-sc[funcție=M] → NULL

Acest exemplu arată capabilitatea notației TAG de

a reprezenta, în mod explicit, elemente de competență a utilizatorului. Prin modul în care rescrie simbolurile neterminale, gramatica propusă de Payne și Green capturează elementele de consistență ale limbajului. Se observă că este necesară o singură schemă pentru ambele sarcini elementare (regula 1), definită pe baza trăsăturii "funcție".

Cu toate acestea, în cazul selecției cu mouse-ul a funcției forma regulii este aceeași deși funcția este diferită. Notăția propusă de autori pentru aceste situații, permite o reprezentare alternativă a regulilor 2 și 3:

sel-cu-mouse → acțiune[tip=DEPLASEAZA-MOUSE, pozție=loc-dinobiectiv]+CLICK

Trebuie însă remarcat că nu este același lucru selectarea unui punct, a cărui poziție depinde de obiectivul urmărit cu selectarea unui articol din meniu. În al doilea caz, este vorba de o altă comandă, deci, un element care trebuie luat în considerație în estimarea dificultății în învățare. Aceasta este o deficiență a notației, care are la baza caracterul general al conceptului de trăsătură semantică. O reprezentare a universului utilizatorului necesită o conceptualizare mai detaliată, care să diferențieze concepte, evenimente, atribute și relații.

O altă deficiență a ambelor notații este dificultatea de a reprezenta stilul de interacțiune. Spre deosebire de TAG, BNF admite reprezentarea alternativei. De exemplu, funcția poate fi selectată prin tastare sau cu ajutorul mouse-ului:

sel-fun:=KEY"M"+KEY"Enter" | *sel-cu-mouse*

Evaluarea a două tehnici de interacțiune, fiecare bazată pe alt stil de interacțiune necesită însă scrierea a două metode.

Utilizarea notației furnizate de ETAG pentru exemplul considerat conduce la următoarea descriere:

```
UVM
type [OBJECT>FEREASTRA]
  themes: [OB];
  attributes<XWmin>,<XWmax>,<YWmin>,<YWmax>;
end [FEREASTRA].
type [OBJECT>OB]
  relations: [loc-pe([OB], [FEREASTRA])];
  attributes:<SCALA><POZITIE>;
end [OB].
type <ATTRIBUTE>SCALA>
  object type: [OB]
  value set: {*s@1-3}
end<SCALA>.
type <ATTRIBUTE>POZITIE>
  object type: [OB]
  value set: {(*x@XWmin-XWmax0),(*y@YWmin-YWmax)}
end <POZITIE>.
type [EVENT>MUTA-OB]
  description: for {[OB:*n]}
    event.SET POZ([OB],<POZITIE>,(x1,y1));
  precondition: [loc-pe([OB], [FEREASTRA])];
```

```
postcondition: [arepoz([OB],[POZITIE] ,(x1,x2));
end [MUTA>OB].
type [EVENT>SCAL-OB]
  description: for {[OB:*n]}
    event.SET-VAL([OB],<SCALA>"sc");
  precondition: [loc-pe([OB], [FEREASTRA])];
  postcondition: [are-scala([OB],[SCALA],"sc1")];
end [MUTA>OB].
```

DBT

Entry 1:

```
[TASK>MUTA-OB]
[EVENT>MUTA-OB]
[OB]
T1[EVENT>MUTA-OB][OBJECT>OB]<ATTRI>
```

Entry 2:

```
[TASK>SCAL-OB]
[EVENT>SCAL-OB]
[OB]
T2[EVENT>SCAL-OB][OBJECT>OB]<ATTRIBUTE>SCALA>
```

Specificatie

```
T1[EVENT>MUTA-OB]
[OBJECT=OB]:=specify[EVENT]+specify[OBJECT>OB]+specify<ATTRIBUTE>POZITIE>
```

Referință

a) cu accelerator

```
specify[EVENT]::=name 'symbol[EVENT]'
```

b) cu mouse

```
specify[EVENT]::=select 'symbol[EVENT]'
```

Lexical

a) symbol [EVENT>MUTA-

```
OB]::=[%COMMAND%:"M"]
```

b) symbol [EVENT>MUTA-OB]::=[ART-MENIU:"M"]

Fizic

```
a)name[%COMMAND%:"M"]::=KEY["M"]+KEY["Enter"]
```

```
b)select[%ART-MENIU%:"M"]::=DEPLASEAZA-MOUSE[%ART-MENIU%:"M"]+CLICK
```

Ceea ce aduce în plus ETAG este specificarea explicită a tuturor elementelor care intervin în interacțiune, fapt care conferă o putere de descriere mult mai mare. Descrierea sarcinii se face pe mai multe niveluri, în plus față de TAG fiind nivelul conceptual (mașina virtuală a utilizatorului - UVM) și introducerea nivelului referință în cadrul regulilor de producție, cu care se poate descrie stilul de interacțiune. Trebuie însă remarcat că acest plus în reprezentarea limbajului sarcinii, așa cum este perceput de către utilizator, se face cu prețul unui efort de specificare. Acest efort nu se justifică decât în măsura în care reprezintă un aport al etapei de analiză a sarcinii.

Se observă că niciuna dintre cele trei reprezentări ale limbajului sarcinii care sunt bazate pe o gramatică sarcină-acțiune nu permite descrierea procedurilor. De

exemplu, în cazul considerat este utilă aplicarea funcției selectate la o succesiune de obiecte. O asemenea facilitare se poate reprezenta cu ajutorul notației NGOMSL.

metoda-pentru-obiectiv: sarcina

1. *îndeplinește-obiectiv: sel-fun*
2. *îndeplinește-obiectiv: sel-ob*
3. *îndeplinește-obiectiv: spec-fp*
4. *îndeplinește-obiectiv: spec-tp*
5. *decide-obiectiv[continuare: da, nu]*
6. *IF[continuare=da] THEN tastează "Esc" ELSE goto 2*
7. *Return cu obiectiv îndeplinit*

Aceasta arată capacitățile de reprezentare a dinamicii interacțiunii de către o notație orientată spre procesul de execuție a sarcinii de lucru.

7. Aspectele cognitive modelate

Din punctul de vedere al aspectelor cognitive modelate, interesează două aspecte: limbajul de interacțiune, care reflectă cunoștințele de care are nevoie utilizatorul pentru descompunerea sarcinilor de lucru în acțiuni și scenariile de interacțiune, prin care se proceduralizează această descompunere.

Gramaticile sarcină-acțiune oferă posibilitatea de descompunere a sarcinii de lucru într-o specificație de acțiuni care reflectă, într-o măsură mai mică sau mai mare, reprezentarea mentală a acestei operații pe care o are utilizatorul. Modelele cognitive din această categorie caută să confere interacțiunii un anumit grad de validitate psihologică, raportându-se la cunoștințele pe care le are utilizatorul despre aplicație și la modul în care trebuie delegată o sarcină de lucru calculatorului.

Introducerea unor elemente care țin de reprezentarea cunoștințelor în memoria umană, cum sunt trăsăturile semantice din TAG, permite o specificație mai apropiată de cunoștințele pe care le posedă utilizatorul. Pe această bază, se poate estima mult mai eficient complexitatea unei structuri de interacțiune, care determină ușurința în învățare și în operare.

Următorul pas în direcția raportării la cunoștințele utilizatorului este făcut de ETAG, care extinde baza conceptuală la o structură mai elaborată, distingând obiecte, atribute, relații, evenimente și stări. În acest sens, ETAG se apropie de modelul de reprezentare a cunoștințelor pe baza cadrelor (frames). Totodată, la nivelul lexical, reprezentarea ETAG cuprinde elementele de prezentare asociate (meniuri, articole, iconițe), fapt care permite o specificație completă a limbajului de interacțiune. În acest fel, este posibilă o definire, la nivel cognitiv, a cerințelor pentru interfața de prezentare, utilă pentru etapa următoare de dezvoltare.

Un al doilea aspect modelat de gramaticile sarcină-acțiune este completitudinea limbajului de interacțiune. Specificația TAG permite evaluarea consistenței semantice și alinierii semantic-sintactice prin care valori ale trăsăturilor care sunt semnificative pentru utilizator își găsesc un corespondent în

structura producțiilor gramaticii. În acest fel, se evită omiterea arbitrară a unor comenzi și se ușurează procesul de învățare, permițând utilizatorului să determine prin inferențiere unele comenzi pe baza altora.

Un plus de completitudine este adus de ETAG a cărei notație oferă posibilități mai mari de specificare a variabilelor. Spre deosebire de TAG care a introdus variabilele numai pentru specificarea unor valori rezultate din sarcina curentă, ETAG extinde această facilitare la nivelul obiectelor și al evenimentelor. Acesta este un aspect important pentru proiectant, care poate prelua din specificație marje de valori admisibile și condiții de validare.

Un aspect care nu poate fi analizat pe baza modelelor de competență este modul cum este executată de fapt o sarcină. Distanța are în vedere, pe de o parte, succesiunea unor secvențe de interacțiune, ilustrată de exemplul prezentat în secțiunea precedentă, și, pe de altă parte, complexitatea cognitivă, privită în dinamică. În acest scop, este utilă o specificație orientată pe procedură, cum este NGOMSL.

Capabilitatea de reprezentare a unor scenarii de interacțiune este importantă în analiza sarcinii, pentru descompunerea unor sarcini complexe în sarcini elementare. Spre deosebire de gramaticile sarcină-acțiune, care pornesc de la acest nivel de sarcină elementară, referită ca sarcină simplă de Payne și Green [15] respectiv sarcină de bază, de Tauber [18], descompunerea GOMS nu are restricții, reprezentarea unor structuri arborescente fiind chiar un obiectiv al modelului. În plus, notația mai flexibilă adoptată de NGOMSL permite descrierea principalelor structuri de control din programare, fiind deci capabilă să reprezinte, la nivelul de detaliere dorit, un program mental.

Din acest motiv, NGOMSL permite o analiză cognitivă a dinamicii interacțiunii om-calculator, cu posibilitatea de a estima complexitatea sarcinii, gradul de încărcare a memoriei de lucru (a utilizatorului), frecvența apelurilor la memoria de durată, timpul de învățare și timpul de execuție a sarcinii. Spre deosebire de TAG și ETAG, care se bazează pe modele de reprezentare a cunoștințelor, NGOMSL se bazează pe o arhitectură cognitivă de procesare a informației, ceea ce permite predicții privind parametrii de desfășurare a interacțiunii în funcție de nivelul de instruire al operatorului uman.

8. Concluzii

Obiectivul acestui articol a fost evaluarea potențialului de aplicabilitate a tehnicilor și metodelor de interacțiune adecvate în scopul de a facilita constituie selectarea celor mai adecvate în analiza și în proiectarea sistemelor interactive. Acest demers este justificat de existența unui număr mare de abordări apărute în contextul unor aplicații diverse și utilizate în diferite etape ale ciclului de viață al unui produs informatic.

Principalele concluzii care se desprind din acest studiu sunt următoarele:

1. Evoluția modelelor cognitive din ultimii 15 ani reflectă atât obiectivul de a extinde aria de aplicabilitate în planul reprezentării mentale a cunoștințelor și proceselor cognitive ale utilizatorului, cât și dorința de a le face mai utile, prin transformarea acestora în metode de analiză și proiectare a interacțiunii.
2. Pentru definirea unei metodologii cu caracter practic, este necesară o evaluare a caracteristicilor pe care le au modelele cognitive / limbajele de reprezentare, din trei perspective: nivelurile de abstractizare utilizate, caracteristicile notației formale a limbajului de interacțiune și aspectele cognitive modelate.
3. Analiza potențialului de aplicabilitate identifică o complementaritate între două metodologii: ETAG și NGOMSL bazate, fiecare, pe alt tip de notație și având ca direcții prioritare de investigație competența utilizatorului care operează cu un anumit tip de interfață respectiv execuția propriu-zisă a unei sarcini de lucru.

Bibliografie

1. **ABOWD, G., J. BOWEN, A. DIX, HARRISON, M. & TOOK, R.**: User interface languages: A survey of existing methods, Technical Report, University of York, 1989.
2. **BARNARD, P.**: Cognitive resources and the learning of human-computer dialogues. În: J.M.Carroll (Ed.), *Interfacing Thought: Cognitive Aspects of Human-Computer Interaction*, MIT Press, Cambridge MA., 1987.
3. **CARD, S. K., T. P. MORAN & A. NEWELL**: The psychology of Human-Computer Interaction, Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1983.
4. **CARROLL, J. M., W. A. KERLOG & N.B.ROSSON**: The Task - Artefact Cycle. În: J.M.Carroll (Ed.), *Designing Interaction - Psychology at the Human-Computer Interface*, Cambridge University Press., 1991.
5. **CARROLL, J. M. & REITMAN, O.**: Mental models in Human Computer Interaction. M.Helander, *Handbook of Human-Computer Interaction*, North Holland, Amsterdam, 1988, pp.45-65.
6. **DIX, A., J. FINLAY, G. ABOWD & R. BEALE** Human-Computer Interaction, Prentice Hall, 1993.
7. **HAAN, G. DE, G. C. VAN DER VEER & J. C. VAN VLIET**: Formal modelling techniques in HCI, *Acta Psychologică*, 78, 1991.
8. **JOHNSON, P., K. DRAKE & S. WILSON**: A Framework for Integrating UIMS and User Task Models in the Design of User Interfaces. În: D.A. Duce, M.R. Gomes, F.R.A. Hopgood, J.R.Lee (eds.) *User Interfaces Management and Design*, Springer Verlag, 1991.
9. **KIERAS, D. & P. G. POLSON**: An approach to the formal analysis of user complexity. În: *International Journal of Man-Machine Studies*, 1985, 22, pp. 365-394.
10. **KIERAS, D.**: Towards a practical GOMS model methodology for user interface design", În: M.Helander, *Handbook of Human-Computer Interaction*, 45-65, North Holland, Amsterdam, 1988.
11. **KIERAS, D.**: A Guide to GOMS Task Analysis, UM, Spring, 1994.
12. **KNOWLES, C.** Can CCT produce a measure of system usability. În: D.M.Jones & R.Winder (Eds.), *People and Computers IV*, Cambridge University Press, 1988.
13. **MORAN, T.**: The command language grammar: a representation for the user interface of interactive computer systems. *International Journal of Man-Machine Studies*, 15, 1981.
14. **NIELSEN, J.**: A virtual protocol for computer-human interaction. În: *International Journal of Man-Machine Studies*, 1986, 24, pp.301-312.
15. **PAYNE, S. J. & T. R. G. GREEN**: Task action grammars: a model of the mental representation of task languages, *Human-Computer Interaction*, 1986, 2, pp.93-133.
16. **REISNER, P.**: Formal grammars and the design of an interactive system. În: *IEEE Transaction on Software Engineering*, 1981, 7, pp.229-240.
17. **SIMON, T.**: Analysing the scope of cognitive models in Human-Computer Interaction: A Trade-off Approach. În: D.M.Jones și R.Winder (Eds.), *People and Computers IV*, Cambridge University Press, 1988.
18. **TAUBER, M. J.**: ETAG: Extended Task Action Grammar - a language for the description of the user's task language. În: D.Diaper et al (Eds) *Human - Computer Interaction - INTERACT 90*, Elsevier Science Publishers B.V., North-Holland), 1990.
19. **YOUNG, R. M., T. R. G. GREEN & T. SIMON**: Programmable user models for predictive evaluation of user interface design. În: K. Bice & C. Lewis (Eds.), *Proc. of CHI'89: Human factors in Computing Systems*. ACM Press, New York, 1989.