

## Flexibilitatea programelor informatice – o solutie pentru înlaturarea dificultatilor cu care se confrunta utilizatorii

Lect.dr. Adriana SCHIOPOIU-BURLEA  
Facultatea de Stiinte Economice, Universitatea din Craiova

*Un program informatic "inteligent" trebuie sa dispuna de capacitatea de a propune raspunsuri pertinente într-un mod autoadaptativ, ceea ce presupune un nivel de automatizare ridicat. Ideea principala este ca un control automat al interactiunilor realizate cu sistemul poate sa genereze raspunsuri corecte, iar achizitionarea datelor sa se obtina printr-o analiza a actiunilor pe dispozitivele de intrare. Fiecare utilizator se confrunta cu dificultati particulare, generate de propriile reprezentari ale actiunilor si, de aceea, un program "inteligent" trebuie sa permita rezolvarea dificultatilor prin adaptare. Programul trebuie sa poata furniza o informatie tinând cont de contextul, de gradul de complexitate a problemei sau de precizarea unei cereri.*

**Cuvinte cheie:** *flexibilitate interindividuala, flexibilitate intraindividuala, logica formala, logica naturala, dificultati particulare.*

**U**n numar important de esecuri în informatizarea întreprinderilor sunt datorate, în mare parte, neadaptarii la logica relatiei utilizator-calculator si sunt generate de:

- \* *serviciile* care necesita gestionarea în paralel a informatiilor atât pe hârtie, cât si pe calculator, sunt trecute în totalitate pe calculator;
- \* *lipsa de timp* si/sau nerealizarea nivelului propus al productivitatii muncii;
- \* *gradul redus de utilizare* a posibilitatilor oferite de calculator;
- \* *erorile de utilizare*, care paralizeaza sistemul pe parcursul a mai multor zile sau chiar saptamâni.

Diferitele categorii de utilizatori nu au ace-easi logica, deoarece, pe de o parte, *informaticienii* doresc sa respecte *logica formala informatica* (precisa, binara, care sa excluda ambiguitatile) iar pe de alta parte, *operatorii* au o *logica naturala* (bazata pe subiectivitate, creativitate euristica).

Erorile sistematice pot sa releve un conflict de logica între o aplicatie conversationala si reprezentarea pe care o are utilizatorul. O

aplicatie conversationala are în comun cu o aplicatie generala, faptul ca defineste o structura a datelor si o structura a tratamentului asociat acestora. Cea mai mare diferenta consta în modul de concepere a relatiei practice om-calculator, relatie care sa permita comunicarea utilizatorului cu ansamblul practic - "prelucrare - date". Consecintele pot sa fie minimizezate prin urmatoarele masuri:

- *detectarea surselor* cât mai repede posibil;
- *posibilitatea de revenire*, în mod simplu, la o situatie anterioara;
- *definirea "blocului de validare"*;
- *explicatiile practice* în limbajul utilizatorului;
- *pozitionarea cursorului pe eroarea detectata*;
- *informarea utilizatorului despre erorile care se pot corecta* în mod automat.

Progresele tehnologice au condus la o diversificare a aplicatiilor informatice, ceea ce a implicat aparitia unor necesitati noi legate de:

- integrarea aplicatiilor la acelasi post de lucru;

- adaptabilitatea interfetei în functie de modelele de utilizatori (novici sau experti);
- posibilitatea de întrerupere a activitatii, pentru a efectua alte tranzactii sau pentru a reveni la contextul initial.

Numeroase erori pot fi evitate, în diferite faze de concepie a unui program, daca se observa la timp cauzele ce le genereaza :

- *confundarea a doua functionalitati ale aceluiasi post de lucru* (pentru a se realiza operatia de stergere a unor date se pot utiliza tastele DEL sau RESET, fiecare dintre aceste taste având un regim de utilizare diferit);
- *aparitia erorilor de codificare sau de decodificare*;
- *ignorarea unei proceduri si încercarea de utilizare a diverselor actiuni "euristice de incompetenta"*;
- *utilizarea procedurilor complexe, în mod inutil*;
- *actiuni inoportune* generate de timpul de raspuns foarte lung sau aleatoriu.

Activitatea de optimizare a utilizarii programelor se desfasoara pe patru planuri:

- *definirea functionalitatilor utile* pe care programul trebuie sa le ofere, în functie de tipul sarcinii si de tipul utilizatorului;
- *adaptarea programului la reprezentarile mentale ale utilizatorului*;
- *alegerea modalitatilor de dialog* utilizator-calculator;
- *optimizarea codificarii informatiei*, atât pentru prezentare cât si pentru comenzi.

Necesitatile utilizatorului nu pot fi rezolvate doar prin înregistrarea unor progrese tehnologice, deoarece este stiut faptul ca fiecare inovatie conduce la aparitia unor noi probleme. Cea mai buna maniera în care se poate tine cont de utilizator nu este aceea de a reflecta simplu doar locul sau de munca, ci si metodele de analiza, care trebuie sa se extinda asupra comportamentului cognitiv al acestuia.

Utilizatorii sunt caracterizati de urmatoarele variabile:

- *experienta în utilizarea instrumentului*;
- *experienta în realizarea activitatii*;
- *motivatie* în utilizarea instrumentului;
- *îndemânare* în domeniul informatic;
- *caracterul ocazional sau permanent* al utilizarii;
- posibilitati de *acomodare*;
- *limite psihologice si fiziologice* legate de capacitatea senzoriala, postura, rezistenta la oboseala, stress.

Principalele elemente si etape de proiectare a unui sistem informatizat, demne de luat în considerare, chiar la elaborarea unui element aparent izolat, sunt:

- proiectarea cerintelor sistemului complex, a conditiilor de functionare si a performantelor;
- delimitarea locului utilizatorului în sistemul proiectat si precizarea functiilor umane, fiind conturat rolul functional al omului în detectarea, identificarea, stocarea, interpretarea si valorificarea informatiilor;
- alocarea functiilor între utilizator si calculator, tinându-se cont de posibilitatile si limitele componentelor sistemului, de factorii motivationali, de alternativele posibile;
- precizarea mijloacelor tehnice de selectie si de pregatire a utilizatorilor (prin probe practice de selectie, învatamânt profesional programat, utilizarea de modele si metode matematice).

Activitatea unui operator poate sa fie considerabil simplificata prin modificarea modului de prezentare a informatiilor pe ecran<sup>1</sup>.

Un program inteligent trebuie sa ofere utilizatorului posibilitatea de:

- evaluare a solutiilor vizate pentru rezolvarea unei probleme;

---

<sup>1</sup> Modul de dispunere a informatiilor pe ecran trebuie sa fie optimizat în functie de ordinea pe care utilizatorul o foloseste frecvent pentru prelevarea informatiei.

- testare a ipotezelor, plecând de la simulări locale, prin întreruperea temporară a unei activități principale;

- luarea în considerare a anumitor activități;

- utilizarea de ghiduri, sfaturi, recomandări.

Un program cu o flexibilitate interindividuală poate să fie utilizat de către subiecții care au cunoștințe relative în domeniu și strategii de lucru personale.

Satisfacerea exigențelor legate de flexibilitatea interindividuală necesită conceperea unui sistem personalizat, ceea ce limitează aproximările statistice care conduc la elaborarea unor dispozitive destinate satisfacerii majorității nevoilor unei populații eterogene.

Nevoile utilizatorului evoluează pe parcursul desfășurării practicii și, de aceea, sprijinul acordat debutantului într-un anumit domeniu este diferit de ceea ce va avea nevoie atunci când va fi experimentat. Mai mult, pentru utilizarea unui dispozitiv, dificultățile întâlnite în primele tranzacții și cele care continuă (sau apar) la sfârșitul celor 6 luni de practică nu sunt de aceeași natură. Un sistem trebuie

poate prezenta afișaje adaptate la nevoile individului, în diferite stadii ale practicii sale; satisfacerea exigenței de flexibilitate intraindividuală necesită conceperea unor dispozitive evolutive.

Mesajele de informare, planurile de acțiune, explicațiile vor fi prezentate într-un mod detaliat pentru un utilizator debutant și sub formă laconică pentru un utilizator experimentat.

Datele referitoare la utilizator și problema întâlnită, trebuie recepționate și puse în relație cu un model al utilizatorului, care să permită, între altele, definirea la nivel de detaliu a răspunsului.

Programul de asistență trebuie să fie conceput ca un observator exterior atent, un supervisor care să trateze, într-un mod "inteligent", evenimentele care îl controlează. Un asemenea program trebuie să fie capabil să detecteze cauzele unei dificultăți și să elaboreze un diagnostic.

Principiile de elaborare a unui răspuns pertinent sunt reprezentate în figura nr. 1.

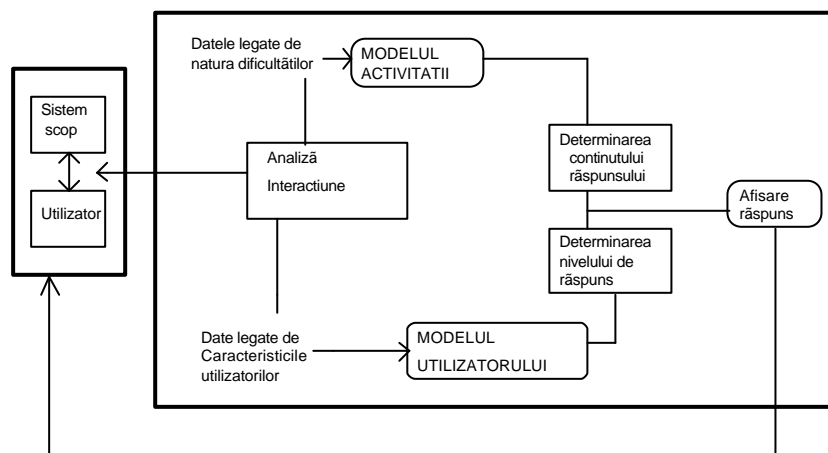


Fig. 1. Principii de elaborare a unui răspuns adecvat

Cea mai mare parte a sistemelor de ajutor nu comportă un model explicit care să permită prelucrarea datelor. De exemplu, unele nu analizează natura dificultăților întâlnite, iar altele nu comportă modele ale activității și

conținutul răspunsurilor afișate este predeterminat și independent de dificultatea efectivă.

În absența unui diagnostic al dificultăților, pertinenta unui afișaj nu poate să fie decât pur

circumstantiala: este tipic cazul sistemelor de informatică clasice care afișează, la cererea utilizatorului, texte preînregistrate. Aceste afișaje sunt în puține cazuri satisfăcătoare, deoarece nu prezintă, în general, decât un tip de informație și acesta este adesea cunoscut fie foarte puțin, fie foarte detaliat. Aceste dificultăți au fost subliniate în [Jackson, Lefrere, 1984], referitoare la un sistem clasic sofisticat, "CADHELP", în care subiectul a fost ales din trei tipuri de aplicații (un rezumat care prezintă explicații generale arbitrare, o explicație detaliată și un catalog care are ca obiectiv prevenirea erorilor posibile care apar în urma utilizării unei funcții).

Adaptarea unui răspuns ridică o dublă problemă de definire a dificultăților de utilizare și a soluțiilor lor specifice. În prezent caracteristicile afișajelor (natura explicației, nivelul abstracției) sunt definite plecând de la principii generale care permit îmbunătățirea pertinentei răspunsului. Se admite faptul potrivit căruia conținutul ajutorului trebuie să fie îmbogățit și să fie în măsură să găsească singur soluția problemei (trebuie să se prezinte, de exemplu, informațiile care să permită completarea cunoștințelor utilizatorului despre funcționarea sistemului).

Tehnicile folosite constau în generalizarea problemelor utilizatorului, în furnizarea unei descrieri detaliate a efectului comenzilor, a condițiilor specifice de execuție și a efectelor de margine esențiale. Cu toate acestea, aplicarea acestor principii nu garantează ca fiind necesară adaptarea la răspuns, aceasta necesitând un veritabil diagnostic al dificultăților de utilizare.

Analiza unei dificultăți poate să fie abordată pe două niveluri:

- *un prim nivel de diagnostic care este asigurat printr-un caz particular* este pus în legătură cu o clasă de dificultate cunoscută. De exemplu, procedurile puse în joc de utilizator pot comporta acțiuni inegale. Afișarea clasificării permite caracterizarea modului operațional (inversarea operațiilor, omiterea operațiilor, condiții de realizare nefolosite).

- *un al doilea nivel de diagnostic* constă în elaborarea unei interpretări cauzale a acestor dificultăți, adică ținându-se cont de cauzele apariției lor. Aceasta interpretare privește structurile de cunoștințe puse în joc în situația unei activități cognitive, care permite determinarea proceselor responsabile de comportamentele neadecvate.

O clasificare a erorilor care pot să apară în relația utilizator-calculator este prezentată în tabelul 1.

Informatizarea implică creșterea complexității sistemului și, în consecință, probabilitatea ca la proiectarea și exploatarea acestuia să intervină erori, în loc ca acestea să se reducă. Trebuie menționat faptul că activitatea desfășurată poate avea efecte atât asupra organizării muncii, cât și asupra relațiilor umane și a climatului de muncă. Aceste influențe se află într-o relație de interacțiune reciprocă, în sensul că anumite consecințe negative pentru individ devin condiții nocive de muncă, degradând performanța și fiabilitatea utilizatorului.

**Tabelul 1.** Cauzele erorilor care pot să apară în informatizarea unei activități

EROARE	CAUZA
Defecțiuni în detectarea semnalelor	Supraîncărcare la intrare. Substimulare la intrare. Zgomot ambiental.
Identificare incorectă	Orientare inadecvată a semnalelor

Evaluare incorecta	Model conceptual inadecvat
Alegere incorecta a actiunii	
Defectiuni în efectuarea actiunii	Actiune incorecta la momentul potrivit sau actiune corecta la momentul nepotrivit

### Bibliografie

[Bisseret, Boutin, Michard, 1980]

Bisseret, A., Boutin, P., Michard, A., *Elements introductifs a l'ergonomie des systemes hommes-machines*, Informatique et sciences humaines, 1980.

[Burlea Schiopoiu, Rosca, 1998]

Burlea Schiopoiu A., Rosca D., *Tehnici informationale pentru managementul competentelor umane*, Editura CERTI, Craiova, 1998.

[Burlea Schiopoiu, 1999]

Burlea Schiopoiu A., *Relatia om – calculator. Consideratii ergonomice*, Editura CERTI, Craiova, 1999.

[Cohen, 1990]

Cohen J-C, *Les informateurs - ou comment maîtriser une information*, Les Editions D'Organisation, Paris, 1990.

[Coutaz, 1990]

Coutaz J., *Interfaces homme - ordinateur*, BORDAS, Paris, 1990.

[Jackson, Lefrere, 1984]

Jackson P., Lefrere P., *On the application of rule-based techniques to the design of advice giving system*, International journal of man-machine studies, 1984.