

TESTAREA INTERFEȚELOR OM - CALCULATOR

Prof. dr. Ion Ivan
Asist. univ. Cristian Toma

Facultatea de Cibernetică, Statistică și Informatică Economică,
Academia de Studii Economice, București

Rezumat: Comunicarea între om și calculator se realizează prin intermediul interfețelor. Cu cât acestea sunt proiectate mai bine, cu atât gradul de confort al utilizatorilor aplicațiilor crește. De asemenea, în cazul unor interfețe bine realizate, timpul de accesare al informațiilor scade, iar gradul de acomodare al utilizatorilor cu aplicația este mult mai bun.

Cuvinte cheie: interfețe om-calculator, interfețe grafice, testări ale interfețelor grafice.

1. Definirea problemei

O aplicație informatică, destinată unui număr mare de utilizatori, cetățeni ai unei localități, unui oraș, unei arii geografice, există sau nu prin interfața pe care o percepe fiecare dintre aceștia.

Dacă la elaborarea interfeței sunt luate în considerare o serie de elemente specifice, la testare se urmăresc două aspecte și anume:

- măsura în care conținutul specificațiilor de interfață a fost implementat și se regăsește în produsul finit;
- reacțiile utilizatorilor la traversarea succesiunii de pași necesari pentru soluționarea problemei.

Așa cum crearea de stocuri se efectuează pentru asigurarea continuității proceselor de producție, creându-se conceptul de „stocuri orientate pe producție”, tot astfel, dezvoltarea interfețelor este orientată exclusiv spre utilizatori.

În cazul în care se urmărește măsura în care conținutul specificațiilor de interfață a fost implementat și se regăsește în produsul finit, în testarea interfețelor, se execută următorii pași:

- identificarea mulțimii de mesaje pe care aplicația informatică le furnizează ca ieșiri;
- construirea vocabularului cuvintelor de intrare furnizate de utilizatori și program;
- definirea situațiilor în care dialogul este strict, adică datele de intrare trebuie să fie complete, în ordinea impusă, și a situațiilor în care dialogul nu este strict, adică unele mesaje pot lipsi sau cuvintele cheie au o dispunere oarecare;
- stabilirea tipurilor de probleme, împreună cu frecvențele lor de apariție, în vederea acordării unei atenții sporite situațiilor în care orice aplicație informatică este definită și realizată pentru a rezolva cel puțin o problemă.

Dacă aplicația A rezolvă problemele P_1, P_2, \dots, P_n , atunci pentru o problemă P_i sunt identificate m_i cazuri distincte, fiecărui caz C_{ij} corespunzându-i selecții de mesaje, respectiv, seturi de date D_{ij} cu structură diferită de a celorlalte cazuri.

Pentru fiecare caz în parte C_{ij} se obțin rezultate specifice R_{ij} , ca în figura 1:

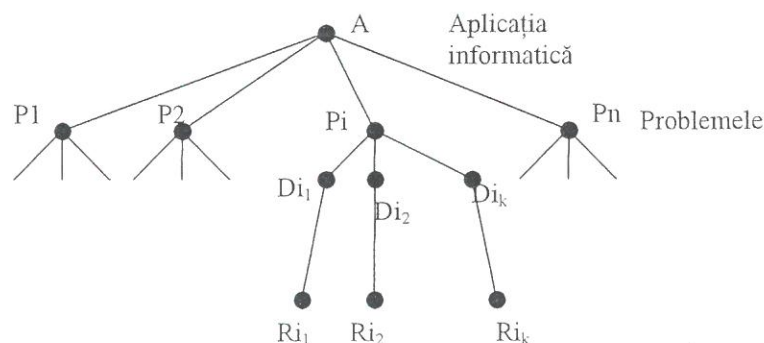


Figura 1: Structura arborescentă asociată interfeței aplicației informatică A

A testa o interfață a aplicației A revine la a defini fluxurile pe care le generează seturile de date asociate fiecărei probleme și a vedea concordanța între rezultatele efectiv obținute și rezultatele date în specificațiile interfeței.

Rezultatul testării este fie acceptarea interfeței ca fiind corespunzătoare cerințelor, fie respingerea ca nesatisfăcând specificațiile. Acceptarea este dată în situația în care peste 78% din eșantionul utilizatorului traversează complet pașii necesari obținerii rezultatului final, iar rezultatul final este complet și corect, tot peste 78% din cazuri.

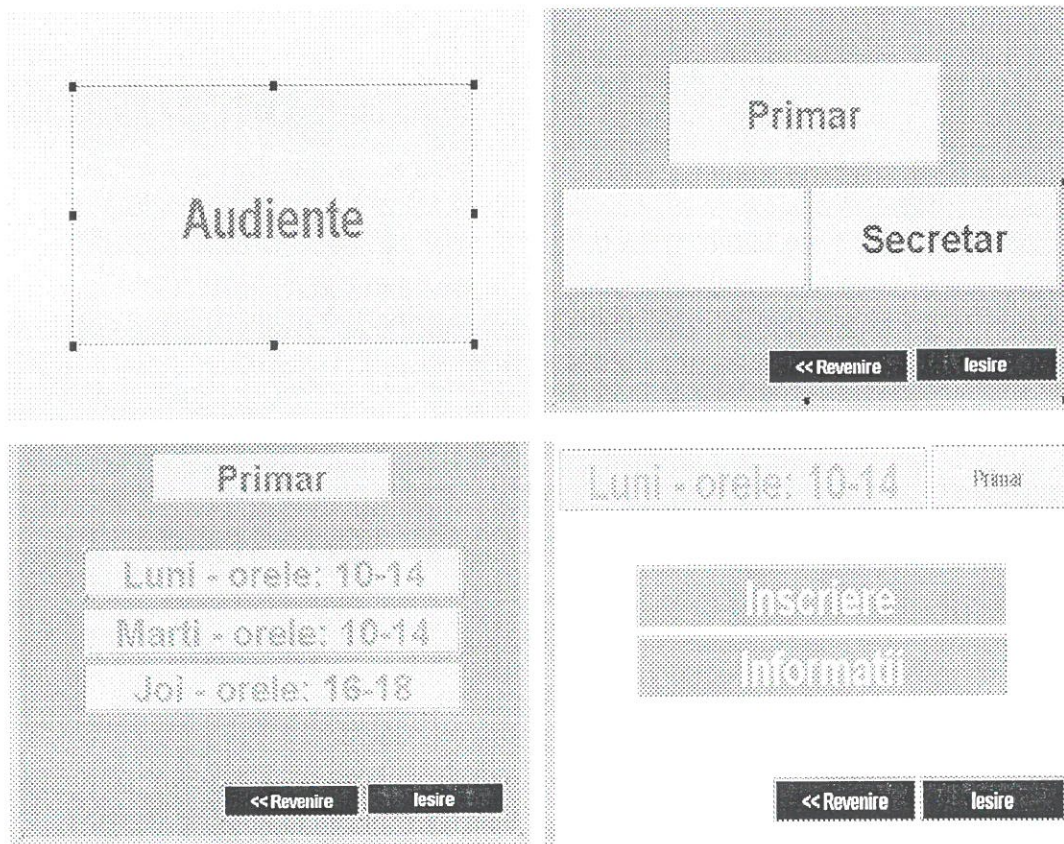
- diferențele găsite între specificații și comportamentul real al interfeței;
- cazurile care generează diferențele dintre specificații și comportamentul efectiv;
- identificatorii persoanelor care efectuează testarea;
- modalități de îndreptare a neconcordanțelor fie prin construirea de comportamente care cresc performanțele interfeței, fie prin modificări în modulele de prelucrare.

Testarea interfețelor este un proces complex, care conține elemente standardizate, iar persoana care trebuie să facă testarea trebuie să cunoască principiile de elaborare a interfețelor.

2. Interfețe cu selecție opțiuni

Se consideră o problemă P căreia i se asociază cuvinte dispuse pe niveluri, iar utilizatorul selectează succesiunea de cuvinte în ordinea impusă de cerințele situației pe care dorește să o rezolve.

Selecția se efectuează din aproape în aproape și se obține, în final, rezultatul. De exemplu, dacă se dorește înscrierea în audiență, interfața de selecție se definește:



La sfârșit, aplicația eliberează un tichet prin care utilizatorul este înștiințat că audiența este posibilă și, eventual, ora la care este așteptat.

A testa această interfață revine la:

- verificarea dacă utilizatorii pot delimita cu ușurință butoanele;
- stabilirea modului în care utilizatorii acordă priorități persoanelor, zilelor, orelor;
- studierea comportamentului utilizatorilor știut fiind faptul că atenția se focalizează asupra rezolvării problemei.

În cazul în care sunt ocupate resursele, interfața se modifică prin:

- eliminarea resursei epuizate;
- re poziționarea butoanelor corespunzătoare resurselor existente.

Se impune definirea unor acțiuni-evenimente sau butoane de validare a selecțiilor făcute, respectiv, de anulare. Dacă se produce validarea, se eliberează tichetul. Dacă se anulează selecția făcută, există posibilitatea reluării selecțiilor.

În acest caz, datele de test vizează o structură arborescentă, organizată pe patru niveluri, arborele fiind ilustrat în figura 2:

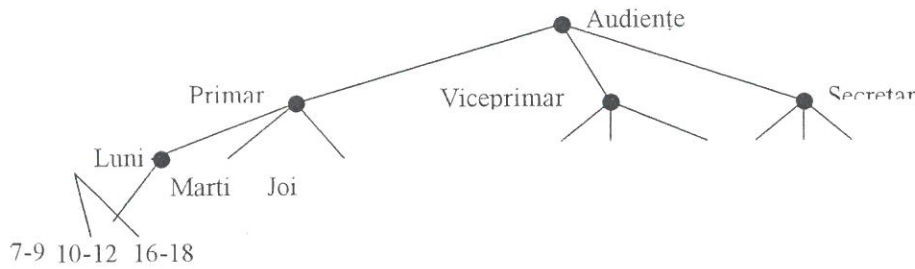


Figura 2: Arborele asociat interfeței pentru programare audiențe.

3. Interfețe cu introducere coduri

Codurile sunt necesare identificării celor care doresc să acceseze resurse specifice aplicației. Codurile sunt date sub formă de șiruri de caractere. În cazul în care se presupune accesul la resurse ale bazelor de date privind o persoană, un proprietar, un consumator, un abonat, evident, accesul trebuie să fie dat prin introducerea unei parole sau a codului numeric personal al persoanei. Deja, sunt proiectate aplicații care nu iau în considerare acest mod de referire, conducând la soluții eronate, întrucât generează coduri în plus, ceea ce duce la obligarea utilizatorului de a folosi alte coduri sau documente pentru a se autentifica. În cazul parolelor, coduri cu structură impusă, se urmărește o prezentare care să elimine orice ambiguitate, privind lungimea și, mai ales, structura. În cazul caracterelor eronate, acestea trebuie afișate. De exemplu, pentru codul numeric personal, se consideră structura:

“s aa ll zz xxx yyy” unde: s ∈ {1,2}; ll ∈ {1..12}; zz ∈ {1..31}; iar x,y ∈ {0..9}.

În cazul în care se tastează șirul: “a 01 0c 03 d43 38r”, întrucât caracterul a este eronat, va fi afișat, asemenea și caracterului c, d sau r în formatul: “a ** *c ** d** **r”. Utilizatorul are în față caracterele care au generat eroarea la validarea codului său numeric personal. În continuare, pentru accesul la resurse speciale, precum eliberările de numerar, cunoașterea unor date protejate, administratorul aplicației impune distribuirea de parole de acces, pe care utilizatorii trebuie să le introducă și numai după validarea lor se va produce accesul la resurse. Se impune realizarea unei interfețe ca în figura 3:

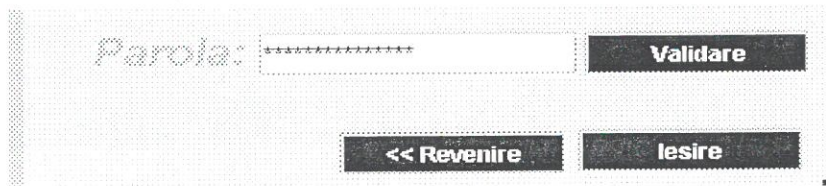


Figura 3: Interfață grafică pentru introducerea parolei

În primul rând, nu se specifică lungimea șirului de caractere care formează parola, în al doilea rând, nu se specifică la introducerea eventualele erori din șir, iar, în al treilea rând, dacă s-au introdus trei parole eronate, se produce blocarea accesului la resurse pentru respectivul utilizator. Deblocarea se realizează contra cost pentru a pune în ordine derularea fluxurilor în mod normal.

În cazul în care parola este corectă, se vizualizează resursele aplicației. De exemplu, se afișează situația impozitului pentru utilizatorul Gheorghe Popescu:



Figura 4: Interfață de vizualizare taxe și impozite

Trebuie avut în vedere, la fiecare pas, posibilitatea utilizatorului de a ieși din aplicație. Mai trebuie avut în atenție și numărul de utilizatori care folosesc aplicația, dacă sunt foarte mulți și se dorește degrevarea aplicației de procese concurente inutile, se setează o opțiune prin care utilizatorul, dacă nu folosește timp de cinci minute aplicația, este scos din program automat. Pentru identificarea corectă, după introducerea parolei, pe lângă la accesul la anumite resurse, se afișează codul numeric personal, numele și adresa pentru câteva secunde, după care dispar.

Acest tip de interfață este testat pentru a evidenția dacă:

- sunt acceptate șirurile de caractere corecte numai în raport cu o structură dată;
- se realizează un nivel de siguranță maxim în ceea ce privește protecția față de cei care sunt în jurul utilizatorului, deși se impun restricții de păstrare a unei distanțe minime.

În cazul în care accesul se efectuează din locuri izolate, al doilea punct de mai sus dispare.

4. Interfețe cu tastarea textelor

Numeroase aplicații impun introducerea de texte. Înscrierea la o formă de instruire, înscrierea la obținerea unei vize, înscrierea cu lucrări la simpozioane, încheierea de contracte, presupune introducerea de texte.

Restricțiile impuse sunt:

- lungime șir de caractere;
- utilizarea cuvintelor cheie dintr-un vocabular specificat;
- limba de prezentare;
- structurarea în subdiviziuni discrete pentru a descrie adrese și alte tipologii de coordonate;
- efectuarea unei unice înregistrări;
- existența de înregistrări cu un nivel de ortogonalitate ridicat.

A testa o interfață în care textele introduse sunt preponderente, revine la:

- analizarea comportamentului pentru texte sau șiruri de caractere video presupune că o interfață bine proiectată și construită semnalizează apariția șirurilor de lungime 0, adică a șirurilor vide, din construcție precizându-se situațiile în care se acceptă șiruri de caractere vide;
- stabilirea modului de reacție pentru introducerea de texte identice; în cazul în care se definesc șirurile s_1, s_2, \dots, s_n se impune stabilirea unui mod de răspuns în cazul în care două sau mai multe șiruri sunt identice;
- apartenența unui șir de caractere la o altă submulțime decât cea impusă prin enunț; există șiruri formate din caractere, cifre, iar, în cazul introducerii unor separatori, pentru creșterea flexibilității se definesc mecanisme de dezvoltare a șirurilor fundamentale prin normalizare în raport cu delimitatorii și separatorii pe care un utilizator îi introduce între cuvinte;
- utilizarea literelor mari sau a literelor mici, ajustând nivelul de sensibilitate al interfeței, în concordanță cu modul în care este definită interfața.

Interfețele care sunt bazate pe introducerea de texte trebuie să fie proiectate astfel încât:

- în primul pas să aibă loc introducerea de șiruri;
- în al doilea pas, trebuie să aibă loc validarea textelor;
- în al treilea pas, să se producă unele corecții; pentru efectuarea de corecții trebuie să se producă localizarea în una dintre mulțimile de șiruri și, după aceea, în cadrul fiecărui șir, să se efectueze actualizări – inserare, modificare, ștergere – de caractere;
- în al patrulea pas, se ia decizia de introducere în baza de date.

Repetabilitatea deciziei de introducere în baze de date este limitată la un număr impus. În cazul în care durata unei interacțiuni depășește un timp suficient de mare, este rezonabilă întreruperea dialogului. Este necesar ca o dată introduse datele, în caz de senalare erori, întregul serviciu, tot ce s-a executat până atunci, să fie anulat. Structura textelor care trebuie introduse este necesară a fi suficient de bine definită astfel încât elementele de ambiguitate să fie minime. Probleme de mare dificultate apar atunci când se produc alocări, rezervări sau înscrieri la competiții și, mai ales, se efectuează plăți și tranzacții în timp real și în mod concurrent.

Testarea interfețelor care operează cu șiruri de caractere este deosebit de importantă atunci când șirurile introduse, de lungimi oarecare, nu se regăsesc în mulțimile gestionate de bazele de date ale aplicației.

De exemplu, în cazul introducerii unui șir de caractere, corespunzător unei străzi al cărui nume este format din două cuvinte c_i și c_j , se introduce fie un șir format din c_i spațiu c_j , fie un șir format din c_j spațiu c_i . Sunt situații în care prenumele este prescurtat sau lipsește. Este important ca aplicația să aibă o bază de date a numelor de străzi, cu toate dificultățile ce apar din schimbările de denumiri și gestionarea proceselor de actualizare. În același fel sunt puse problemele pentru orașe, comune, localități. De asemenea, testarea interfețelor vizează comportamentul utilizării de șiruri de caractere, în care apar subșiruri formate din secvențe ale aceluiași șir.

În cazul interfețelor de acest tip, testarea vizează însăși capacitatea interfeței de a efectua corecții automate. De exemplu, la testarea șirului: „MMMMAAAARRRRRIAAA”, corecția automată revine la a elimina aparițiile repetate, obținându-se cuvântul: „MARIA”. Nu același lucru se întâmplă pentru cuvintele ce conțin secvențe cu două apariții ale unor litere, de exemplu, litera „o” în „cooperativ sau „c” în „accent”, „acces”.

5. Interfețe grafice

În realitate, toate interfețele conțin elemente de grafică. Interfețele identificate sub acest nume conțin exclusiv imagini, simboluri, nefiind necesare texte ca elemente principale de comunicare între om și computer.

Pentru a construi interfețe grafice, sunt necesare clarificări esențiale, privind obiectivul urmărit și mai ales mijloacele necesare atingerii obiectivului.

Un rol esențial în dezvoltarea și implementarea interfețelor grafice îl au modalitățile de selecție a reprezentărilor grafice fie direct prin presarea ecranului, fie prin intermediul mouse-ului.

Testarea unei interfețe grafice vizează:

- caracterul sugestiv al simbolurilor și imaginilor utilizate;
- modul de dispunere a elementelor de grafică în vederea unei identificări rapide și fără un efort special;
- delimitările dintre ariile asociate fiecărui simbol/imagini, în așa fel încât să nu fie implicate zone de neutralitate care sunt însoțite fie de inacțiune, fie de comportament nedeterminist;
- ierarhizarea în raport cu cerințele unei abordări naturale a problemei de soluționat.

Orice interfață grafică presupune un raționament în dezvoltarea tuturor elementelor necesare obținerii informațiilor dorite. În condițiile unei abordări corecte, o astfel de interfață are menirea de a deschide noi teritorii în ceea ce privește universalitatea unor echipamente. Folosind simboluri simple, consumatorul deslușește cum este utilizat corect un aparat electrocasnic. Mai ales, va afla ce nu trebuie să facă pentru ca respectivul aparat să nu devină periculos.

Construirea interfeței grafice permite utilizatorilor selectarea rapidă a ariilor de interes, prin captarea atenției de către imagini sau simboluri.

În cazul magazinelor virtuale obișnuite sau a magazinelor virtuale de artă, abordările grafice sunt esențiale, aducându-l pe cumpărător în situația de a selecta rapid tipul de produs și cantitatea.

Testarea unei interfețe grafice revine la a vedea dacă într-adevăr simbolurile utilizate sunt cele cerute prin specificații, fără a apare unele confuzii.

De asemenea, testarea unei interfețe grafice vizează gradul de satisfacție pe care îl obține utilizatorul, în raport cu problema sa particulară. În cele mai multe situații, se elaborează interfețe grafice pentru a oferi informații complete și corecte.

Pentru obținerea de trasee turistice, pentru obținerea informațiilor privind muzee, pentru ghidarea pe o hartă, pentru a obține un traseu între două puncte de pe hartă, pentru a alege cel mai apropiat hotel în raport cu poziția dată, se construiesc interfețe extrem de utile, pentru care nu trebuie utilizate limbi străine, ci numai câteva cuvinte uzuale.

A testa aceste interfețe revine la a prelucra combinații de puncte și a vedea traseele generate. Aceste trasee se verifică și, mai ales, se analizează pentru a le evidenția optimalitatea.

În cazul stabilirii punctelor de optim, în raport cu o poziție dată, interfața are la bază un algoritm de aflare a drumului minim cu luarea în discuție a punctelor din intervalul grafului.

Testarea interfeței revine la a verifica dacă optimul obținut prin calcule are corespondent în reprezentarea grafică, sub forma unei hărți. Interfețele grafice sunt dificil de testat având în vedere multitudinea de aspecte pe care le implică.

Este important ca, în procesul de testare, să se evidențieze gradul de ortogonalitate al simbolurilor grafice, a imaginilor. În cazul unei slabe ortogonalități, apar confuziile, cele care conduc la reluarea unor trasee eronate de către utilizator.

De asemenea, este important de stabilit care sunt simbolurile cărora li s-a atribuit o altă semnificație decât cea uzuală și să se măsoare efectul pe care schimbarea o determină asupra ratei de insucces, la atingerea obiectivelor utilizatorilor.

Interfețele grafice sunt caracterizate și printr-un grad de omogenitate a tehnologiei de reprezentare. Se utilizează figuri geometrice plane și regulate, conducând la o anumită structură de interfață. Se folosesc, pentru alte structuri, figuri neregulate cu amplasamente oarecare, neluându-se în considerare realizarea unei contribuții armonioase.

Oricare din modalități este utilizată, testare interfeței trebuie să evidențieze impactul pe care apariția fiecărei reprezentări îl are asupra utilizatorului, în sensul creșterii gradului de satisfacție în ceea ce privește traversarea pașilor necesari soluționării unei probleme particulare de către fiecare dintre utilizatori.

Interfața grafică presupune utilizarea culorilor ca mijloc de creștere a gradului de ortogonalitate a simbolurilor. Orice construcție grafică presupune componente individualizate, dispuse pe o suprafață sau în spațiu pe diferite planuri. Culoarea joacă un rol activ în evidențierea și diferențierea componentelor interfeței. Păstrând o culoare pentru o succesiune de simboluri, se asigură construirea unui lanț de pași care trebuie urmați consecutiv pentru a obține o soluție. Testarea interfeței grafice are menirea de a evidenția impactul pe care îl are culoarea asupra utilizatorilor și efectele de natură psihologică generate.

Raportul de testare trebuie să evidențieze riguros întregul ansamblu de măsuri ce trebuie întreprinse pentru a ameliora efectele negative.

6. Testarea totală

Fiecărei interfețe i se asociază o structură arborescentă care vizează multitudinea nivelurilor care trebuie atinse pentru a obține toate tipologiile de prelucrări pe care le generează aplicația informatică.

Testarea presupune:

- a vedea concordanța între specificații și produsul real care se lansează în uz curent;
- a stabili comportamentul utilizatorilor care doresc să soluționeze problema concretă.

Procesul de testare este total. Se utilizează generatoare de date de test, care evidențiază toate combinațiile de selecții care stau la dispoziția utilizatorilor. Testarea interfeței vizează derularea rând pe rând a secvențelor, consemnându-se rezultatul obținut, distinct dacă este un rezultat corect, față de rezultatul eronat.

În cazul în care raportul:

$K=A/B$ unde:

A = numărul traseelor cu rezultate corecte;

B = numărul total al traseelor testate;

îndeplinește condiția $K>0.78$ se concluzionează că interfața este bine structurată și se continuă perfecționarea ei, până când $K=1$. În caz contrar, efortul de ajustare pentru interfață devine costisitor și produsul software este la un pas de a fi compromis. Este deosebit de important să se considere că, într-un produs software, aproximativ 80% din resurse se alocă pentru asigurarea calității dialogului om-calculator prin asigurarea calității datelor de intrare.

Testarea interfeței, în raport cu cerințele unui lot reprezentativ de utilizatori, vine să completeze testarea tuturor variantelor de trasee. Se trece la testarea în condiții efective a interfeței. Există șansa de a vedea cauzele de activare fără succes ale aplicației. Se consideră N numărul de utilizatori ai lotului reprezentativ, N_i numărul activărilor soldate cu insuccese, datorate unor cauze C_1, C_2, \dots, C_m , cu $m \leq s$.

Pentru cauzele C_1, C_2, \dots, C_m , se află:

- ortogonalitate redusă a comenzilor;
- imposibilitatea de a selecta o opțiune;
- continuitatea redusă în raport cu alte aplicații;
- neadecvarea simbolurilor;
- lipsa facilităților de corecție și revenire la etapele precedente.

Testarea totală este costisitoare ca efort, însă, este obligatorie dacă se dorește obținerea unui produs performant. De exemplu, procedura de generare a calendarului și procedura de validare a datei calendaristice, o dată realizate, se utilizează ori de câte ori este necesar. Orice eroare în acest perimetru este cu efecte defavorabile dintre cele mai grave. De aceea, procedurile legate de dezvoltarea interfețelor trebuie să fie perfecte, absolut corecte, fiabile și portabile. Testarea totală a interfețelor elimină erorile procedurilor de calcul și permite

efectuarea de teste asupra acestora, fără a exista riscul erorilor combinate, generate de date de intrare eronate, strecurate în procesul de comunicare om-calculator și de erorile de prelucrare propriu-zisă.

Deși acum problema specializării celor care se ocupă de testare a interfețelor este la început, au apărut suficiente elemente care să determine o atenție sporită acestui aspect. Raportul de testare este suficient de amănunțit, indicând tipurile de erori, cauzele fiind stabilite de cei care depanează programele. Înregistrările includ și durate de căutare a simbolurilor de selectare și de lansare a opțiunilor. Analizele interfețelor vizează atât erorile, cât și duratele, fără luarea în considerare a calificărilor celor care utilizează aplicația. Interfața este aceea care stabilește atitudinea dezvoltatorilor de aplicații informatice față de utilizatori.

Dacă la definirea problemei și elaborarea specificațiilor sunt luate în considerare abilitățile utilizatorilor, la testarea pe eșantion reprezentativ nu vor apare diferențe semnificative.

7. Efectuarea de măsurători

Testarea interfeței este un bun prilej pentru efectuarea de măsurători prin care se definește performanța acesteia. Testele pe baza specificațiilor conduc la un set de măsurători și, respectiv, la un set de valori ale indicatorilor de performanță. Este preferabil ca, în procesul de testare, să fie adăugate noi seturi ce corespund datelor introduse de utilizatorii unui eșantion reprezentativ. Se constituie un alt set de măsurători și, respectiv, niveluri ale indicatorilor de performanță. Dacă, din punct de vedere statistic, sunt diferențe semnificative, toate analizele se efectuează folosind datele de test, provenind din înregistrările efectuate pe eșantionul de utilizatori, dacă acesta este într-adevăr reprezentativ.

Indicatorul rată de succes R_s este obținut de relația:

$$R_s = \frac{K}{N}$$

unde: K = numărul de activări care duc la obținerea rezultatului dorit de utilizator din punctul de vedere al interfeței; N = numărul total de activări asupra interfeței.

De regulă, pentru datele din specificații, proiectanții și realizatorii interfețelor ca $R_s^s = 1$ (rată succes din specificații). În realitate, prin diferite variații ale utilizării de către subiecții ce formează eșantionul de test, $R_s^e < 1$ (rată succes efectivă). În final $R_s = \min\{R_s^s, R_s^e\}$.

Indicatorul lungime traseu L_t este dat de numărul de pași parcurși pentru a obține rezultatul dorit. Dacă datele de test se presupun de lungimile $L_{t1}, L_{t2}, \dots, L_{tN}$, atunci lungimea medie a traseului \bar{L}_t este dată de relația:

$$\bar{L}_t = \frac{\sum_{i=1}^N L_{ti}}{N}$$

Și în acest caz, se calculează lungimea medie a traseului, rezultată din specificații \bar{L}_t^s și lungimea medie a traseului, rezultată din măsurătorile pe eșantionul reprezentativ de utilizatori, \bar{L}_t^e . În final, interfața are lungimea de traseu dată de relația: $\bar{L}_t = \max\{\bar{L}_t^e, \bar{L}_t^s\}$.

Indicatorul durată medie de eșantionare a problemei este calculat atât pentru datele din specificații \bar{D}_s cât și pentru datele provenind din eșantionul de utilizatori, \bar{D}_e . Durata medie de timp se obține din relația: $\bar{D} = \max\{\bar{D}_s, \bar{D}_e\}$.

De asemenea, alți indicatori se obțin prin măsurători efectuate după proceduri care să nu distorsioneze semnificația întregului proces.

Indicatorul durată de acomodare cu interfața A_a reprezintă un element deosebit de important în acceptarea sau respingerea aplicației. Când sunt folosite comenzi deja cunoscute, când simbolurile deja sunt întrebuițate, sunt create premisele pentru o largă utilizare. Dacă o interfață revoluționează interacțiunea om-calculator, prima consecință care apare trebuie să fie legată de reducerea severă a duratei de acomodare cu interfața A_a , simultan cu creșterea ratei de succes și reducerea duratelor necesare atingerii obiectivelor.

Se produc aceste deziderate atunci când testarea este temeinică și induce acțiuni de corecție ale interfețelor supuse procesului de testare.

8. Concluzii

Cele mai multe interfețe conțin combinații ale tipologiilor descrise în acest articol. Pentru fiecare tip de dialog, se realizează testarea după regulile prezentate. Testarea interfețelor vizează un plan riguros, construit în faza de proiectare, căutându-se minimizarea riscurilor de introducere de date eronate și crescând, pe cât posibil, capacitatea interfeței de a scurta revenirea utilizatorului, în vederea derulării corecte.

Pentru automatul bancar, este vitală definirea de interfețe care să scurteze procesele și, mai ales, să elimine situațiile în care se produc blocări de dialog, știută fiind dimensiunea efectelor și disconfortul asociat.

Testarea interfețelor trebuie să vizeze simplitatea dialogului, continuitatea traseelor în raport cu alte interfețe și, mai ales, lungimea fluxurilor.

Rezultatul testării unei interfețe este RAPORTUL de TESTARE, în care este subliniată diferența între comportamentul dat în specificații și cel efectiv, pentru seturile de date descrise în specificații.

Este deosebit de important ca raportul de testare să includă și unele observații produse pe marginea interacțiunii om - calculator cu eșantioane de utilizatori, pentru a înzestra interfața cu caracteristica de operaționalitate, în raport cu cerințele reale ale utilizatorilor.

Bibliografie

1. **ALBĂSTROIU, P., A.M. BOROZAN, R. CIOCEA, M. CRĂCIUN:** Realizarea unei platforme experimentale de servicii în rețea cu interfață grafică și/sau multimedia. Noi tehnologii pentru servicii grafice, multimedia și legătura cu baze de date în rețea, septembrie, 1998, 60p. Tema de cercetare ICI: A.37.2.
2. **DUMITRESCU, E.:** Java API: manualul interfeței de programare a aplicațiilor, Editura Teora, București, România, 1998, 358p. (traducere după: Fraizer, C., J. Bond :Java API, New Riders Publish., SUA, 358p.).
3. **HOMA JAVAHERY, AHMED SEFFAH:** A Model for Usability Pattern-Oriented Design, Task Models and Diagrams for user interface design, Tamodia 2002, INFOREC Printing House, Bucharest 2002, pp. 104-109.
4. **LUNGU, C., CHILIM, S., A. STĂTESCU, C. PRIBEANU:** SR ISO/CEI 11072 Tehnologia informației. Grafica pe calculator. Model de referință pentru grafica pe calculator, 1998, 32p.
5. **LUNGU, C., CHILIM, S., M. CĂLINESCU, C. PRIBEANU, M. POPA, A. POPA:** SR ISO/CEI 10641 Tehnologia informației. Grafica pe calculator. Testarea pentru conformitate a standardelor de grafică pe calculator, 1998, 31p.
6. **IANCULESCU, M.:** Realizarea de interfețe generice, centrate pe om, cu elemente de ergonomie cognitivă. Tehnici și metode utilizate la nivelul programării obiectelor în cadrul unei interfețe om-mașină, octombrie, 1998, 44p. Tema de cercetare ICI: A. 22.2.
7. **MANOLIU, C., O. CONSTANTINESCU, GH. RĂUȚ, C. UNGUREANU:** Proiectarea și experimentarea sistemului Registrului informatizat al agenților economici și sociali. Specificații de definire și realizare modul interfață cu utilizatori, dezvoltatori de aplicații, iulie, 1998, 37p. Tema de cercetare ICI: A.50.1.
8. **PRIBEANU, C., M. SAVIN:** Realizarea de interfețe generice, centrate pe om, cu elemente de ergonomie cognitivă. Tehnici și instrumente suport pentru evaluarea euristică a interfețelor om-calculator, noiembrie, 1998, 56p. Tema de cercetare ICI: A.22.4.
9. **VASILIU, C., A. T. NERȚAN, CL. IONESCU:** Realizarea de interfețe generice, centrate pe om, cu elemente de ergonomie cognitivă. Sisteme de tip asociat bazate pe calculator al operatorului uman. Elaborare de programe pentru interfețe operator, octombrie, 1998, 51p. Tema de cercetare ICI: A.22.3.