

EXPERIENȚA REALIZĂRII APLICAȚIILOR ORIENTATE LA UTILIZATOR

Gheorghe CĂPĂȚĂNĂ

Catedra Tehnologii de Programare

During the transition period from the existing technologies to new ones based on a different philosophy, all methods regarding information system design are welcome if they are capable to be immediately exploited and modified by the final users.

Introducere

Tehnologia proiectării *sistemelor informatice (SI)* a acumulat o experiență bogată. Însă, și în prezent rămân actuale problemele proiectării în termene minime și cu cheltuieli reduse a **SI**, care ar respecta concomitent calitățile de a fi dinamice, adaptabile la evoluția obiectului condus și a mediului de activitate; după implementare să poată fi susținute și dezvoltate de către înșiși utilizatorii finali (neafectând fiabilitatea **SI**).

În lucrare este expusă experiența elaborării **SI** orientate la utilizator și demonstrat că aceste produse informaționale respectă exigențele formulate.

1. Exigențele înaintate față de sistemele informatice

Realizarea unui limbaj natural (interfață în limbaj natural este una dintre caracteristicile tehnologiei informaționale a generației a cincea de calculatoare) este la fel de dificilă ca și realizarea "*modelului lumii*".

Pe de altă parte, întreprinderile industriale sunt sisteme evolutive, iar conceptele care evoluează sunt termeni ai unui limbaj natural. Submulțimea acestui limbaj utilizată în scopul elaborării modelelor dinamice pe calculator trebuie să satisfacă unui șir de cerințe severe:

- ✓ să posede o semantică dinamică continuu evolutivă; în caz contrar, la o modificare a obiectului, calitatea modelului se alterează;
- ✓ semantica limbajului unui model dinamic trebuie să reflecte modificările anturajului economic și ale stării obiectului, dar în același timp trebuie să conserve versiunile anterioare, adică să posede "*memorie*"; de exemplu, semanticile anterioare vor fi în continuare utilizate pentru calculul: bilanțului economic anual, concediilor etc. Rezultă că un model dinamic cu "*memorie*" reprezintă o mulțime extensibilă, integrată, de modele statice ale obiectului, al căror număr este în continuă creștere;
- ✓ limbajul modelului dinamic trebuie să fie simplu, firesc, specializat pe profesii, deoarece evoluția sa continuă solicită modificări permanente care în procesul exploatării se transferă de la autorul modelului la utilizatorul acestuia;
- ✓ elementele lingvistice utilizate în elaborarea modelelor reprezintă termeni și sintagme ale limbajelor specializate pe profesiile utilizatorilor finali. Deci, limbajele utilizate la realizarea modelelor dinamice din cadrul întreprinderilor industriale reprezintă de fapt submulțimi ale limbajului natural limitate sintactic, semantic și pragmatic;
- ✓ obiectele industriale sunt obiecte ierarhizate; în consecință, modelele dinamice cu "*memorie*" ale acestei clase de obiecte reprezintă modele ierarhizate în care fiecare subsistem, la rândul său, reprezintă un model dinamic cu "*memorie*". Atât limbajul coordonatorului sistemului, cât și limbajele fiecărui subordonator în parte trebuie să satisfacă toate aceste cerințe severe;
- ✓ să permită rezolvarea problemelor slab structurate.

Aceste condiții impuse limbajelor sistemelor software destinate întreprinderilor industriale sunt în concordanță cu principiile filosofice care fundamentează noile tehnologii informaționale ale generației a cincea de calculatoare.

Unele subsisteme ale sistemului informatic operează în timp real în rețea cu informații care reprezintă cantități mari de bunuri materiale și resurse financiare și, din acest motiv, solicită un software cu fiabilitate ridicată. De asemenea, accidente în funcționarea hardware, în alimentarea cu utilități sau erori de manevră în condițiile unui flux enorm de informații pot deveni fatale pentru sistemul condus.

În consecință, sistemului deja realizat i se atașează un model suplimentar, care permite excluderea mutuală a proceselor paralele de prelucrare a informației în timp real și garantează cu probabilitatea de 100% restabilirea bazelor de date în cazuri excepționale și continuarea imediată a procesului de prelucrare a informației din momentul repornirii calculatorului.

După cum se observă, ipotezele de lucru utilizate în formularea problemei sunt de natură lingvistică și, în consecință, soluționarea va fi tot de natură lingvistică. Pe scurt, esența rezolvării problemei de studiu se reduce la proiectarea unei tehnologii de elaborare a stațiilor de lucru destinate întreprinderilor industriale în concordanță cu principiile filosofice ale noii tehnologii informatice. Fiecare produs al acestei tehnologii este un sistem semiotic care reprezintă fie un sistem de asistare a deciziilor (structurate sau slab structurate), fie îmbină unele funcții ale acestora, ale sistemelor expert și ale sistemelor bazate pe logică și calcul.

2. Domeniul de aplicație

Produsele program aplicative, sau aplicațiile, oferă servicii informaționale utilizatorilor finali care activează într-un oarecare domeniu de aplicație.

Definiția 1. Domeniul de aplicație reprezintă o totalitate de obiecte (esențe), de legături și relații dintre aceste obiecte și proceduri de transformare a acestor obiecte în procesul rezolvării problemelor din acest domeniu [1].

Domeniul de aplicație $\mathbf{DA(t)}$ este triplu:

$$\mathbf{DA(t) = \langle \mathbf{OBIECTE(t)}, \mathbf{RELATII(t)}, \mathbf{PROCEDURI(t)} \rangle,}$$

unde:

- $\mathbf{OBIECTE(t)}$ este mulțimea obiectelor din $\mathbf{DA(t)}$:

$$\mathbf{OBIECTE(t) = \{ \mathbf{OBIECT}_1(t), \dots, \mathbf{OBIECT}_n(t) \};}$$

- $\mathbf{RELATII(t)}$ este mulțimea relațiilor dintre aceste obiecte:

$$\mathbf{RELATII(t) \subseteq (\mathbf{OBIECTE(t)} \times \mathbf{OBIECTE(t)});}$$

- $\mathbf{PROCEDURI(t)}$ este mulțimea procedurilor:

$$\mathbf{PROCEDURI(t) = \{ \mathbf{PROCEDURĂ}_1(t), \dots, \mathbf{PROCEDURĂ}_m(t) \},}$$

fiecare procedură $\mathbf{PROCEDURĂ}_i(t)$ realizează transformarea:

$$\mathbf{PROCEDURĂ}_i(t): \mathbf{OBIECTE(t)} \rightarrow \mathbf{OBIECTE(t)}.$$

La alegerea limitelor domeniului de aplicație sunt importanți doi factori:

- *domeniul de aplicație constă din conceptele utilizate în activitățile lor de către utilizatorii finali;*
- *baza de cunoștințe a domeniului de aplicație trebuie să nu fie prea mare, încât să complice culegerea informației și soluționarea problemelor din acest domeniu.*

Definiția 2. Universul unei probleme (unui set de probleme) $\mathbf{UProb(t)}$, reprezintă o submulțime a domeniului de aplicație $\mathbf{DA(t)}$ folosită în procesul rezolvării acestei probleme (set de probleme).

Domeniul de aplicație $\mathbf{DA(t)}$ constă din unul sau mai multe $\mathbf{UProb(t)}$:

$$\mathbf{DA(t) = UProb}_1(t) \cup \dots \cup \mathbf{UProb}_n(t).$$

Definiția 3. Universul unui program $\mathbf{UProg(t)}$ reprezintă reuniunea universurilor setului de probleme $\mathbf{UProb(t)}$ realizat de acest program.

Universul $\mathbf{UProg(t)}$, de asemenea, reprezintă o submulțime a domeniului de aplicație $\mathbf{DA(t)}$:

$$\mathbf{UProg(t) \subseteq UProb(t)}.$$

Limbajele utilizatorilor finali reprezintă submulțimi ale $\mathbf{DA(t)}$:

$$\mathbf{DA(t) = LU}_1(t) \cup \dots \cup \mathbf{LU}_s(t).$$

3. Exemplu de elaborare a limbajului utilizatorului final

Utilizând metoda elaborării limbajului utilizatorului final expusă în [2-4], vom elabora limbajul declarativ al unei aplicații dotate cu procesor în LU.

Problemă: Fie că un Combinat de Panificație are în subordine 5 fabrici. Combinatul a comandat elaborarea sistemului informatic pentru fabrici și Combinat, inclusiv problema:

Problemă_{1(t)}: Realizarea planului de producere al unităților economice pe un orizont de timp

Formularea problemei **Problemă₁(t)** este expusă în limbaj natural și conține următoarele sintagme semnificative:

- „realizarea planului de producere”;
- „unitate economică”;
- „orizont de timp”.

Supunem fiecare din aceste sintagme unui proces de decompoziție cu scopul elaborării universului acestei probleme - **UProb₁(t)**:

UProb₁(t) ::= <realizarea planului de producere><unitate economică><orizont de timp>

<realizarea planului de producere (r.p.p.)> ::= <r.p.p. în valori absolute> |
<r.p.p. în valori relative>

<r.p.p. în valori absolute> ::= <produse>, **planificat, realizat**, <deviere realizat de la planificat>

<deviere realizat de la planificat> ::= **realizat – planificat**

<r.p.p. în valori relative> ::= <produse>, **planificat, realizat**,

<% devierii realizat de la planificat>

<% devierii realizat de la planificat> ::= **100 · realizat / planificat**

<produse> ::= **Denumire produs în Nomenclatorul „Produse finite”** |

<sortiment de produse> | **Total sortiment** |

<grup de produse definit de cuvinte-cheie> |

<grup al Nomenclatorului „Produse finite”> | **Total grup** |

<subgrup al Nomenclatorului „Produse finite”> | **Total subgrup**

<sortiment de produse> ::= **Sortiment**

<grup de produse definit de cuvinte-cheie> ::= <cuvinte-cheie>

<cuvinte-cheie> ::= <cuvânt-cheie> | <cuvânt-cheie>, <cuvinte-cheie>

<cuvânt-cheie> ::= „**cuvânt-cheie**”

<unitate economică> ::= **Fabrică de pâine nr.1 | Fabrică de pâine nr.2 |**

Fabrică de pâine nr.3 | Fabrică de pâine nr.4 | Fabrică de paste făinoase |

Combinatul de Panificație „Franzeluța” din Chișinău

<orizont de timp> ::= <ziua curentă> | **1 | 2 | ... | 31** |

<luna curentă> | **ianuarie** | ... | **decembrie** |

<trimestrul curent> | **trimestrul I** | ... | **trimestrul IV** |

<anul curent>

4. Aplicații orientate la problemă, domeniu și utilizator

Definiția 4. O aplicație este considerată orientată la problemă dacă problema realizată în cadrul acestei aplicații se rezolvă în întreprinderi din diverse domenii de aplicație.

Problema realizată în cadrul unei aplicații orientate la problemă își păstrează valoarea aplicativă trecând de la un domeniu de aplicație la altul și nu depinde de domeniul de aplicație. De aceea, este numită aplicație orientată la problemă.

De exemplu: Un pachet de contabilitate, un Sistem Suport pentru Asistarea Deciziilor Manageriale, pachetul WORD etc. pot fi considerate orientate la problemă, deoarece facilitățile informaționale oferite de aceste aplicații sunt zilnic solicitate în unități din diverse domenii de activitate: educație, industria alimentară, medicină, organele de conducere ale republicii și municipiului etc.

Definiția 5. O aplicație este considerată orientată la domeniu dacă problema realizată de aplicație poate fi implementată la întreprinderi dintr-un unic domeniu de aplicație.

De exemplu: Aplicația „Calculul necesarului de materii prime pentru realizarea planului de producere la o brutărie pe un orizont de timp” poate fi implementată la toate întreprinderile din industria panificației. Implementarea acestei aplicații la întreprinderi din alte domenii va solicita anumit efort intelectual, financiar și de timp.

Definiția 6. O aplicație este considerată orientată la utilizator dacă problema realizată de aplicație poate fi implementată la un unic post informațional de lucru.

Aplicațiile orientate la utilizator apelează aceleași compartimente ale bazei de date, pot avea aceleași forme de rapoarte, dar diferă la regimul de funcționare și acces, încât sunt produse „personalizate” la utilizator.

De exemplu: Sistemul informatic „Asistența procesului de expertizare a proiectelor din sfera științei, inovării și transferului tehnologic” are în componența sa mai multe posturi informaționale de lucru (PIL):

- PIL al Direcției Management în Sfera Științei și Inovării;
- PIL al Agenției pentru Inovare și Transfer Tehnologic;
- PIL al Consiliului Consultativ de Expertiză etc.

Aceste posturi accesează aceleași fișiere, generează informații și rapoarte similare. Dar, PIL sunt considerate orientate la utilizator, fiindcă sunt elaborate pentru utilizatori concreți și nu pot satisface solicitările informaționale ale altor utilizatori, decât ale celui pentru care a fost elaborat.

Calitățile aplicațiilor de a fi ori nu „orientare la domeniu”, „orientare la problemă” și „orientate la utilizator” reprezintă un reper tridimensional pentru sistemele informatice (Fig.1).

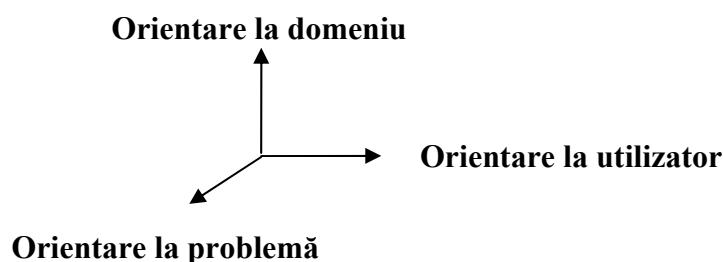


Fig.1. Reperul sistemelor informatice.

5. Reprezentarea conceptelor dinamice

În funcție de timp, sunt următoarele noțiuni ale SI:

- ✓ structura organizatorică a obiectului condus;
- ✓ structura tehnologică a obiectului condus;
- ✓ nomenclatorul și caracteristicile materiilor prime;
- ✓ nomenclatorul și caracteristicile produselor finite;
- ✓ nomenclatorul furnizorilor;
- ✓ nomenclatorul cumpărătorilor;
- ✓ programul de producere (5,6,7 zile lucrătoare, săptămână necompletă, 1,2,3 schimburi, sezon, la solicitare etc.);
- ✓ lista materiilor prime care pot fi înlocuite;
- ✓ nomenclatorul: tarife, servicii;
- ✓ rețeturile de preparare a produselor finite;
- ✓ indicii, a căror valoare cantitativă este stabilită de organele de administrare (impozit, acciz etc.) și algoritmi de calculare a acestora;
- ✓ indicii, a căror valoare cantitativă este stabilită de condițiile contractului (prețuri, remunerare, condiții și termene de furnizare, condiții de penalizare etc.);
- ✓ personalul unității economice și caracteristicile acestuia etc.

Conceptele dinamice ale sistemului informatic pot fi reprezentate utilizând următorul model de reprezentare a cunoașterii:

$\langle \text{obiect} \rangle ::= \{ \langle \text{termen valabilitate} \rangle, [\langle \text{obiect nou} \rangle], \{ \langle \text{proprietate} \rangle \}^*, \langle \text{remarcă} \rangle \}^{**}$

$\langle \text{termen valabilitate} \rangle ::= \langle \text{începutul termenului} \rangle - \langle \text{sfârșitul termenului} \rangle$

$\langle \text{proprietate} \rangle ::= \langle \text{atribut} \rangle = \langle \text{valoare} \rangle$

$A^* ::= A \mid A^*, A$

$A^{**} ::= A \mid A^{**} . A$

$\langle \text{remarcă} \rangle ::= \text{„semnificația propoziției în limbajul utilizatorului final”}$

Un obiect dinamic în baza de cunoștințe a sistemului informatic se păstrează ca o consecutivitate de propoziții (afirmații). Fiecare propoziție termină cu semnul „punct”. Propoziția constă din una sau câteva sintagme. Fiecare sintagmă servește pentru descrierea proprietăților obiectului și termenului de valabilitate al acestor proprietăți.

Prima sintagmă servește pentru specificarea termenului de valabilitate a proprietăților obiectului din propoziția procesată.

Se utilizează două forme de reprezentare a termenului de valabilitate:

- $t_0 \leq t \leq t_1$, unde t_0 este începutul termenului de valabilitate, iar t_1 este sfârșitul termenului de valabilitate a proprietăților obiectului din prima propoziție;
- $t_m < t$, unde t_m este începutul termenului de valabilitate a proprietăților obiectului din ultima propoziție. Sfârșitul termenului de valabilitate a proprietăților obiectului din ultima propoziție la moment nu este cunoscut, de aceea în ultima propoziție nu se completează.

Sintagma a doua, dacă este folosită, permite modificarea numelui obiectului.

Dacă în sistemul informatic al întreprinderii este utilizat acest model de reprezentare a obiectelor dinamice, atunci în orice moment se poate de efectuat o reevaluare a stării obiectului condus la ziua de azi și în oricare zi din trecut. Bazele de date relaționale nu permit această facilitate. Modelul discutat de reprezentare a semnificației obiectelor dinamice a fost implementat în sistemul informatic elaborat pentru Asociația „Vitanta”.

Un exemplu de reprezentare a obiectelor dinamice din domeniul educației este prezentat în Tabel. Exemplul demonstrează că, în cazul utilizării acestui model pentru reprezentarea informației referitoare la studenți, operativ se poate de aflat starea contingentului de studenți la ziua de azi și în oricare zi din trecut.

Tabel

Reprezentarea unui obiect dinamic în baza de cunoștințe

Nume concept	Termen valabilitate $t_0 \leq t \leq t_1$		Semnificație: atribut - valoare				Remarcă
	t_0	t_1	Nume după căsătorie	Specialitatea	Cursul	Grupa academică	
<i>Toma Elena</i>	010903	040704		MI	1	MI-12	<u>Ordin:</u> nr.15/st din 040704
	040704	100705		MI	2	MI-21	<u>Ordin ...</u>
	110705	080706	<i>Vlas Elena</i>	MNi	3	MNi-32	Nume după căsătorie <u>Ordin ...</u> , <u>Ordin...</u>
	090706			MNi	3	MNi-33	
<i>Adrei Petru</i>	...						

6. Principiul de funcționare a unei aplicații orientate la utilizator

Un produs program aplicativ orientat la utilizator este un produs inteligent și constă din următoarele componente (Fig.2):

- rezolvitorul de probleme;
- baza de date;
- baza de cunoștințe dinamice;
- baza de modele în LU a domeniului de aplicație;
- interfața inteligentă a utilizatorului final.

Soluționarea pe calculator a unei probleme utilizând produsul program inteligent descris este următoarea:

- 1) utilizatorul final formulează problema de rezolvat în limbaj profesional, care este o submulțime structurată a limbajului natural;
- 2) rezolvitorul de probleme selectează din Baza de modele modelul în LU potrivit problemei formulate;
- 3) în baza modelului selectat rezolvitorul de probleme generează automat în memoria calculatorului algoritmul soluționării problemei formulate;
- 4) în caz dacă algoritmul utilizează concepte dinamice, rezolvitorul de probleme oferă algoritmului semnificația actuală a acestor concepte la momentul de timp la care se rezolvă problema.

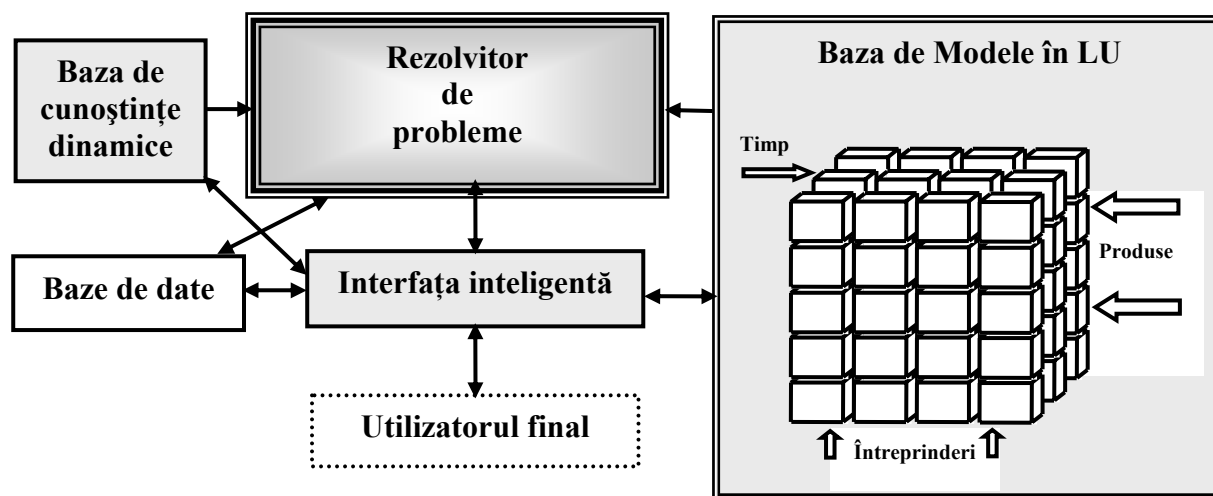


Fig.2. Principiul de funcționare a unei aplicații orientate la utilizator.

7. Concluzii

Metoda descrisă poate fi extrapolată și în alte domenii de aplicație a sistemelor informatice. Fiecare produs al acestei tehnologii este dotat cu un limbaj specializat pe profesie, numit limbajul utilizatorului final. Utilizatorul final poate ușor completa baza de modele cu noi modele, deoarece acestea sunt elaborate în limbaj profesional.

Aplicațiile elaborate și dotate cu procesoarele LU păstrează un grad înalt de fiabilitate, deoarece în cel mai nereușit caz poate fi afectat de eroare doar modelul care este elaborat. Dar, întrucât: a) modelul este elaborat în limbaj de nivel înalt și b) utilizatorul final este asistat de o bibliotecă de exemple corecte (*modelele în LU*), experiența a demonstrat că utilizatorul final se isprăvește de sine stătător cu problemele de elaborare a noilor modele.

Fiecare produs al tehnologiei propuse reprezintă un model asemănător ceasului, cu excepția că mecanismul este în evoluție continuă, mulțimea de taste este în posibilă creștere, fiecare tastă apelează un program în LU, iar "tastele" sintetizate de către utilizatorul final pot fi modificate.

Tehnologia descrisă în articol a fost utilizată la elaborarea sistemelor informatice pentru Combinatul de Panificație „Franzețuța” din Chișinău, Asociația „Vitanta”, Combinatele de producere a berii și a băuturilor nealcoolizate din orașele Lipețk și Novosibirsk, Organizația Obștească de Informare și Instruire Ecologică „Terra Nostra” etc.

Tehnologia informațională de elaborare a sistemelor informatice orientate la utilizator oferă programatorului de aplicații facilități de a elabora productiv produse program aplicative. În acest context ne referim fără comentarii la performanța documentată de către Combinatul de Panificație „Franzețuța” din Chișinău: „Aplicația „Calculul planului de producere al fabricii de panificație, combinatului de panificație, asociației” constă dintr-un program dotat cu procesor LU și testele corespunzătoare. Volumul aplicației este de 1025 rânduri în limbaj de programare. Timpul elaborării – 10,5 ore”.

Bibliografie:

1. Толковый словарь по искусственному интеллекту / А.Н. Аверкин, М.Г. Гаазе-Рапопорт, Д.А. Поспелов. - Москва: Радио и связь, 1992.
2. Căpățână Gh. The methods and instruments for elaboration of the flexible informatics systems. - In: "Computer Science". The proceedings of the 3rd International Symposium of Economic Informatics - Mai 1997. - Bucharest: Academy of Economic Studies, 1997, p.82-86.
3. Căpățână Gh. Elaborarea software aplicativ // Realizări și tendințe în implementarea mecanismelor economiei concurențiale. Simpozion științific internațional (22-23 aprilie 1999). Materiale și sinteze. Vol. I. - Chișinău: UCCM, 2000, p.173-175.
4. Căpățână Gh. Metoda limbaj-utilizator de proiectare a sistemelor informaționale / Buletin științific. - Anul III. - 2000. - Nr.1. - Bacău: Universitatea „George Bacovia”, 2000, p.43-48.

Prezentat la 04.04.2007