

SISTEME INFORMATICE ADAPTIVE

Gheorghe CĂPĂȚĂNĂ

Catedra Tehnologii de Programare

Synthesis, cod multiplication and system adaptabilities are concepts inspired from biology. In the paper are related the technique of informational systems development oriented on problem. This class of systems proves the following qualities: the elaboration efficiency, the mobility at the application and adaptability domains to the evolution of the application area.

Introducere

În articol sunt expuse unele aspecte ale dezvoltării sistemelor informatice (SI). Realizarea unui SI este determinată de domeniul de aplicație, resursele disponibile, exigențele formulate etc. F. Filip (2007) observă că în aplicațiile economice sistemul informatic trebuie să fie „adaptiv față de evoluția cerințelor utilizatorului și a mediului său (organizația), cât și față de schimbările tehnologice”. SI realizate cu contribuția tehnologiilor informaționale convenționale solicită modificarea textelor programelor la fiecare astfel de evoluție. Modificările textelor programelor pot afecta fiabilitatea SI. De aceea, elaborarea SI adaptive este actuală și avantajoasă în condițiile mediilor de exploatare evolutive. În cele ce urmează va fi expusă experiența autorului la elaborarea SI adaptive. Pentru realizarea acestui scop sunt folosite scheme tehnologice care preconizează elaborarea preventivă a unui SI generic, cu ajutorul căruia ulterior sunt generate versiuni ale SI pentru o gamă de aplicații.

La elaborarea conceptului de sistem informatic evolutiv a fost folosită creativ experiența elaborării: sistemelor ierarhizate, în timp real, cu prelucrarea distribuită a datelor (Bragaru T., Bulat E., Crăciun I., 2000; Căpățână Gh., Organ A., 2007; Guran M., Filip F., 1986; Oprea D., 1999; Stanciulescu F., 2003), sistemelor adaptive (Cojocaru S., 2007; Filip F., 2007), sistemelor suport pentru decizii (Donciulescu D.A., 1998; Filip F., 1997; Gaindric C., 1998; Rădulescu D., Gheorghiu O., 1992), sistemului inteligent distribuit de asistență a unei clase de experimente în genetică (Duca M., Căpățână Gh., Levițchi A., Podduchin V., 2006), P-sistemelor (Alhazov A., Margenstern M., Rogozhin V., Rogozhin Yu., Verlan S., 2006; Păun Gh., 2000), sistemului expert în medicină (Butnaru M., Căpățână Gh., Popov Al., Sturza G., 2007), sistemului inteligent pentru rezolvarea aproximativă a ecuațiilor integrale (Căpățână Gh., Seiciuc E., 2004), sistemelor orientate pe problemă (Căpățână Gh., 2007; [21]). În articol au fost folosite metode de descriere aplicate în teoria sistemelor (Mesarovic M.D., Takahara Y., 1975).

1. Domeniu de aplicație. Sistem informatic

Definiția 1. Sistemul este un „ansamblu de elemente aflate într-o relație structurală, de interdependență și interacțiune reciprocă, formând un tot organizat” [13].

Sistemul este caracterizat de legătura cu mediul ambiant – domeniul de aplicație.

Un domeniu de aplicație reprezintă totalitatea obiectelor (esențelor), de legături și relații dintre aceste obiecte și proceduri de transformare a acestor obiecte în procesul rezolvării problemelor din acest domeniu [22].

Definiția 2. Un domeniu de aplicație (abreviat DA) este un 3-uplu

$$DA = (O, R, P) \quad (1)$$

unde:

(i) O este mulțimea obiectelor din DA

$$O = \{o_i \mid i \geq 1\},$$

(ii) R este mulțimea relațiilor dintre aceste obiecte

$$R \subseteq \{O \times O\},$$

(iii) P este mulțimea procedurilor

$$P = \{ p_k \mid k \geq 1 \},$$

(iv) Fiecare procedură p_k realizează transformarea:

$$p_k : O \rightarrow O.$$

După cum observă W.Păvăloaia (2000), „sistemul economic definește concepte și ansambluri economice. O unitate economică, o ramură a economiei și însăși economia națională sau cea mondială se comportă ca sisteme”.

Economia națională (abreviat EN) este organizată pe ramuri economice (abreviat R), iar în cadrul fiecărei ramuri – pe subramuri economice, considerate în cadrul articolului domenii de aplicație.

$$EN = \{ R_j \mid j \geq 1 \},$$

$$R_j = \{ DA_{i,j} \mid i \geq 1 \}.$$

Vom nota cu D mulțimea domeniilor de activitate ale economiei naționale

$$D = \{ DA_{i,j} \mid i = \overline{1, n_j}, j > 0 \} = (O, R, P), \quad (2)$$

unde O, R, P sunt, respectiv, mulțimile din (1).

Definiția 3. Fie SE un sistem economic. Universul sistemului economic SE (abreviat *Univers (SE)*) este submulțimea conceptelor mulțimii D realizată de către acest sistem $Univers (SE) \subseteq D$.

Fiecare sistem economic constă din următoarele componente: a) *sistemul condus*, b) *sistemul de conducere* și c) *sistemul informațional*. *Sistemul condus* transformă intrările (resursele materiale, umane, financiare, informaționale) în ieșiri (produse materiale, servicii, informații). *Sistemul de conducere* asigură realizarea obiectivelor formulate sistemului condus.

Definiția 4. Sistemul informațional este un ansamblu de componente care colectează, stochează și distribuie informații pentru suportul decizional și al controlului dintr-o întreprindere (Laudon K.C., Laudon J.P., 2000).

Locul sistemului informațional în structura sistemului economic este reprezentat în Figura 1.

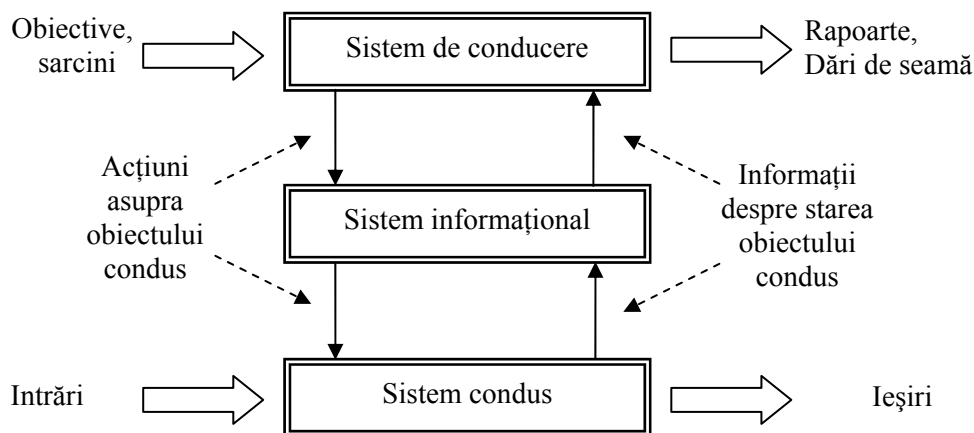


Fig.1. Locul sistemului informațional în structura sistemului economic.

Sistemul informațional al unui sistem economic este conceput ca un grup de posturi informatice, bazate pe tehnologia bazelor de date relaționale și tehnologiile moderne de comunicare.

Sistemul informatic este un sistem care permite introducerea de informație, stocarea informației, prelucrarea informației și extragerea informației sub formă de rapoarte (Păvăloaia W., 2000).

Definiția 5. Un sistem informatic este un 5-uplu

$$SI_{se} = (Input, P, BD, Post, Output) \quad (3)$$

unde:

- (i) *Input* este ansamblul intrărilor SI,
- (ii) *P* sunt procesele de transformare a intrărilor în ieșiri,
- (iii) *BD* este baza de date a SI,
- (iv) *Post* este mulțimea posturilor informatice de lucru (în cele ce urmează – posturi) ale SI;
- (v) *Output* este ansamblul ieșirilor SI.

2. Exigențe impuse sistemului informatic

Procesul de proiectare a SI presupune anumite exigențe. De exemplu, condițiile mediului sistemului economic evoluează. De aceea, SI trebuie să demonstreze adaptabilitate la evoluția acestor condiții. Modificările componentelor software ale SI, efectuate în legătură cu aceste evoluții, nu trebuie să afecteze fiabilitatea sistemului. La etapele de dezvoltare a SI exigențe se impun, de asemenea, asupra securității sistemului, confidențialității informațiilor, calității și productivității dezvoltării SI, bazei de date, posturilor și economiei de resurse (personal calificat, fond de salarizare, timp etc.). Subiectul articolului se înscrie în această tendință, autorul având drept scop propunerea unor structuri SI adaptive, fiabile și elaborarea metodelor de construire a SI care satisfac acestor exigențe.

3. Adaptabilitatea sistemelor informatice

Definiția 6. Sistem adaptiv este sistemul care automat modifică algoritmi funcționării sale și (uneori) structura sa în scopul păstrării sau atingerii stării optimale la schimbarea condițiilor mediului [7].

Calitatea de *adaptare* a sistemului informatic poate fi realizată sub următoarele aspecte (Cojocaru S., 2007):

- Adaptarea se efectuează apriori, prin proiectarea orientată spre un anumit tip de utilizatori.
- Sistemul permite o adaptare dinamică, care este efectuată de către utilizator în procesul de comunicare cu sistemul.

- Adaptarea este realizată de către sistem în procesul de comunicare cu utilizatorul.

Sistemele informatice adaptive pot fi elaborate prin mai multe metode. Printre acestea nominalizăm:

- Adaptarea sistemului informatic la domeniul de aplicație se face la etapa proiectării acestuia.

- Mediul de dezvoltare al sistemelor informatice comandat este dotat cu funcții de macroprocesare.

Sistemul solicitat este realizat în formă de *sistem generic parametrizat*. *Parametrii de adaptare* ai sistemului generic sunt macrovariabile. Atribuind valori simbolice oportune parametrilor de adaptare ai sistemului generic este generată versiunea SI solicitată de către beneficiar.

- Sistemul informatic comandat de către beneficiar este parametrizat și dotat cu un interpretor, care permite utilizatorului modificarea parametrilor de adaptare. În rezultatul acestei atribuirii, sistemul informatic este adaptat la condițiile curente ale domeniului de aplicație.

- Tehnologia informațională îmbină tehnici mixte de realizare a adaptivității. O parte din parametrii de adaptare ai SI solicitat de către beneficiar, care își modifică valorile rar, sunt realizați în sistemul generic. Altă parte, care își modifică frecvent valorile, sunt realizați în interpretor.

4. Generalizarea și concretizarea

Totalizând experiența dezvoltării familiei de sisteme suport pentru decizii DISPECER, F.Filip evidențiază că „durata oricărui proiect de introducere a unui sistem informatic poate fi redusă prin folosirea unor elemente prefabricate, „mai mult sau mai puțin integrate”. Costul se micșorează atunci când elementele prefabricate au un grad cât mai mare de generalitate și sunt refofosibile într-un număr cât mai mare de aplicații” [10].

Generalizarea și concretizarea sunt două tehnici generale de dezvoltare a obiectelor pornind de la alte obiecte.

În Dicționarul explicativ al limbii române *generalizarea* este definită ca o operație „prin care se trece de la particular la general, extrăgându-se caracterele comune esențiale unor obiecte asemănătoare” [13].

În rezultatul unei operații de generalizare se poate trece de la o familie de obiecte asemănătoare, a căror structură integrează aceleași elemente, la un nou obiect, încercându-se astfel a se face abstracție de respectivele obiecte. Obiectul obținut în rezultatul unei operații de generalizare reprezintă un *obiect generic*. *Concretizarea* este operația inversă generalizării.

Definiția 7. Fie D un domeniu de aplicație, O – mulțime arbitrară de obiecte din acest domeniu $O = \{o_i \mid i \geq 2\}$; o_1, o_2 – două obiecte din această mulțime $o_1, o_2 \in O$; A este un oarecare alfabet; cod_A este o oarecare funcție de codificare a mulțimii obiectelor O în acest alfabet; codurile obiectelor o_1, o_2 pot fi reprezentate:

$$\begin{aligned} cod_A(o_1) &= xu_1y = o_1(u_1), \\ cod_A(o_2) &= xu_2y = o_2(u_2), \end{aligned}$$

unde x, y, u_1, u_2 sunt cuvinte peste alfabetul A .

Obiectul o cu codul $cod_A(o) = xuy$ este generalizarea obiectelor o_1 și o_2

$$o = G(o_1, o_2) \quad (4)$$

G este operatorul de generalizare. Conceptul notat cu cuvântul u (u este cuvânt peste alfabetul A), este generalizarea conceptelor notate cu cuvintele u_1 și u_2

$$u = G(u_1, u_2).$$

Obiectul $o(u)$ este un model obținut în rezultatul generalizării obiectelor $o_1(u_1)$ și $o_2(u_2)$

$$o(u) = G(o_1(u_1), o_2(u_2)).$$

Producțiile $u_1 \rightarrow u$ și $u_2 \rightarrow u$ se numesc producții de generalizare. Producțiile inverse $u \rightarrow u_1$ și $u \rightarrow u_2$ se numesc producții de concretizare a modelului $o(u)$. Conceptul u este un parametru al modelului.

Definiția 8. Fie D un domeniu de aplicație, O – mulțime arbitrară de obiecte din acest domeniu, $o(u)$ – un model, $o(u) \in O$. Codul obiectului $o(u)$ într-un oarecare alfabet A poate fi reprezentat în forma $cod_A(o(u)) = xuy$, unde x, u, y sunt cuvinte peste A . Obiectul $o_1(u_1)$ este rezultatul operației de concretizare a modelului $o(u)$, dacă acesta poate fi obținut din $o(u)$ cu o oarecare substituție $u \rightarrow u_1$

$$cod_A(o_1(u_1)) = xu_1y = cod_A(K_{u \rightarrow u_1}(o(u))), \quad (5)$$

unde:

- (i) u_1 este cuvânt peste alfabetul A ,
- (ii) $o_1(u_1)$ este o concretizare a modelului $o(u)$,
- (iii) $K_{u \rightarrow u_1}$ este operația de concretizare cu substituția $u \rightarrow u_1$.

Operația de concretizare K_{\rightarrow} este o operație inversă operației de generalizare G . Operațiile de generalizare și de concretizare pot fi aplicate multiplu asupra mulțimii obiectelor domeniului de aplicație. În rezultat, pot fi obținute noi modele (obiecte abstracte, generalizate) și noi obiecte concretizate. Concretizarea este un

caz particular al operației de adaptare. Operațiile de generalizare și de concretizare pot fi folosite pentru dezvoltarea SI adaptive la condițiile domeniului de aplicație, iar în timpul exploatării sistemelor informatice – la evoluția acestor condiții.

5. Produse informatice orientate la problemă

SI solicitat de către o unitate economică din domeniul de aplicație $DA_{i,j}$ realizează pe calculatoarele rețelei setul de probleme comandat. Acest set specifică, de asemenea, mulțimea procedurilor de transformare P din (1). Numărul problemelor setului poate fi în creștere. Problemele realizate pentru un domeniu de aplicație iau în considerație structura și specificul acestui domeniu. Dezvoltarea SI pentru n unități economice ar solicita elaborarea a n versiuni de software pentru fiecare SI. Dacă elaborarea se face prin metoda *orientării la problemă* ([21]; Căpățână Gh., 2007), atunci volumul de lucru diminuează considerabil.

Aplicând metoda proiectării orientate pe probleme, aplicația obține următoarele avantaje:

- ✓ diminuarea volumului de proiectare și de elaborare;
- ✓ diminuarea timpului și resurselor necesare pentru realizarea aplicației;
- ✓ un grad înalt de standardizare a modulelor software;
- ✓ o calitate mai bună a sistemului elaborat ș.a.

O aplicație orientată la problemă poate demonstra mobilitate și la alte domenii de aplicație, diferite de cele inițial preconizate realizării pe calculator.

6. Dezvoltarea SI generic și adaptarea acestuia la condițiile mediului

Proiectarea SI trebuie să ia în considerație condițiile mediului de implementare. SI conceput flexibil va demonstra adaptabilitate la aceste condiții. Proiectarea SI în formă de sistem generic poate facilita generarea ulterioară a versiunilor sistemului informatic adaptate la condițiile domeniilor de aplicație.

Fiecare *unitate economică* (abreviat UE) este specializată într-un oarecare domeniu de aplicație. Acest domeniu reprezintă universul unității respective.

Definiția 9. Fie UE o unitate economică, D mulțimea domeniilor de aplicație

$$D = \{ DA_{i,j} \mid i > 0, j > 0 \}. \quad (6)$$

Universul unei unități economice (abreviat $Univers(UE)$), este submulțimea conceptelor mulțimii D realizată de unitatea economică UE

$$Univers(UE) \subseteq DA_{i,j} \subset D.$$

Fie că unitatea economică UE solicită realizarea pe calculator a sistemului informatic SI_{UE} . Dezvoltarea acestui sistem, luând în considerație (1), va fi construcția

$$SI_{UE} = (Input_{UE}, P_{UE}, BD_{UE}, Post_{UE}, Output_{UE}). \quad (7)$$

Elaboratorul sistemului informatic, prognozând viitoarele elaborări ale sistemelor informatice, selectează o submulțime D^* care include universul $Univers(UE)$

$$Univers(UE) \subseteq DA_{i,j} \subseteq D^* \subseteq D.$$

Mulțimea D^* poate include în funcție de strategia elaboratorului:

- ✓ domeniile de aplicații ale unei ramuri economice;
- ✓ domeniile de aplicație ale mai multor ramuri economice.

Mulțimea D^* constă dintr-o oarecare mulțime de elemente

$$D^* = \{ DA_{i,j} \mid i > 0, j > 0 \}, \quad (8)$$

unde

(i) i este indicele ramurii economice R_i ,

(ii) j este indicele domeniului de aplicație $DA_{i,j}$ în ramura economică R_i .

Intenția elaboratorului este de a realiza pe calculator un SI^* cu univers extins D^* , care include universul unității economice UE , $Univers(UE) \subset D^*$,

$$SI^* = \mathfrak{R}(D^*) = (Input^*, P^*, BD^*, Post^*, Output^*), \quad (9)$$

unde \mathfrak{R} este operatorul de realizare pe calculator a software SI.

Utilizând sistemul abstract SI^* în calitate de model, ulterior se poate genera SI pentru unitățile economice din mulțimea domeniilor de aplicație D^* .

Dacă pentru fiecare domeniu de aplicație $DA_{i,j}$ ar fi elaborat un SI, acestea ar avea forma

$$SI_{i,j} = (Input_{i,j}, P_{i,j}, BD_{i,j}, Post_{i,j}, Output_{i,j}). \quad (10)$$

Fiecare domeniu de aplicație $DA_{i,j}$ din (9), utilizând forma (1), reprezintă construcții

$$DA_{i,j} = (O_{i,j}, R_{i,j}, P_{i,j}). \quad (11)$$

Este elaborată mulțimea D^* , care include mulțimea conceptelor (obiecte, relații dintre obiecte, proceduri de transformare a obiectelor) ale uneia sau mai multor ramuri economice care includ universul unității economice UL , $Univers(UE) \subset D^*$:

$$D^* = (O^*, R^*, P^*), \quad (12)$$

unde $O^* = \bigcup_{i,j} O_{i,j}$, $R^* = \bigcup_{i,j} R_{i,j}$, $P^* = \bigcup_{i,j} P_{i,j}$ și i, j sunt, respectiv, indicii ramurilor și subramurilor (domeniilor de aplicație) preconizate realizării pe calculator.

Supunem unui proces de generalizare (4) asupra mulțimilor (12)

$$O_{generic} = G(O^*), R_{generic} = G(R^*), P_{generic} = G(P^*).$$

Elementele mulțimilor $O_{generic}$, $R_{generic}$ și $P_{generic}$ sunt modele, adică concepte abstracte ale domeniului D^* .

Elaborăm modelul $D_{generic}$

$$D_{generic} = (O_{generic}, R_{generic}, P_{generic}). \quad (13)$$

Urmează elaborarea modelului sistemului informațional $SI_{generic}$

$$SI_{generic} = \mathfrak{R}(D_{generic}) = (Input_{generic}, P_{generic}, BD_{generic}, Post_{generic}, Output_{generic}). \quad (14)$$

$SI_{generic}$ (14) poate fi folosit pentru dezvoltarea unei familii de aplicații adaptate la unitățile economice din domeniile de aplicație D^* .

$$\bar{v} = \{v_i : SI_{generic} \rightarrow SI_{i,j} \mid DA_{i,j} \in D^*\}. \quad (15)$$

O operație de dezvoltare a unui $SI_{i,j}$ pentru domeniul de aplicație $DA_{i,j}$ se efectuează cu o operație de concretizare a $SI_{generic}$ aplicând mulțimile de substituții:

$$\begin{aligned}
 &Input_{generic} \rightarrow Input_{i,j}, P_{generic} \rightarrow P_{i,j}, BD_{generic} \rightarrow BD_{i,j}, \\
 &Post_{generic} \rightarrow Post_{i,j}, Outut_{generic} \rightarrow Output_{i,j}.
 \end{aligned}$$

Codul sistemului informatic $cod_A(SI_{i,j})$ într-un oarecare alfabet A utilizat la elaborarea textelor programelor familiei sistemelor informatice (15) este

$$\begin{aligned}
 cod_A(SI_{i,j}) &= cod_A \left(\begin{array}{c} K \\ Input_{generic} \rightarrow Input_{i,j} \\ P_{generic} \rightarrow P_{i,j} \\ BD_{generic} \rightarrow BD_{i,j} \\ Post_{generic} \rightarrow Post_{i,j} \\ Output_{generic} \rightarrow Output_{i,j} \end{array} (SI_{generic}) \right) = \\
 &= cod_A((Input_{i,j}, P_{i,j}, BD_{i,j}, Post_{i,j}, Output_{i,j})).
 \end{aligned} \tag{16}$$

Sistemul informatic adaptat la condițiile unui domeniu de aplicație $DA_{i,j}$ este

$$SI_{i,j} = (Input_{i,j}, P_{i,j}, BD_{i,j}, Post_{i,j}, Output_{i,j}). \tag{17}$$

Schema obținerii unei versiuni a sistemului informatic adaptată la condițiile domeniului de aplicație $DA_{i,j}$ este reprezentată grafic în Figura 2.

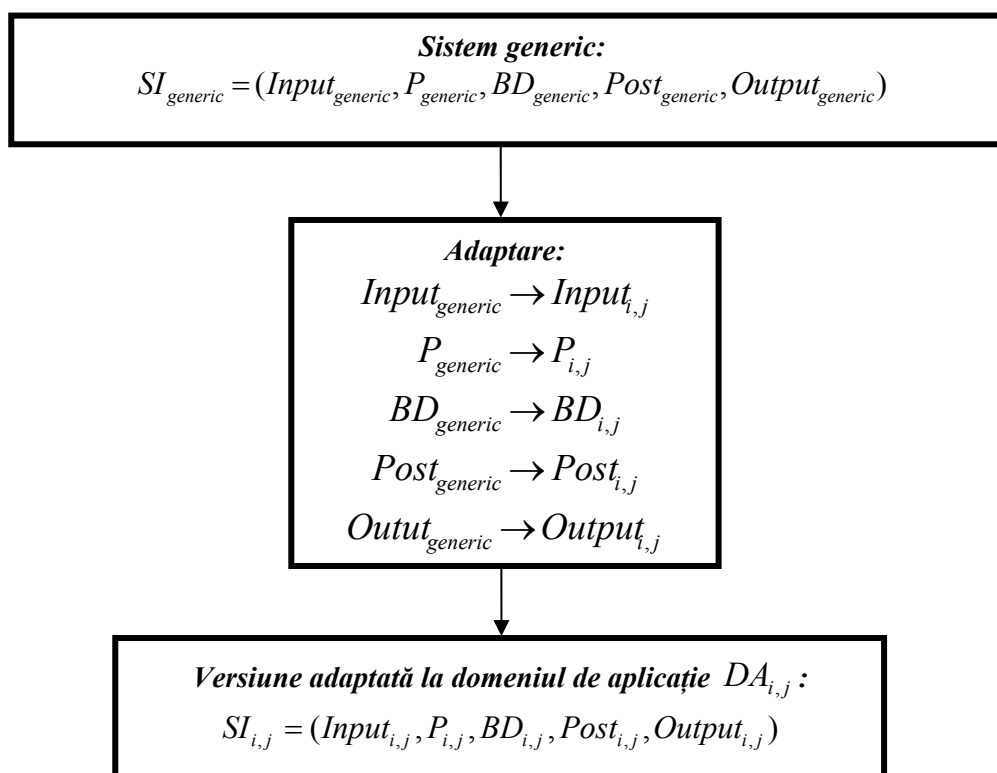


Fig.2. Dezvoltarea versiunii sistemului informatic pentru un domeniu de aplicație.

Sistemul informatic (17) mai departe poate fi rafinat pentru orice unitate economică din acest domeniu de aplicație. Procesul generării pe domenii de activitate a unei familii de SI utilizând codul SI generic este redat în Figura 3.

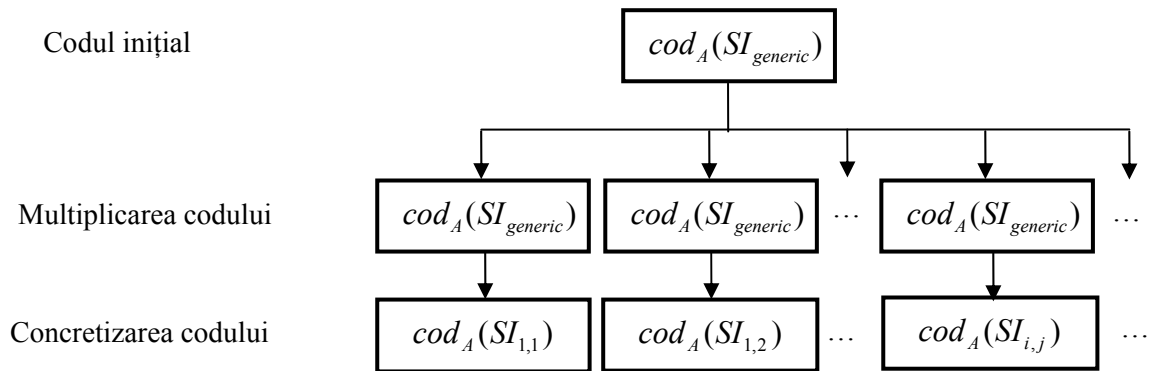


Fig.3. Generarea sistemelor informatice pe domenii de activitate.

Observăm că operația de sinteză a unei versiuni SI lucrative poate fi obținută din codul sistemului generic printr-o operație de transcripție a acestuia urmată de operații de adaptare a codului transcris la domeniul de aplicație:

$$\text{Sinteză } (SI_{UE}) = \text{Adaptare } (\text{Transcriere } (SI_{generic})).$$

7. Dezvoltarea SI specific

Tehnologia elaborării SI flexibile constă din următoarele etape:

1. Beneficiarul formulează problema de elaborare a sistemului informatic

$$SI_{UE} = (Input_{UE}, P_{UE}, BD_{UE}, Post_{UE}, Output_{UE}).$$

2. Este elaborată mulțimea conceptelor D^* , care include obiecte, relații dintre obiecte, proceduri de transformare a obiectelor ramurii economice în care activează unitatea economică UE , $Univers(UE) \subset D^*$. Dacă se preconizează implementări ale SI și în alte ramuri economice, conceptele acestor ramuri, de asemenea, se includ în mulțimea D^* :

$$D^* = (O^*, R^*, P^*),$$

unde $O^* = \bigcup_{i,j} O_{i,j}$, $R^* = \bigcup_{i,j} R_{i,j}$, $P^* = \bigcup_{i,j} P_{i,j}$ și i, j sunt, respectiv, indicii ramurilor și subramurilor (domeniilor de aplicație) preconizate realizării pe calculator.

3. Este efectuată generalizarea conceptelor mulțimii D^* :

$$O_{generic} = G(O^*), R_{generic} = G(R^*), P_{generic} = G(P^*).$$

4. Este elaborat domeniul de aplicație generic $D_{generic}$:

$$D_{generic} = (O_{generic}, R_{generic}, P_{generic}).$$

5. Este elaborat modelul sistemului informațional $SI_{generic}$, care realizează pe calculator domeniul de aplicație generic $D_{generic}$:

$$SI_{generic} = \mathfrak{R}(D_{generic}) = (Input_{generic}, P_{generic}, BD_{generic}, Post_{generic}, Output_{generic}).$$

Cel puțin o copie a $SI_{generic}$ va fi depozitată. Alte copii ale $SI_{generic}$ vor fi utilizate în procesul de elaborare a versiunilor lucrative ale SI .

6. Este elaborată versiunea sistemului informațional comandată de către beneficiar SI_{UE} .

6.1. Preventiv se transcrie $SI_{generic}$.

6.2. Copia $SI_{generic}$ obținută este supusă unui proces de adaptare:

$$\begin{aligned} cod_A \left(\begin{array}{c} K \\ Input_{generic} \rightarrow Input_{UE} \\ P_{generic} \rightarrow P_{UE} \\ BD_{generic} \rightarrow BD_{UE} \\ Post_{generic} \rightarrow Post_{UE} \\ Output_{generic} \rightarrow Output_{UE} \end{array} (SI_{generic}) \right) = \\ = cod_A ((Input_{UE}, P_{UE}, BD_{UE}, Post_{UE}, Output_{UE})) = \\ = cod_A (SI_{UE}). \end{aligned}$$

În rezultatul concretizării este obținută versiunea lucrativă a SI_{UE} comandată de către beneficiar. SI_{UE} este un 5-uplu $SI_{UE} = (Input_{UE}, P_{UE}, BD_{UE}, Post_{UE}, Output_{UE})$.

7. Pentru fiecare sinteză ulterioară a unui SI pentru alte domenii de activitate incluse în D^* sunt efectuați pașii 7.1. și 7.2.:

7.1. Preventiv se transcrie $SI_{generic}$.

7.2. Copia $SI_{generic}$ obținută este supusă unui proces de adaptare:

$$\begin{aligned} cod_A \left(\begin{array}{c} K \\ Input_{generic} \rightarrow Input_{i,j} \\ P_{generic} \rightarrow P_{i,j} \\ BD_{generic} \rightarrow BD_{i,j} \\ Post_{generic} \rightarrow Post_{i,j} \\ Output_{generic} \rightarrow Output_{i,j} \end{array} (SI_{generic}) \right) = \\ = cod_A ((Input_{i,j}, P_{i,j}, BD_{i,j}, Post_{i,j}, Output_{i,j})) = \\ = cod_A (SI_{i,j}). \end{aligned}$$

În rezultatul concretizării este obținută o versiune lucrativă a $SI_{i,j}$ pentru unitățile economice din domeniul de aplicație $DA_{i,j}$ (i este indicele ramurii economice R_i , j este indicele domeniului de aplicație $DA_{i,j}$ în ramura economică R_i). $SI_{i,j}$ este un 5-uplu

$$SI_{i,j} = (Input_{i,j}, P_{i,j}, BD_{i,j}, Post_{i,j}, Output_{i,j}).$$

8. Ciclul de viață

Produsul program $SI_{generic}$ este un model orientat pe problemă, care reprezintă un *generator de sisteme informatice* pentru o gamă de unități economice. Sistemul generic este dotat cu mai multe procesoare de generare a: intrărilor, ieșirilor, posturilor, modulelor de validare și procesare a datelor, BD, paginilor de acces la date ș.a.

Analiza și proiectarea unui produs program aplicativ orientat la probleme trece etapele de dezvoltare discutate în literatura de specialitate, de exemplu în [2, 16]. Faptul că acest tip de produse realizează orientarea pe probleme implică în ciclul de viață etapele discutate în paragraful precedent.

Ciclul de viață al unei familii de SI orientate pe probleme este reprezentat grafic în Figura 3.

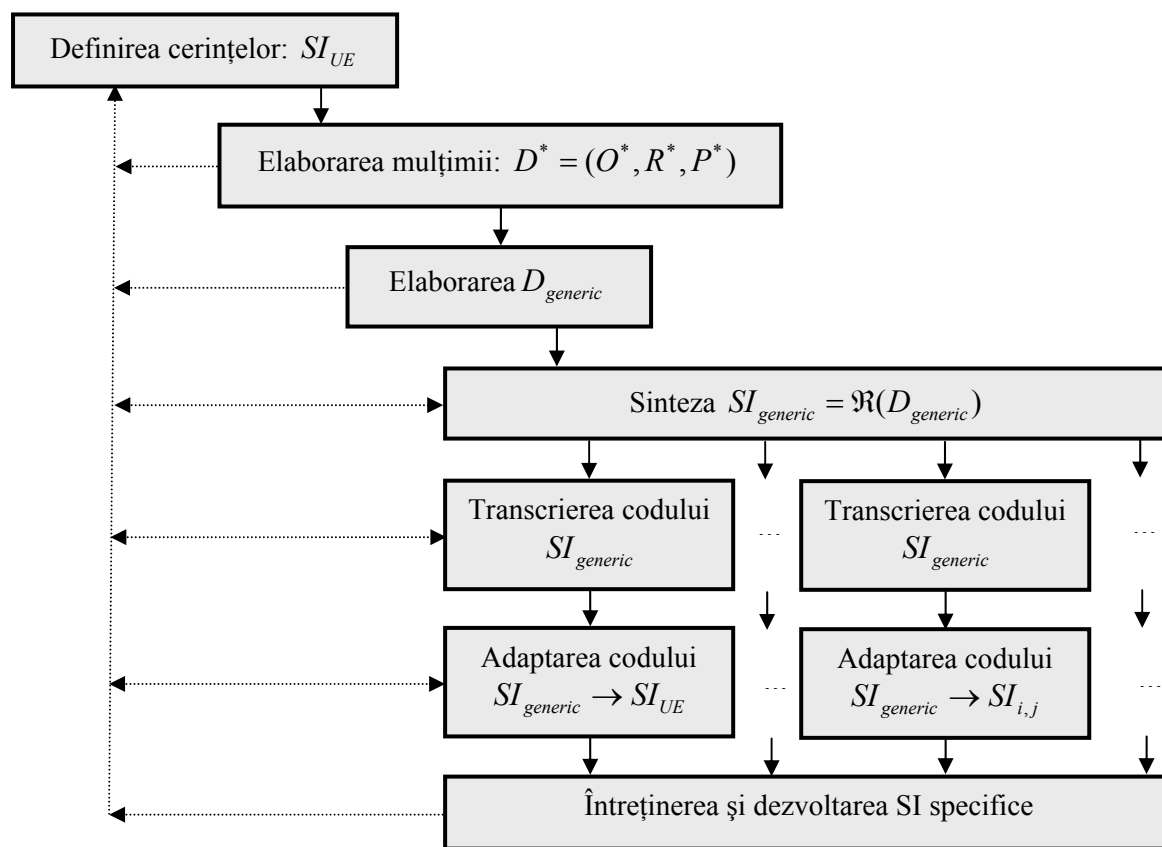


Fig.3. Ciclul de viață al unei familii de SI.

Concluzii

Dezvoltarea și adaptarea sistemelor informatice la condițiile evolutive ale mediului, tehnicile de multiplicare și personalizare a posturilor au analogie în biologie. În lucrare a fost expusă o tehnologie de proiectare a sistemelor informatice adaptive la domeniile de aplicație. Tehnologia expusă demonstrează o productivitate înaltă de elaborare a produselor program, care, de asemenea, ia în considerație metode speciale de asigurare a fiabilității și securității sistemului distribuit cu funcționare în timp real.

Referințe:

1. Alhazov A., Margenstern M., Rogozhin V., Rogozhin Yu., Verlan S. Communicative P Systems with Minimal Cooperation. - In: Lecture Notes in Computer Science, Springer Berlin / Heidelberg, Volume 3365/2005, Membrane Computing, 2005, p.161-177.
2. Bragaru T., Bulat E., Crăciun I. Dezvoltarea sistemelor informatice (suport de curs). – Chișinău: CEP USM, 2005.
3. Butnaru M., Căpățână Gh., Popov Al., Sturza G. Sistem de diagnosticare în psihiatrie. - In: Proceeding of the 5th International Conference on „Microelectronics and Computer Science”, sept. 19-21, 2007, Chișinău, Moldova. - Chișinău: UTM, 2007, p.229-230.
4. Căpățână Gh. Experiența elaborării sistemelor informatice orientate la problemă // Studia Universitatis. Seria „Științe exacte și economice”. - 2007. - Nr.2. - P.23-28.
5. Căpățână Gh., Organ A. Sisteme informatice pentru prelucrarea rezultatelor anchetării populației // Studia Universitatis. Seria „Științe exacte și economice”. - 2007. - Nr.8. - P.51-55.
6. Căpățână Gh., Seiciuc E. Limbaje naturale la proiectarea sistemelor inteligente pentru rezolvarea aproximativă a ecuațiilor integrale, aplicate la restabilirea stării în economie // Conferința Internațională „Știința, Businessul, Societatea: evoluții și intercorelări în condițiile integrării în spațiul economic european”, 12-14 februarie 2004, ASEM, Chișinău, p.58-61.
7. Cojocaru S. Interfețe inteligente pentru sisteme de calcul simbolic: Teză de doctor habilitat în informatică. – Chișinău: CNAA, 2007, <http://www.cnaa.acad.md/thesis/5844/>

8. Donciulescu D.A. Sisteme suport pentru decizie în conducerea producției: Teză de doctorat. – București: Universitatea Tehnică „Politehnica”, Facultatea de Automatică, 1998.
9. Duca M., Căpățână Gh., Levițchi A., Podduchin V. Sistem inteligent de asistență a unei clase de experimente în biologie // Conferința științifică internațională cu genericul „Învățământul superior și cercetarea – piloni ai societății bazate pe cunoaștere” dedicată jubileului de 60 ani ai Universității de Stat din Moldova, 28 septembrie 2006. Rezumatele comunicărilor. Științe reale. - Chișinău: CEP USM, 2006, p.23-24.
10. Filip F.Gh. Sisteme suport pentru decizii. Ed. a II-a, revăzută. - București: Editura Tehnică, 2007.
11. Găindric C. Luarea deciziilor: metode și tehnologii. – Chișinău: Știința, 1998.
12. Guran M., Filip F.Gh. Sisteme ierarhizate, în timp real, cu prelucrarea distribuită a datelor (PPD). – București: Editura Tehnică, 1986.
13. <http://www.dexonline.ro/>
14. Laudon K.C., Laudon J.P. Essentials of Management information systems: organization and technology. - New Jersey: Prentice Hall, 2000.
15. Mesarovic M.D., Takahara Y. General Systems Theory: Mathematical Foundations. – New York, San Francisco, London: Academic Press, 1975.
16. Oprea D. Analiza și proiectarea sistemelor informaționale economice. – Iași: Polirom, 1999.
17. Păun Gh. Computing with Membranes // Journal of Computer and Systems Science. - 2000. - No 61. - P.108-143.
18. Păvăloaia W. Analiza economico-financiară și informatică de gestiune. – Bacău: Moldavia, 2000.
19. Rădulescu D., Gheorghiu O. Optimizarea flexibilă și decizia asistată de calculator. - București: Editura Științifică, 1992.
20. Stănciulescu F. Modelarea sistemelor de mare complexitate. - București: Editura Tehnică, 2003.
21. Методические материалы по проблемно-ориентированным комплексам и потребительским комплексам (Материалы Секции специалистов №1 Совета главных конструкторов СМ ЭВМ). - Москва, 1981.
22. Толковый словарь по искусственному интеллекту / Аверкин А.Н., Гаазе-Рапопорт М.Г., Поспелов Д.А. - Москва: Радио и связь, 1992.

Notă: Cercetarea este realizată în cadrul Proiectului nr.4032, codul instituțional I1313, grupul de cercetare al Centrului Științific și Tehnologic din or. Kiev, Ucraina.

Prezentat la 28.03.2008