

## Adaptable Systems Linguistic and Processing Properties

Dumitru TODOROI, Diana MICUȘA, Zinaida TODOROI  
Academia de Studii Economice din Moldova  
[todoroi@ase.md](mailto:todoroi@ase.md)

*Metaactivity's Tools meant to the introduction of new constructions (data, operations, instructions and controls) in the language and system are language's and processor's adaptation tools. By the ACM SIGPLAN (for the Former USSR, Novosibirsk, 1983) was organized "The Adaptable Language & Systems" Special Interest Group (SIG ADAPT). President of the SIG ADAPT was elected Prof. PhD Dr Habilitatus Dumitru TODOROI.*

*The Integrity of Microsoft Products and the Adaptability of Microsoft Products are the main strategies, correspondingly, for the 2000 – 2004 and the 2005 – 2007 Microsoft Products Development periods. The development of recent publiced authors' results [ZTod-05, Mic-04, Tod-04.2] are presented. The adaptable tools definitions are developed on the base of extensible and reductible linguistic and system tools [Tod-80,83,92]. The multilanguage adaptable system is developed on the base of evolution of logic programming language LISTIC and of geometric programming language GEOGRAFIS creation [Tod-80, Mic-02,03], ADA, C++, and JAVA Technologies development.. The linguistic and system adaptable propertises [Mic-02,03] are analysed. This paper is the extenso of [Tod-05].*

**Keywords:** data, operations, adaptable tools, programming language.

### Introducere

La sfârșitul Mileniului doi și începutul Mileniului trei [Tod-98] tehnologiile informaționale și telecomunicaționale revoluționează funcționarea economiei și a societății și generează noi direcții de producere, comercializare și comunicare [Mic-03.1,03.2]. Tehnologiile informaționale au un impact esențial atât în viața cotidiană, cât și în producție. Competitivitatea viitoare a tuturor industriilor și standardele de viață a populației depind pe deplin de eforturile, depuse în cercetarea și implementarea tehnologiilor de informatizare a societății cu scopul de a pregăti generațiile viitoare de produse, procese și servicii informaționale. În aceste tehnologii informaționale calculatoarele, interfețele și rețelele vor fi integrate în activitatea zilnică și vor face accesibile, prin intermediul interacțiunilor adaptabile, simple și naturale, multiplele servicii și aplicații.

Această viziune a mediului intelectual adaptiv interactiv cere punerea utilizatorului și a bunăstării populației în centrul viitoarei soci-

etăți, bazate pe cunoștințe, în evoluția logică a societății de la societatea informațională la societatea cunoașterii [Tod-04.3, NTod-04] și de acolo la societatea conștiinței.

**Extensibilitatea** și, în continuare, **adaptabilitatea** ca idee în tehnologiile informaționale [Mic-03.1,2] și, în particular, în limbajele și sistemele de programare a apărut cu primele limbaje și sisteme orientate la calculator (prima generație): comenzi, instrucțiuni și **subprograme**, utilizate prin intermediul **definițiilor și apelurilor**. În sistemele orientate la specialist sau la un grup de specialiști (generația a 2<sup>a</sup> și a 3<sup>a</sup>) extensibilitatea este deja prezentată prin **macroui, subrutine, structuri și tipuri** cu aceeași modalitate de **definire și apel**. Sistemele orientate la obiect, probleme și apoi la utilizatori neprofesioniști (generația a 4<sup>a</sup> și a 5<sup>a</sup>) sunt caracterizate printr-un set de **facilități extensibiloadaptabile**, care pot fi divizate în facilități lingvistice și cele de sistem. Unele din aceste facilități ale procesoarelor adaptabile sunt [Mic-03.1,2] următoarele (figura1):

#### **Facilități lingvistice:**

- Universalitatea BASEI
- Compactitatea BASEI

#### **Facilități de sistem:**

- Openabilitatea
- Mobilitatea

- Modificabilitatea parțială	- Transferabilitatea
- Lizibilitatea programului	- Modifiabilitatea
- Concizia programului	- Cognibilitatea
- Crearea dialectelor	- Speciabilitatea
- Extensibilitatea	- Universibilitatea
- Reductibilitatea	- Atingerea nivelului formulării problemei
- Eficienta implementării	- Obținerea sistemului pentru limbajul natural formalizat
- Continuitatea experienței umane și a calculatoarelor	- Procesarea limbajului natural
- Multilingvibilitate	

**Fig. 1.** Facilități lingvistice și de sistem

Viabilitatea acestor facilități este totalmente în viziunea lucrării de față și din punct de vedere teoretic și din punct de vedere practic. Primele forumuri internaționale consacrate în special limbajelor și sistemelor extensibile [Pro-69,71] au evidențiat proprietățile și facilitățile limbajelor și sistemelor extensibile și au accentuat problemele, care apar în această direcție de cercetări și implementări. Prima generalizare a rezultatelor obținute a fost efectuată de Solhseff și Yezerski [Sol-73] din punct de vedere al divizării mijloacelor extensibile în dependență de fazele de translatare [Tod-67]. În țările FUS și în continuare CSI, în anii 1982-1993 a activat o grupă specială de interese “Sisteme adaptabile de programare” [Cuz-89,Tod-87], care la întrunirile sale anuale analizau starea de rezolvare a problemelor extensibilității și adaptabilității din punct de vedere lingvistic, sistemic și realizatorico-pragmatic [Tod-92]. Ultimii 18 ani la forumurile anuale internaționale OOPSLA-1987 - OOPSLA-2004 problemele adaptabilității sunt intensiv abordate ca mijloace ale programării adaptabile. Realizările C++, Ada și Java sunt confirmări practice ale teoriei și practicii programării adaptabile.

### 1. Adaptabilitate și extensibilitate.

Mijloacele metalingvistice, destinate introducerii de noi construcții (date, operații, instrucțiuni, dirijări) în limbaj se numesc **extinderi de limbaj**.

Aceste extinderi sunt elemente de limbaj împreună cu elementele de bază a limbajului. Extinderea de limbaj este reprezentată prin **definitorul** extinderii și prin **apelul extinderii**.

Definitorul de extindere permite extinderea limbajului cu noi elemente derivate ale limbajului. El realizează definirea extinderii.

Definitorul de extindere la fel ca și definitorul de reducere (individualizare) include:

- definirea pragmaticii elementului - extindere,
- definirea sintaxei elementului-extindere,
- definirea semanticii elementului - extindere,
- definirea contextului de utilizare a elementului-extindere,
- exemple de apel a elementului-extindere.

**Adaptorul** este compus din definitorul de extindere (Adaptor-extendor) și definitorul de reducere (Adaptor-reductor).

**Schema generala a adaptorului-extendor este următoarea:**

- \_BL\_ pragmaticele elementului
- \_SY\_ sintaxa elementului
- \_SE\_ semantica elementului
- \_CO\_ contextul de utilizare a elementului
- \_EX\_ exemple de apel a elementului
- \_EL\_

Din definiție reiese că părțile componente ale definiturii elementului-extindere sunt pragmaticele, sintaxa, semantica, mediul și exemplele noului element definit.

**Pragmatica identifică** locul, necesitatea extensiei în limbaj, explică necesitatea includerii în limbaj a elementului nou. **Sintaxa definește** forma de apel a extensiei –elementului nou. **Semantica reprezintă** descrierea conținutului elementului nou prin intermediul conținutului altor elemente deja introduse în limbaj. **Contextul apreciază** locul de utilizare a acestor elemente derivate, ierarhia lor în cadrul elementelor limbajului. **Exemplele**

**conțin** exemple de alel și de utilizare a elementului nou introdus.

**Adaptorul-reductor** conține și informația despre tipurile procesoarelor adaptabile, care realizează elementele derivate incluse în sistemul adaptabil individual.

Biblioteca, care conține definiții ale extensiei se numește **biblioteca extensibilă**. Ea este organizată sub forma de lista bidimensională.

**Apelul** extensiei permite utilizarea noului element la programarea problemelor în limbajul extensibil. **Sintaxa apelului** este definită de partea sintactică al definatorului de extensie.

**Pragmatica și semantica** apelului de extensie definesc necesitatea introducerii noului element în sistemul adaptabil, sensul și eficiența utilizării acestei extensii.

**Limbajul adaptabil** este limbajul compus din: elemente de bază, elemente derivate, extindori și reductori.

Limbajul compus doar din extindori, reductori și elemente de bază se numește **baza adaptabilă** (baza limbajului extensibil).

Programele scrise în limbaj extensibil se numesc **programe extensibile**.

Biblioteca compusă din aceste programe extensibile este numită **biblioteca programelor extensibile**.

Elementele de translatore a extinderilor se numesc **implementatori de extensii**. Implementatorul este compus din două părți:

- a) implementatorul definirii extensiei,
- b) implementatorul apelului extensiei.

**Implementatorul definirii extensiei** este un element translatoric a sistemului adaptabil de programare care:

- (1) identifică extensia,
- (2) compilează și/sau interpretează definiția de extensie,
- (3) contribuie la fixarea elementelor-extensii în sistemul adaptabil de programare,
- (4) pregătește acest element, introdus de definator, pentru a fi utilizat.

Implementatorul definirii extensiei este reprezentat printr-un modul sau șir de module translatorice legate într-un lanț. El este partea componentă a subsistemelor de definiții și fixare din sistemul de programare adaptabilă.

**Implementatorul apelului extensiei** este un modul translatoric (sau un lanț de module translatorice) al sistemului de programare adaptabilă, care identifică apelul extensiei în diferite forme în corespondență cu definirea sintaxei acestei extensii, traduce (compilează sau interpretează) acest apel prin intermediul elementelor de niveluri mai joase (sau recursiv prin intermediul elementelor de același nivel). Este un modul sau un lanț de module translatorice.

**Procesorul adaptabil** este un translator, care realizează limbajul adaptabil. El este compus din: (1) modulele de realizare a elementelor de bază, (2) modulele utilitare de guvernare, (3) modulele de realizare a definatorilor extensiilor și a apelurilor lor, (4) modulele de completare a programelor obiecte produse.

**Sistemul de programare** este adaptabil, dacă este compus din limbaj și procesor extensibil.

**Produsul-program** este adaptabil, dacă este compus din: (1) extindori, (2) programe extensibile, (3) limbaj adaptabil, (4) procesor adaptabil, (5) sistem de programare adaptabil, (6) tehnologii de creare, fixare în sistem și utilizare a extensiilor.

**Tehnologia de programare** este adaptabilă dacă garantează crearea, guvernarea și utilizarea produselor-program extensibile în situații operaționale ale sistemului de programare extensibil.

**Mijloacele de programare** sunt adaptabile, dacă garantează: (a) crearea, (b) guvernarea și (c) utilizarea produselor – program extensibile.

**Mijloacele de programare** adaptabile includ: (1) rețeaua de calcul, (2) sistemul adaptabil de programare, (3) tehnologia adaptabilă de programare, (4) biblioteca extensibilă și (5) biblioteca de programare extensibilă.

## 2. Metodele de obținere a limbajului adaptabil

Cercetările, efectuate în cadrul obținerii sistemelor multilingve în construirea sistemelor graficii pe calculator [Tod-04.1,04.2,80.2], au dat posibilitatea de a obține diferite tipuri de decizii de adaptare, evaluate pentru metodele model-realizare de tip Top-Down,

Bottom-Up, Horizontal și mixt.

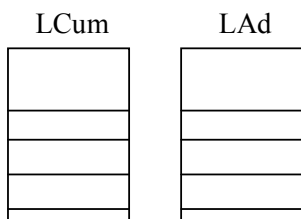
Metoda de obținere a limbajului adaptabil multilingv constă în:

- (1) ierarhizarea elementelor limbajelor, evidențiate pentru obținerea limbajului multilingv,
- (2) introducerea elementelor “celui mai reprezentativ limbaj” în baza limbajului multilingv,
- (3) adaptarea limbajului ierarhizat la limbajul multilingv prin comparare elementelor ierarhizate a acestor limbaje și
- (4) includerea elementelor limbajului ierarhizat în limbajul multilingv după metodele: model-realizare de tip Top-Down, Bottom-Up, Horizontal sau mixt.

Evaluarea: Fie dată baza limbajului multilingv (LCum: Limbaj cumulativ). Elementul limbajului ierarhizat (LAd: Limbaj adăugat) este comparat cu elementele de diferit nivel  $i+1$ ,  $i$  și  $i-1$  a limbajului cumulativ.

Sunt evidențiate mai multe situații:

- (A) Nivelurile elementelor de bază ale LCum și LAd coincid (Fig.2. Modelul Horizontal);
- (B) Nivelul elementelor de bază al LCum este mai mic decât nivelul elementelor de bază al LAd (Fig.3. Modelul Top-Down);
- (C) Nivelul elementelor de bază al LCum este mai mare decât nivelul elementelor de bază al LAd (Fig.4. Modelul Bottom-Up)



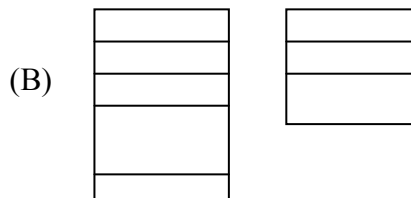
LCum – limbaj cumulativ, LAd – limbaj adăugat. Nivelurile elementelor de baza al LCum și LAd coincid

Fig.2. Modelul Horizontal

Cercetarea fiecărui element din LCum și LAd se efectuează după **metoda de coincidență/necoincidență** [Tod-80, Mic-04, Tod-04.1,04.2] a sintaxei (SY), a semanticii (SE) și a transcripției (RT) lor. În dependență de corelarea elementelor respective se obține

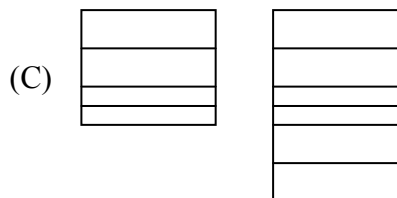
limbajul adaptabil multilingv corespunzător după modelul Horizontal, Top-Down, Bottom-Up și/sau mixt. Evaluarea LCum se bazează pe cercetarea următoarelor situații:

- (1) LCum (SY + SE + TR) LAd :: (Elementul limbajului LCum are aceeași sintaxă (goto M --- goto M), semantică și transcripție ca și elementul din LAd);



(B) Nivelul elementelor de bază al LCum este mai mic decât nivelul elementelor de bază al LAd

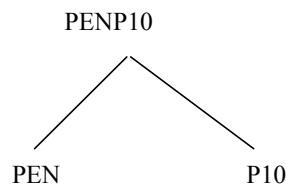
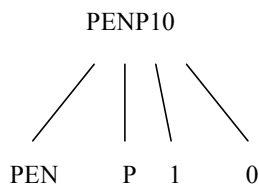
Fig.3. Modelul Top-Down



(C) Nivelul elementelor de bază al LCum este mai mare decât nivelul elementelor de bază al LAd

Fig.4. Modelul Bottom-Up

- (2) LC (SY + SE +  $\gamma$ TR) LA :: (Elementul limbajului LCum are aceeași sintaxă (GOTO M --- goto M) și semantică dar transcripție diferită cu elementul din LAd);
- (3) LC (SY +  $\gamma$ SE + TR) LA:: (Elementul limbajului LCum are aceeași sintaxă și transcripție, dar semantica diferă (A\*B+C din limbajul SIRIUS cu semantica: A rotit în jurul lui B și translatat cu C) --- A\*B+C din limbajul PASCAL (cu operații aritmetice de înmulțire și adunare) de elementul din LAd);
- (4) LC ( $\gamma$ SY +  $\gamma$ SE + TR) LA :: (Elementul limbajului LCum are sintaxă și semantică diferite cu elementul din LAd, dar aceeași transcripție): PENP10 din limbajul GRAFIC (penița neagră cu grosimea 1 și forma punctată) și PENP10 din limbajul GEOMAL (penița este poziționată în punctul P10):



(5) LC (SY +  $\gamma$ SE +  $\gamma$ TR) LA:: (Elementul limbajului LCum are semantica și transcripția diferite cu elementul din LAd, dar aceeași sintaxă): A rot B tran C din limbajul GRAFOR se compară cu A \* B + C din limbajul JAVA, cu care au același arbore sintactic;

(6) LC(( $\gamma$  SY) + SE +  $\gamma$ TR):: (Elementul limbajului LCum are sintaxa și transcripția diferite cu elementul din LAd, dar aceeași semantică): FOR i = B BY C TO D; S; END din limbajul PERL :: for i:= a step B UNTIL C do S din limbajul ALGOL-68;

(7) LC ( $\gamma$ SY + SE + TR) LA :: (Elementul limbajului LCum are aceeași semantică și transcripție dar sintaxă diferită cu elementul din LAd);

(8) LC ( $\gamma$ SY +  $\gamma$ SE +  $\gamma$ TR) LA :: (Elementul limbajului LCum nu coincide după sintaxă, semantică și transcripție cu elementul din LAd).

**3. Metodele de obținere a procesoarelor adaptabile**

Procesoarele adaptabile sunt repartizate pe trei niveluri de complexitate transla-torică, în baza carora este situată reprezentarea modulară și recursivă [DTod-67]. Nivelul unu [DTod-03] îl reprezintă procesoarele adap-tabile de tip Timp-Realizare al extinderelor: Pre-procesoarele, Inter-procesoarele și Post-procesoarele. De acelaș nivel unu aparțin și procesoarele adaptabile, construite după metodele Model-Realizare ale extinderelor: rea-lizarea extinderilor de tip Nivel-Nivel, Nivel-Direct și Nivel-Nivel-Direct. Al treilea set de procesoare adaptabile de nivelul unu îl con-stituie procesoarele adaptabile de tip Tip-Traducere adaptabilă: Compilare adaptabilă, Interpretare adaptabilă și Compilare-Interpretare adaptabilă a extiderilor. Aceste

procesoare formează baza cubului adaptabili-tății (Fig. 5).

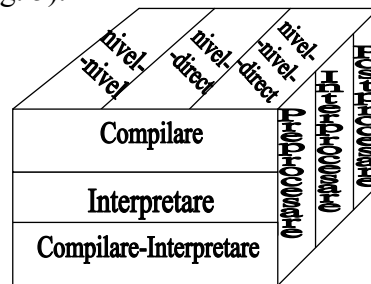


Fig. 5. Cubul adaptabilității

Procesoarele adaptabile de nivelul unu de tip Nivel-Nivel realizează extinderile prin meto-da « de coborîre » de pe un nivel al limbaju-lui adaptabil de programare (LAP) pe altul până la atingerea nivelului bazei limbajului adaptabil de programare (BLAP). Procesoa-rele adaptabile de tip Nivel-Direct « coboa-ră » elementele-extideri din programul scris în LAP transformându-l direct în programul scris în limbajul BLAP. Procesoarele adapta-bile de tip Nivel-Nivel-Direct sunt create în baza modulelor translatorice de procesare de tip Nivel-Nivel și Nivel-Direct. Nivelul doi [ZTod-03] de procesoare adaptabile îl re-prezintă procesoarele de tip Timp-Model-Realizare, Tip-Timp-Realizare și Tip-Model-Realizare a extiderelor. Ele formează al doi-lea nivel de complexitate transla-torică din cubul adaptabilității și reprezintă combinații de procesoare de nivelul unu de tip Timp-Realizare și Model-Realizare a extinderilor și de Tip-Traducere adaptabile [DTod-03]. Procesoarele de nivelor doi sunt prezente prin intermediul procesoarelor adaptabile de tip Timp-Model-Realizare a extinderilor (Ta-belul 1.), de tip Tip-Model-Realizare a extinderelor(Tabelul 2.) și de tip Tip-Timp-Realizare a extinderilor (Tabelul 3.).

**Tabel 1.** Procesoare adaptabile de tip Timp-Model-Realizare a extinderelor.

	Nivel-Nivel	Nivel-Direct	Nivel-Nivel-Direct
Pre-procesare	Preprocesor-N-N	Preprocesor-N-D	Preprocesor-N-N-D
Inter-procesare	Interprocesor-N-N	Interprocesor-N-D	Interprocesor-N-N-D
Post-procesare	Postprocesor-N-N	Postprocesor-N-D	Postprocesor-N-N-D

**Tabel 2.** Procesoare adaptabile de tip Tip-Model-Realizare a extinderelor.

	Nivel-Nivel	Nivel-Direct	Nivel-Nivel-Direct
Compilare	Compiler-N-N	Compiler-N-D	Compiler -N-N-D
Interpretare	Interpreter-N-N	Interpreter-N-D	Interpreter -N-N-D
Compilare-interpretare	Interpreter- Interpreter-N-N	Interpreter- Interpreter-N-D	Interpreter- Interpreter-N-N-D

**Tabel 3.** Procesoare adaptabile de tip Tip-Timp-Realizare a extinderelor.

	Preprocesare	Interprocesare	Postprocesare
Compilare	Compiler-Preprocesor	Compiler-Interprocesor	Compiler-Postprocesor
Interpretare	Interpreter- Preprocesor	Interpreter-Interprocesor	Interpreter – Postprocesor
Compilare-interpretare	Compiler-Interpreter-Preprocesor	Compiler-Interpreter-Interprocesor	Compiler-Interpreter-Postprocesor

Sunt demonstrate procesele de obținere automatizată a procesoarelor adaptabile de nivelul doi de tip Timp-Model-Realizare a extinderelor. Demonstrarea obținerii automatizate a celorlalte procesoare de nivelul doi va fi prezentată în următoarele publicații.

Procesoare adaptabile-compiletoare de nivelul trei [Mic-03] de complexitate translativă din cubul adaptabilității sunt prezentate prin intermediul Pre-procesoarelor-Compiletoare adaptabile, Inter-procesoarele-Compiletoare adaptabile și Post-procesoarele -Compiletoare adaptabile

#### 4. Facilități adaptabile lingvistice și de sistem

Sistemele adaptabile, create în baza sistemelor orientate-obiect, orientate-acțiuni și orientate-utilizatori neprofesioniști (software-ul [Mic-03.1,03.2] de generația a 4<sup>a</sup> și a 5<sup>a</sup>), sunt caracterizate [Tod-80, Mic-04, Tod-04.1,04.2] printr-un set de **facilități extensibilo-adaptabile**, care pot fi divizate în facilități lingvistice și cele de sistem. Unele din facilități ale procesoarelor adaptabile sunt prezentate în Fig.1.Facilitățile extensibilo-adaptabile lingvistice și de sistem **sunt ușor de demonstrat.**

##### 4.1. Facilități lingvistice

Limbajul adaptabil are proprietatea de a fi **universal** [Tod-80,83.2, Mic-04, Tod-04.1,04.2] prin procesul de extensie a elementelor în direcția datelor, operațiilor, instrucțiunilor și dirijărilor. Se pot obține astfel proprietăți universale prin extinderea bazei în diferite direcții. Suma acestor extinderi conduce la universalitatea limbajului adaptabil.

**Specializarea** (Reductibilitatea) este atinsă

[Tod-80,83.2, Mic-04, Tod-04.1,04.2] prin intermediul elementelor adaptabile de reducere. Ea este efectuată în baza limbajului adaptabil, care a fost extins deja în diferite direcții. Acest limbaj este universal și deci utilizează toate aceste direcții de extindere. Specializarea este similară cu obținerea limbajului pentru o direcție nouă de utilizare a limbajului adaptabil care poate fi:

- o intersecție a dialectelor de extindere a limbajului de bază,
- o colecție de elemente din diferite dialecte ale limbajului universal,
- o colecție de elemente de alt tip, nedefinite în sistem.

În primele două cazuri se efectuează doar reducerea propriu zisă. În ultimul caz se utilizează metoda de reducere și extindere a sistemului adaptabil individual.

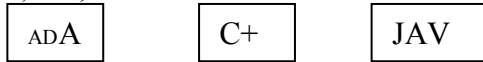
**Facilitatea de modificare a componentelor** este o posibilitate comună extinderii și reducerii. Ea permite (prin intermediul extinderii și reducerii) adaosul unui element nou, schimbarea unui element cu altul, modificarea unui element deja existent în sistem prin schimbul sintaxei, semanticii, contextului și/sau a pragmaticii sau prin adaosul unei noi posibilități a acestui element.

**Facilitatea de a crea un dialect al limbajului.** Un dialect al limbajului este compus din elemente de bază și din elemente necesare pentru utilizarea limbajului într-o anumită direcție. Se poate realiza prin extindere definind elementele noi sau prin reducere, alegând diferite elemente din nivelurile superioare ale limbajului extins, formând un dialect nou.

**Flexibilitatea** constă în posibilitatea limbajului de a fi extensibil (prin intermediul extensiei) sau redus (prin intermediul reducerii).

**Multilingvibilitatea** limbajelor:

1) Inițial:



2) Se obțin elemente necesare din diferite limbaje în același limbaj adaptabil:

$$ADA + C++ + JAVA = LA$$

**Eficacitatea realizării** se sprijină pe aceea, că baza este suficient de bine definită și extinderile se pot realiza automat prin intermediul realizatorului extendorului (fiecare extindere a limbajului adaptabil de bază constituie un anumit dialect).

**Transmiterea experienței:** este făcută de extensibilitate prin intermediul utilizării elementelor deja incluse în limbajul adaptabil la definirea unui element nou al acestui limbaj.

**Facilitatea apropiării de limbajul natural** se bazează pe definirea elementelor unui nivel prin intermediul elementelor bazei, elementelor nivelurilor mai inferioare și recursiv; se ridică nivelul limbajului până la nivelul limbajului natural formalizat.

**4.2. Facilități de sistem**

**Operabilitatea** (Sistem deschis). Prin intermediul extinderii și reducerii pot fi ușor adăugate [Tod-80,83.2,04.1,04.2, Mic-04] module noi translatorice, neschimbând translatorul de baza. Aceste module organic se implantează în translator prin intermediul subsistemelor “declarare” și “fixare” la extindere și prin intermediul subsistemului “utilizare” la utilizarea acestui element. Cu utilizarea subsistemului “reducere” se obține sistemul

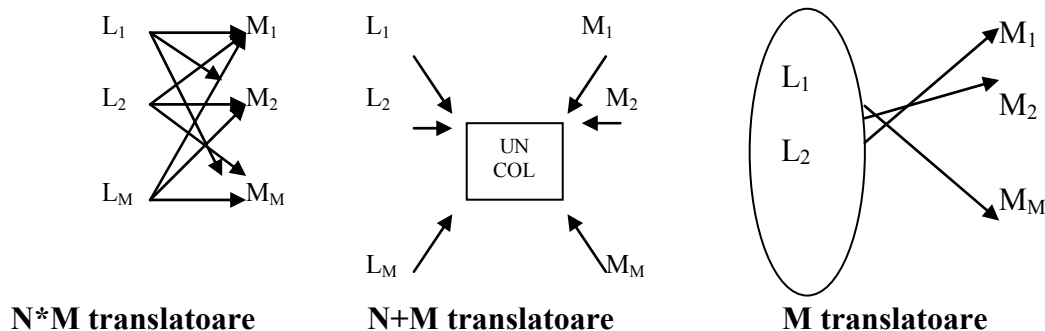
specializat.

**Mobilitatea.** Sistemele adaptabile, create în baza ideii de programare modulară și recursivă [Tod-67,87,04.2] pot fi ușor transferate de la un calculator la altul fiindcă este suficient realizarea doar a bazei la calculatorul nou și automat se primesc dialectele acestui limbaj. Sistemele noi iau elementele din limbajul adaptabil existent și le adaugă la translatorul calculatorului nou prin intermediul subsistemelor de declarare, fixare, utilizare și reducere a componentelor împreună cu realizatorul bazei.

**Modifiabilitate.** Datorită translatorului realizat prin diferite module și lanțuri de module translatorice, aceasta structură este ușor de modificat. Este o facilitate de caracter a sistemului, compus din module, inclusiv recursive [Tod-67,87,04.2].

**Speciabilitatea și universabilitatea procesorului adaptabil.** Procesorul compus din translatorul bazei și modulele realizatorice a unui dialect este un translator specializat care împreună cu limbajul specializat creează sisteme specializate adaptabile. Reuniunea translatorului bazei și a diferitelor dialecte conduce la un translator universal al limbajului extensibil universal.

**Reducerea numărului de procesare** (Fig.6). Fie date N limbaje și M mașini. În mod nominal pentru utilizarea unui limbaj pe mașini diferite trebuie translatorice diferite. Deci sunt necesare N\*M translatorice. Sistemele UNCOL și ALMO au redus numărul translatoricelor până la N+M.



**Fig.6.** Evaluarea adaptabilă a procesoarelor.

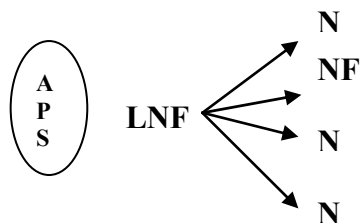
Prin unirea limbajelor într-unul singur - multilingvajul adaptabil- se reduce numărul

translatoricelor până la M translatorice. Este oare posibil de redus numărul transla-

toarelor până la unu?

### Atingerea nivelului formulării problemei.

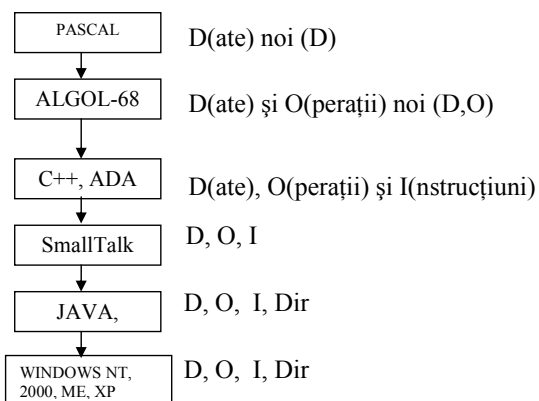
Interacțiunea cu sistemele ce pot realiza la nivelul formulării problemei:



1. Formularea .....limbaj natural (N)
2. Formalizarea...limbaj natural formalizat (LNF)
3. Algoritmizarea.....algoritmi
4. Programarea.....programe
5. Testarea.....limbaj natural formalizat
6. Documentarea... limbaj natural
7. Menținerea..... limbaj natural

### 5. Extensibilitate și Adaptabilitate. Evaluare.

Prin intermediul adaptorilor s-au realizat [Tod-92, Mic-04, Tod-04.1] o serie de proiecte și sisteme adaptabile și extensibile reale și/sau experimentale, care se caracterizează prin extinderea seturilor de D(ate) și/sau O(perații) și/sau I(nstrucțiuni) și/sau de Dir(ijări) noi:



Dirijările interne sunt obținute prin noi facilități de dirijare în cadrul unui program. Acestea sunt dirijările susținute de ierarhia operațiilor la executarea unei expresii. Dirijările instrucțiunilor sunt efectuate de instrucțiunile-dirijări de tipul ramificări, cicluri, adresări la subrutine, proceduri etc. Dirijările noi între diferite programe și între mașini le constituie metodele de multiprogramare și multiproce-

sare, executarea virtuală a programelor, sistemele de divizare a timpului etc.

Un șir de sisteme adaptabile, extensibile, reductibile au fost realizate în perioada din anul 1970 până în prezent [Tod-80,83,87,92,Sol-74]. cu concursul multor alte metode de interacțiuni translatorice. O bibliografie imensă de realizări ale sistemelor orientate obiect, orientate agent și adaptabile, care sunt bazate pe ideea extensibilității și adaptabilității o constituie numirea lucrărilor, publicate în cadrul ultimelor 20 de ani de conferințe anuale OOPSLA.

### 6. Concluzii

Sunt cercetate sistemele (procesoarele și limbajele) de programare tradiționale. Nucleul sistemelor de programare adaptabilă îl constituie adaptorii - mijloacele metalingvistice, compuse din extendori și reductorii. În baza acestor cercetări și implementări au fost obținute următoarele rezultate:

- (1) Este realizată experimental una din multiplele scheme realizatorice ale cubului adaptabilității (Fig.4), prezentată în formă de sistem adaptabil SAGRED3D, care se utilizează în procesul de învățământ cu studenții facultății de matematică și informatică a U.S.M. (Chișinău) și cu studenții facultății de informatică a Universității "Al.I.Cuza" (Iași).
- (2) Sunt cercetate integral facilitățile extensibil-adaptabile ale mijloacelor adaptabile din punct de vedere lingvistic și de sistem;
- (3) Sunt cercetate și analizate funcțional schemele de preprocesor, interprocesor și postprocesor ca reprezentanți principali pe una din dimensiunile cubului adaptabilității; o altă dimensiune din acest cub de procesoare adaptabile cercetate și analizate o constituie procesoarele: Nivel-Nivel, Nivel-Direct și Nivel-Nivel-Direct;
- (4) Procesul de extindere este reprezentat prin intermediul sistemelor de Declarație, Fixare și Utilizare. Procesul de reducere este prezentat de sistemul de Reducere. În teză aceste subsisteme sunt analizate din punct de vedere a creării procesoarelor adaptabile de diferit tip. Procesele de extindere și reducere împreună formează procesul de adaptare;
- (5) Au fost formulate și demonstrate [Mic-



03.1] teoremele de obținere automatizată a procesoarelor adaptabile de nivelul (5.1) unu (de tip model-realizare: Nivel-Nivel, Nivel-Direct și Nivel-Nivel-Direct, de tip timp-realizare: preprocesor, interprocesor și postprocesor și de tip de traducere: compilator, interpretor și compilator-interpretor), (5.2) doi (de tip Timp-Model-Realizare) și (5.3) trei (de tip Timp-Model-Realizare cu compilare) de complexitate translatorică. S-a obținut fundamentarea teoretică a algoritmilor propuși de soluționare a problemelor cercetate.

(6) Un set important de probleme, aflate la etapa de soluționare în vederea evaluării programării adaptabile îl constituie:

6.1. Timpul (metodele timp-realizare) de realizare a extensiilor: metodele de preprocesare, interprocesare, postprocesare și/sau mixt;

6.2. Tipul de traducere adaptabilă: metodele de compilare, interpretare și/sau mixt;

6.3. Modelul de adaptare multilingvistică: metodele Top-Down, Bottom-Up, Horizontal și/sau mixt;

6.4. Metoda model-realizare a extensiilor: Nivel-Nivel, Nivel-Direct, Nivel-Nivel-Direct și/sau mixt.

## Bibliografie

[Tod-05] Dumitru TODOROI, Diana MICUSA, Zinaida TODOROI. **Adaptable Languages and Systems Properties**. // Proc. of the 7<sup>th</sup> Int. Conf. on Informatics in Economy: "Information & Knowledge Age", May 19-20, 2005, p. 349-355.

[ZTod-05] Zinaida Todoroi (AESM, Republic of Moldova), Diana Micusa (Moldova State University, Republic of Moldova), Manfred Kudlek (Hamburg University, Germany), Dumitru Todoroi (AESM, Republic of Moldova). Crearea automatizată a procesoarelor adaptabile de **nivelul doi** bazate pe nivelul unu cu utilizarea formalismul E-T-M de interacțiune translatorică a procesoarelor adaptabile. // Informatica Economica, Vol. IX, Num. 1/2005, p.47-55.

[Tod-04.1] Dumitru TODOROI, (AESM, Republic of Moldova), Manfred Kudlek (Hamburg University, Germany), Zinaida Todoroi (AESM, Republic of Moldova), Diana Micusa (Moldova State University, Republic of Moldova). Crearea **primului nivel** de procesoare adaptabile utilizând formalismul E-T-M de interacțiune translatorică a procesoarelor adaptabile. // Informatica Economica, Vol. VIII, Num. 3/2004, p. 108-116.

[Mic-04.1] Diana Micusa (Moldova State University, Republic of Moldova), Manfred Kudlek (Hamburg University, Germany), Dumitru Todoroi (AESM, Republic of Moldova), Zinaida Todoroi (AESM, Republic of Moldova). Crearea automatizată a procesoarelor adaptabile de **nivelul trei** în baza formalismului E-T-M de interacțiune translatorică a

procesoarelor adaptabile. Vol. VII, Num. 2/2003, p.103-109. [Tod-04.2] Dumitru TODOROI, Zinaida TODOROI, Diana MICUSA. Adaptable processors-compilers' construction and automation. // Proc. of the 29th Annual Congress of the ARA of Sciences and Arts (ARA), Bochum, Germany, September 7-12, 2004, p. 499-504.

[Tod-04.3] D Todoroi, D. Micusa. Sisteme adaptabile. Definitii. Multilingvibilitatea. // Simp. International « Integrarea europeana si competitivitatea economica », 23-24 septembrie 2004, Vol. II, Chisinau 2004, p. 217-220.

[Mic-04.2] D. Micusa, D Todoroi. Sisteme adaptabile. Proprietati lingvistice si de sistem. // Simp. International « Integrarea europeana si competitivitatea economica », 23-24 septembrie 2004, Vol. II, Chisinau 2004, p. 221-223.

[Mic-03] D. Micușă. The Adaptability Cube' Third Level of Translation Complexity' Adaptable Processors – Compilers. // Proc. of the 6th International Conference on Economic Informatics „Digital Economy”, May 8 – 11, 2003, Bucharest, p.93– 99.

[Tod-03] Dumitru TODOROI. The Model Type of Extensions' Realization of the Adaptable Cobe' First Level of Translation Complexity' Adaptable Processors. // Proc. of the 6<sup>th</sup> International Conf. IE'03 "Digital Economy", Ed. Economica, Bucharest, May 8-11, 2003, p. 113-121.

[Tod-03] Zinaida TODOROI. The Second Level Adaptable Processors of he Extensible Time-Models Realiation. // Proc. of the 6<sup>th</sup> International Conf. IE'03 "Digital Economy", Ed. Economica, Bucharest, May 8-11, 2003, p. 343-348.

[Mic-02] D. Micușă. Computerized Natural Language Processing Systems Adaptable Tools. // Proc. of the 27th Annual Congress of ARA, Volume 1, Politechnic International Press, Canada, Montreal, May 29 – June 2, 2002, p.504 – 507.

[Tod-98] D. Todoroi, S. Nazem, T. Jucan, D. Micușă. Transition To A Full Information Society: Stage Development. // Working Paper No. 98-2, UNO, Omaha, USA, March 1998. - 38 p.

[Tod-92] D. Todoroi. Computer Science. The Adaptable Programming. The Basic Conceptions. // ASEM Press, Chisinau, 1992. - 76 p. (Eng.)

[Cuz-89] Кузнецов С., Тодорой Д.. О школе семинаре «Расширяемые средства программирования. Методы оценки трансляторов». // Программирование. 1989. №6. стр.91-92.

[Tod-87] Тодорой Д., Челбаков Б. Целевая программа «Расширяемые средства программирования». // Программирование. 1987, №3, стр. 91-92.

[Tod-83] D. Todoroi. The Extensible Computer Graphics Tools. // Radio and Telecommunication Press, Moscow, 1983. - 208 p. (Rus.)

[Tod-80] D.Todoroi. The Extensible Computer Graphics Systems. // Science Press, Chisinau, 1980, 199 pages. (Rus.)

[Sol-74] N. SOLNTSEFF, A. YEZERSKI. A Survey of Extensible Programming Language. // ARAP, 1974, 4, p.267-307.

[Pro-71] Proceedings of Extensible Languages Symposium. // SIGPLAN Notices. December, 1971, 6, Nr.12.

[Pro-69] Proceedings of the Extensible Languages Symposium (Boston, May, 1969) Eds C. Christiansen. // SIGPLAN Notices, 1969, v.4, Nr.8.