

Șef lucrări ing. Zsongor F. GOBESZ

**CONTRIBUȚII LA DEZVOLTAREA
SISTEMELOR EXPERT ÎN DOMENIUL
CONSTRUCȚIILOR**

- TEZĂ DE DOCTORAT -

**Conducător științific:
Prof.dr.ing. Alexandru CĂTĂRIG**

MULȚUMIRI

Întrucât această lucrare încearcă să reflecte preocupările mele din ultimii ani, dedicați într-o mare măsură cercetărilor legate de aplicarea sistemelor pe bază de cunoștințe în domeniul construcțiilor din țara noastră, ea nu se putea realiza fără sprijinul și ajutorul unor persoane, cărora țin să-mi exprim recunoștința și pe această cale.

În primul rând țin să mulțumesc domnului prof. Alexandru Cătărig, conducătorului științific, pentru ajutorul său, spiritul său critic, sugestiile valoroase pe care mi le-a făcut și nu în ultimul rând pentru răbdarea de care a dat dovadă, în special în ultimii doi ani.

Doresc să mulțumesc domnilor prof. Marian Ivan (U.P. din Timișoara), prof. Ioan Ciongradi (U.T. din Iași) și prof. Ludovic Kopenetz (U.T. din Cluj-Napoca) pentru că au acceptat sarcina de a fi referenți. În plus, doresc să mulțumesc domnului prof. Ciongradi și pentru instrumentele demonstrative și generatoarele de sisteme expert primite de la Iași, iar domnului prof. Kopenetz pentru documentațiile și însemnările pe care mi le-a pus la dispoziție.

Țin să mulțumesc în mod deosebit tatălui meu, atât pentru cunoștințele împărtășite de-a lungul anilor cât și pentru ajutorul moral și material primite pentru a finaliza și a multiplica această lucrare.

De asemenea doresc să mulțumesc următoarelor persoane:

Domnilor prof. Alexa Pavel, prof. Ioan Olariu, prof. Mircea Petrina, șef lucrări ing. Cristina Câmpcean (U.T. din Cluj-Napoca) și prof. Petru Petrina (Cornell University, SUA), pentru materialele documentare valoroase primite (sub formă de cărți, articole, xerocopii etc), fără de care cercetările mele și pregătirea mea ar fi fost mai sărace.

Domnului șef luc.ing. Dan Turda (U.T. din Cluj-Napoca), pentru contribuția deosebită în conceperea și dezvoltarea sistemului AIC Info și a realizărilor conexe prezentate în această lucrare, pentru talentul său comercial datorită căruia dezvoltarea sistemului a putut fi finanțat, pentru discuțiile constructive și pentru ajutorul neprețuit acordat în cercetările realizate și în planul pregătirii informatice.

Domnului ing. Florin Stanciu, de la Ingenieurbüro Deichmann Stanciu & Partner (Immenhausen, Germania) pentru sprijinul moral și material deosebit și pentru echipamentele informatice primite.

Domnilor Ioan Homorodean, George Comșulea și Augustin Man de la Centrul de Informatică Aplicată pentru ajutorul oferit la implementarea și perfecționarea sistemului AIC Info.

Domnului ing. Traian Onofrei de la LogicTelecom S.A., filiala Cluj, pentru sprijinul susținut și pentru suportul de comunicații oferit sistemului AIC Info.

Domnului mat. Constantin Vasile și colectivului condus de dânsul, de la Centrul de Informatică Aplicată (fostul C.T.C.E.) din Cluj-Napoca, pentru îndrumare și pentru discuțiile avute în faza inițială a acestei cercetări.

Domnului dr.ing. Tiberiu Cocheci de la firma Consort Construcții S.R.L. din Cluj-Napoca, pentru șansa oferită de a lucra în cadrul unui atelier de proiectare dotat deosebit, beneficiind de colaborarea unei echipe de specialiști formați în cadrul acestei firme. Dintre membrii acestei echipe, care au contribuit la dezvoltarea instrumentelor de proiectare prezentate în lucrare, doresc să mulțumesc în special domnului ing. Téglás Attila.

Mulțumesc de asemenea tuturor prietenilor care, direct sau indirect m-au ajutat pe parcursul acestor ani, și nu în ultimul rând, sponsorilor care au făcut posibilă multiplicarea și prezentarea acestei lucrări:

- 3M S.R.L. prin bunăvoința domnilor ing. Marius Latiș și ing. Mircea Vodă,
- TRANSILVANIA S.A. prin bunăvoința domnului prof.ing. Ioan Moga,
- MEDMA S.R.L. prin bunăvoința domnilor ing. Bogdan Mert și ing. Vlad Anastasiu,
- EXAROM Consulting S.R.L. din Cluj-Napoca,
- CONSORT Construcții S.R.L. din Cluj-Napoca.

În final, mulțumirile mele sunt adresate soției și familiei mele, care m-au susținut moral pe întreaga perioadă, dorința lor de a vedea lucrarea finalizată m-a impulsionat în perioadele când lipsa resurselor materiale sau lipsa timpului făceau ca finalizarea ei să pară aproape imposibilă.

CURRICULUM VITÆ

DATE PERSONALE:

Numele: GOBESZ
Prenumele: FERDINAND-ZSONGOR
Cetățenia: română
Data și locul nașterii: 8 August 1964, Cluj-Napoca
Domiciliul stabil: str.E.Racoviță, nr.16, ap.9, 3400 Cluj-Napoca, tel. 064-135179
Starea civilă: căsătorit (soția: ANA GOBESZ, ing.), 1 copil (TAMÁS, 11 ani);
Limbi cunoscute: maghiară (l.maternă), română, engleză, germană, franceză.

STUDII, PREGĂTIRE PROFESIONALĂ:

1978–1982: Liceul de Matematică-Fizică nr.3 din Cluj-Napoca; Bacalaureat în 1982;
1982–1983: Stagiul militar cu termen redus;
1983–1988: Institutul Politehnic din Cluj-Napoca, Facultatea de Construcții, secția C.C.I.A. ingineri zi (5 ani); Examen de diplomă în 1988;
1989–1990: Cursul de formare F1A: "Analiza și proiectarea sistemelor și produselor informatice de gestiune FELIX-MINI" (CEPECA-MEI-CTCE Cluj-Napoca); Examen de diplomă în 1990;
1990–1999: Doctorand la Universitatea Tehnică din Cluj-Napoca.

LOCURI DE MUNCĂ:

1988–1989: Antrepriza de Construcții Industriale Oradea, inginer constructor stagiar, șef punct de lucru în cadrul Brigăzii 1 pe șantierele: C.E.T.II Oradea (construcții industriale), U.H.E. Fughiu și U.H.E. Săcădat (lucrări și construcții hidrotehnice);
1989–1991: Trustul Antrepriză-Generală de Construcții Industriale din Oradea, Oficiul de Calcul, analist-programator;
1991– ...: Universitatea Tehnică din Cluj-Napoca, Facultatea de construcții, Catedra de mecanica construcțiilor, din 02.1991 preparator, din 10.1991 asistent, iar din 1995 șef de lucrări (statică, programare, utilizarea calculatoarelor, P.A.C.);
1992–1994: colaborator la ANSON S.R.L. (DTP, grafică pe calculator etc);
1993–1994: detașat la S.C. Construcții COMECO S.R.L., inginer constructor în R.F.Germană;
1995–1997: colaborator la CONSORT Construcții S.R.L. (P.A.C., dezvoltare software, prezentări multimedia etc);
1997- ...: colaborator la EXAROM Consulting S.R.L. (P.A.C., dezvoltare software, prezentări multimedia, etc).

ARTICOLE ȘI LUCRĂRI PUBLICATE:

- 6 manuale de utilizare;
- 7 rapoarte tehnice;
- 12 lucrări științifice (din care 2 în străinătate);
- 3 referate de doctorat.

Cluj-Napoca, 01.12.1999

CUPRINS

CAP.1 INTRODUCERE	1
CAP.2 CONCEPTELE DE SISTEM EXPERT ȘI BAZĂ DE CUNOȘȚINȚE	
2.1 CERCETAREA INTELIGENȚEI ARTIFICIALE	7
2.2 SCURT ISTORIC AL INTELIGENȚEI ARTIFICIALE ȘI TEHNICII DE CALCUL	8
2.3 PRIVIRE GENERALĂ ASUPRA SISTEMELOR EXPERT	
2.3.1 Roluri în cadrul sistemelor expert	14
2.3.2 Ce este baza de cunoștințe?	15
2.3.3 Mașina de inferență	17
2.3.4 Utilitatea sistemelor expert	18
2.3.5 Limitările sistemelor expert	19
2.4 PRIVIRE GENERALĂ ASUPRA BAZELOR DE CUNOȘȚINȚE	
2.4.1 Reprezentarea cunoștințelor	21
2.4.2 Reprezentarea regulilor	24
2.4.3 Cunoștințele incerte și aproximative	26
2.4.4 Limbajele de programare și reprezentarea cunoștințelor	27
2.5 CÂTEVA ASPECTE LEGATE DE GENERAREA SISTEMELOR EXPERT	
2.5.1 Mecanisme de inferență	29
2.5.2 Medii și generatoare pentru sisteme expert	33
2.5.3 Motorul de elaborare și învățarea	36
CAP.3 SISTEMELE EXPERT ÎN CONSTRUCȚII	
3.1 STRUCTURA ȘI DIRECȚIILE DE DEZVOLTARE ALE SISTEMELOR EXPERT	39
3.2 SISTEME EXPERT, MEDII DE DEZVOLTARE PE MICROCALCULATOR	42
3.3 SISTEMELE EXPERT ÎN DOMENIUL CONSTRUCȚIILOR	
3.3.1 Introducere	50
3.3.2 Sisteme pentru resurse umane și perspective profesionale	51
3.3.3 Reprezentarea cunoștințelor și a raționamentelor în construcții	53
3.3.4 Sisteme pentru planificare și organizare în construcții	55
3.3.5 Sisteme pentru management în construcții	60
3.3.6 Sisteme pentru proiectarea construcțiilor	69
3.3.7 Sisteme pentru analiza structurală	70
3.3.8 Sisteme destinate proiectării structurale	77
3.3.9 Sisteme în domeniul ingineriei seismice	84
3.3.10 Sisteme pentru ingineria geotehnică	88
3.3.11 Sisteme pentru tehnologia lucrărilor și robotică în construcții	90
3.3.12 Educația în construcții și sistemele expert	90
CAP.4 SISTEME PROPRII DEDICATE CONSTRUCȚIILOR	
4.1 PROTOTIPURI ȘI APLICAȚII	
4.1.1 Sistemul expert BPC	95
4.1.2 Instrumente pentru proiectare asistată de calculator	99
4.2 DEZVOLTAREA UNUI SISTEM BAZAT PE CUNOȘȚINȚE, DEDICAT CONSTRUCȚIILOR	
4.2.1 Conceperea și realizarea sistemului AIC Info, structurarea cunoștințelor	104
4.2.2 Componentele dezvoltate și funcțiunile realizate	111
4.2.3 Aplicații decizionale implementate prin AIC Info	118
4.2.4 Realizări conexe (prin revalorificarea parțială a modelării implementate)	120

CAP.5 CONCLUZII

5.1	DOCUMENTAREA ȘI SINTEZA REALIZATĂ, SISTEMUL EXPERT BPC	125
5.2	CONTRIBUȚII LA PERFECTIONAREA PROIECTĂRII, INSTRUMENTE "INTELIGENTE"	126
5.3	BAZA DE CUNOȘTINȚE AIC INFO ȘI REALIZĂRI CONEXE	127

BIBLIOGRAFIE

133

ANEXE

ANEXA 1 SISTEMUL EXPERT BPC		
A.	Sistemul expert BPC generat sub <i>Expert System Toolkit</i>	141
B.	Sistemul expert BPC scris în Fortran/2 IBM	160
ANEXA 2 INSTRUMENTE CONCEPUTE ÎN CADRUL FIRMEI CONSORT CONSTRUCȚII S.R.L.		
A.	Capturi de ecran	183
B.	Fișiere sursă	187
ANEXA 3 SISTEMUL AIC INFO		
A.	Exemple de fișiere sursă	209
B.	Privire asupra bazei de cunoștințe	218
C.	Completare la aplicațiile decizionale implementate prin AIC Info	228
ANEXA 4 REFERINȚE SUPLIMENTARE		
A.	Resurse actuale, accesibile prin Internet	233
B.	Exemple de sisteme expert comerciale din domeniul construcțiilor	234

CAP.1 INTRODUCERE

Oamenii au fost fascinați probabil dintotdeauna de capacitatea de lucru a mașinilor. Este evident că din această cauză există o dorință profundă pentru crearea acestora. Dacă privim înapoi în istoria tehnicii, de la mecanisme vechi (cum ar fi scripetele, cântarul etc.) până la sistemele cele mai noi (cum ar fi digitizoarele, echipamentele de comunicații etc.) întotdeauna au existat încercări pentru copierea și reproducerea unor soluții naturale prin mijloace artificiale. Un farmec aparte caracterizează tentativele de imitare ale unor capacități umane sau realizarea unor instrumente pentru extinderea acestor capacități (de exemplu celebrul și misteriosul jucător automat de șah, realizat de Kempelen). Versiunile moderne ale acestora se referă la roboți, androizi și alte caracteristici științifico-fantastice (de la Frankenstein la Star Trek). În planul științelor teoretice, filozofii din Grecia antică au propus deja crearea unor mecanisme formale bazate pe logică. Câteva variante, realizate în cursul timpului, au la bază construcția unor modele de raționare mecanizată. Un exemplu în acest sens ar putea fi mașina construită de Ramon Lull din Spania, pentru demonstrarea existenței lui Dumnezeu. Lull a folosit caractere ca simboluri pentru reprezentarea cuvintelor (și a argumentelor), precum și combinații ale acestora pe baza unui sistem de reguli. După cum reiese din descrieri, la baza acestui sistem a stat o schemă mecanică prin care era posibilă realizarea unei varietăți de figuri geometrice care, dacă erau mișcate unele față de altele, determinau noi argumente logice. Sistemul conceput de Lull a fost limitat de stadiul de dezvoltare al geometriei tridimensionale prin numărul operațiilor geometrice posibile. Asemenea invenții rămâneau de multe ori secrete sau ajungeau să fie privite ca niște jucării interesante.

O mașină ce poate manipula simboluri este și calculatorul. Principala virtute a acestuia este viteza de operare (numărul ridicat de operații realizate într-un interval scurt de timp). Utilizarea eficientă a acestor echipamente este posibilă prin programarea lor, transpunând astfel gândirea omului în aceste mașini. Folosirea sistemelor de calcul s-a extins astfel de la aplicațiile contabile, financiar-bancare, până la aplicațiile ingineresti, de la recunoașterea și sinteza sunetelor până la modelarea virtuală.

Electronica și informatica, tehnicile de calcul și automatizările, sistemele de comunicații sunt domenii care fac parte din viața noastră cotidiană. Mai mult chiar, dezvoltarea acestor domenii a devenit o necesitate pentru modul în care se măsoară azi progresul societății umane. Din această cauză problematicile complexe ale acestor domenii se regăsesc și în agendele politice ale guvernelor. În ultimul timp am putut constata atât dezvoltarea extraordinară de rapidă a acestor domenii ale științei cât și a tehnicilor folosite în cadrul acestora. Ele s-au împletit din ce în ce mai mult, amplificându-se astăzi într-un proces care este considerat inovativ epocal. Pentru ca societatea să poată valorifica rezultatele acestei noi revoluții tehnico-științifice, ca o țară să se poată pregăti printr-un program eficient la primirea provocărilor ce derivă din acest proces, trebuie cunoscute în profunzime direcțiile componente, precum și formele de apariție, și toate acestea trebuie evaluate realist (în acest proces un rol foarte important revenind învățământului tehnic

superior). Una din crestele acestor dezvoltări (parțial maturizate sau doar conturate promițător) este indubitabil ceea ce se cheamă de către literatura de specialitate "inteligență artificială".

În prezenta lucrare îmi propun să descriu cercetările referitoare la o subramură a acestei specialități, cu implicații în construcții, prezentând și preocupările și realizările mele din acest domeniu. Teza este structurată pe cinci capitole principale. În acest prim capitol voi încerca să prezint fundamentele și noțiunile abordate, sub un aspect mai general. În capitolul următor, după o trecere în revistă a aspectelor referitoare la anumite direcții de cercetare și de dezvoltare a sistemelor pe bază de cunoștințe, voi încerca delimitarea rolurilor ce revin dezvoltatorilor și utilizatorilor. Voi descrie, de asemenea, principalele componente și caracteristici ale unor asemenea sisteme, amintind și avantajele și limitările acestora. În capitolul al treilea, după o sinteză a mediilor de generare consacrate și a sistemelor expert mai cunoscute, voi trece la prezentarea aplicațiilor în domeniul construcțiilor. În acest scop voi aminti și câteva dintre principalele realizări din anii '80 (considerați ca "epoca de aur" a acestor sisteme), având în vedere că în acea perioadă s-au publicat cele mai consacrate modele și prototipuri, multe din sistemele actuale bazându-se pe acestea. Cercetările și realizările mele urmează să fie prezentate în capitolul patru, într-o ordine cronologică. După concluziile personale din cel de-al cincilea capitol urmează încă patru anexe pentru exemplificarea și ilustrarea suplimentară a unor aspecte.

Se știe foarte bine că dacă pentru soluționarea unor probleme dintr-un domeniu limitat dispunem de o descriere clară și bine definită a procedurii de rezolvare, atunci această muncă (în unele cazuri cu pronunțat accent repetitiv) poate fi încredințată unui sistem automat de calcul. Cele mai multe programe sunt scrise sub forma unor algoritmi, acestea conținând practic lista instrucțiunilor în ordinea corespunzătoare executării lor de către calculator. De cele mai multe ori un asemenea algoritm utilizează date care sunt organizate într-o anumită formă, în fișiere separate, externe. Orice fișier poate fi definit ca o colecție organizată de date care este tratată uniform (datele pot suferi același tip de prelucrări). Programul de calculator poate fi privit astfel și ca un pachet alcătuit din date + algoritm. De exemplu, un program pentru calculul salariilor poate fi descris în felul următor: caută prima persoană din înregistrări, citește salariul brut al persoanei găsite, determină treapta de impozitare în care se încadrează, calculează impozitul, caută toate celelalte scăzăminte (contribuții la fonduri de asigurări, cotizații etc.) și determină salariul net. Toate aceste elemente ajung sub o formă prestabilită pe fluturașul de salarizare și pe statul de plată. După prelucrarea datelor primului angajat sistemul pentru calcularea salariilor continuă determinarea salariilor pentru ceilalți angajați, indiferent dacă numărul lor este de ordinul zecilor sau al sutelor. Algoritmul nu se schimbă la nici un angajat. Sistemele de acest gen se bazează pe tehnica de calcul timpurie proiectată pentru rezolvarea problemelor numerice repetitive, complet plictisitoare. Aplicațiile mai recente oglindesc deja și schimbările tehnologiei. Răspândirea utilizării calculatoarelor personale a condus la apariția unor aplicații mai mici de salarizare, bazate pe tabele de calcul, care sunt compuse din șiruri și coloane de numere manipulate cu ajutorul sistemului de calcul. Pachetele de programe care au fost vândute în numărul cel mai mare sunt tabelele de calcul și

procesoarele de texte. Oamenii însă nu folosesc numai cuvinte și cifre ci manevrează și o cantitate din ce în ce mai mare de informații. Acest fapt a condus la apariția și dezvoltarea bazelor de date. Bazele de date înmagazinează o cantitate enormă de informații sub formă structurată, utilizatorul având acces la informația dorită oricând, în orice formă dorită. Un exemplu foarte bun pentru ilustrarea celor amintite poate fi baza de date a angajaților. În cadrul acesteia fiecărui angajat îi corespunde un articol, fiecare articol conținând informațiile referitoare la un angajat, de exemplu numele, adresa, data nașterii, salariul de încadrare, locul de muncă etc. Utilizatorul (ex. administratorul societății) poate întocmi, din informațiile conținute de către baza de date (prin folosirea unui limbaj de interogare), o situație pentru analiză despre angajați, ori de câte ori este nevoie. Poate obține rapoarte de genul: lista angajaților cu salariul peste 1.000.000 lei, cu vechime peste 3 ani, domiciliați într-o anumită zonă. Ceea ce putem obține din baza de date este limitat doar de către informația înmagazinată și de către limbajul de interogare.

Omul însă se interesează nu numai de informații ci și de cunoștințe. Informația este mulțimea datelor organizate comprehensibil. Cunoașterea este informația definită din punctul de vedere al aplicației date. Bazele de cunoștințe și sistemele expert ce rulează pe calculatoarele personale au fost create pentru colectarea, înmagazinarea, interogarea și realizarea accesibilității acestei cunoașteri. În această lucrare de doctorat mi-am propus să prezint o parte din problematica acestui domeniu, precum și rezultatele cercetărilor și dezvoltărilor proprii (până la data redactării tezei).

La baza programelor "clasice" de calculator stă deci noțiunea de algoritm, care de cele mai multe ori implică o organizare secvențială, predefinită de operații. Aceste programe însă par să fie incapabile să rezolve probleme în care intervin raționamente, decizii sau în care pot să apară situații ce nu se pot specifica a priori. Din această cauză, destul de mult timp s-a crezut că raționamentul și comportamentul uman nu poate fi modelat prin sisteme informatice. Ideea pare într-adevăr foarte ambițioasă, însă ea reprezintă obiectivul urmărit azi de către cercetătorii din domeniul inteligenței artificiale. Cercetările acestora se desfășoară în diverse direcții, cum ar fi: rețele neuronale, sisteme solvitoare, baze de cunoștințe, sisteme expert, recunoașterea formelor, recunoașterea și reproducerea limbajului natural, robotica etc. Pe lângă acestea nu trebuie însă uitate cercetările fundamentale care privesc sistemele cognitive, reprezentarea și manipularea cunoștințelor.

Ce este un sistem expert? După cum ne sugerează esența însăși denumirea, sistemul expert este un ansamblu computerizat de aplicații care în cadrul unui domeniu de specialitate precis conturat și foarte bine delimitat, respectiv în cadrul unei probleme speciale, oferă servicii speciale de expert (consultanță), rezolvă probleme complexe, într-un mod independent sau în strânsă colaborare cu utilizatorul său. Sistemul expert este deci un sistem specializat deoarece:

- ❶ se limitează la problemele dintr-un domeniu de specialitate relativ restrâns,
- ❷ resursele sale (fizice și logice) sunt de regulă întărite pentru rezolvarea performantă a problemelor din domeniul dat. Aceasta reprezintă una din

caracteristicile ce-l deosebesc de alte sisteme automate destinate prelucrării informațiilor (bazându-se de altfel pe tehnici în mare parte corespunzătoare cu acestea).

Toate acestea se pot afirma însă și despre așa zisele "sisteme dedicate" a căror concepție—ca și una din direcțiile domeniului rezolvărilor de probleme—este cunoscută de mai multă vreme. Din acest punct de vedere noțiunea sistemului expert poate fi înțeleasă ca o renaștere a filozofiei sistemelor dedicate pe o treaptă superioară, datorită îmbogățirii calitative importante a sistemului. Primele exemplare ale sistemelor expert s-au născut atunci când s-a amplificat dezvoltarea unor ramuri depășind limitele clasice ale științei, sau la frontiera ei, și acestea au putut contribui cu rezultate în premieră la formarea acestor sisteme. Din acest punct de vedere am putea afirma, deocamdată ca o definiție vagă, că sistemul expert nu este altceva decât un sistem dedicat plus—mai multă sau mai puțină—inteligență artificială.

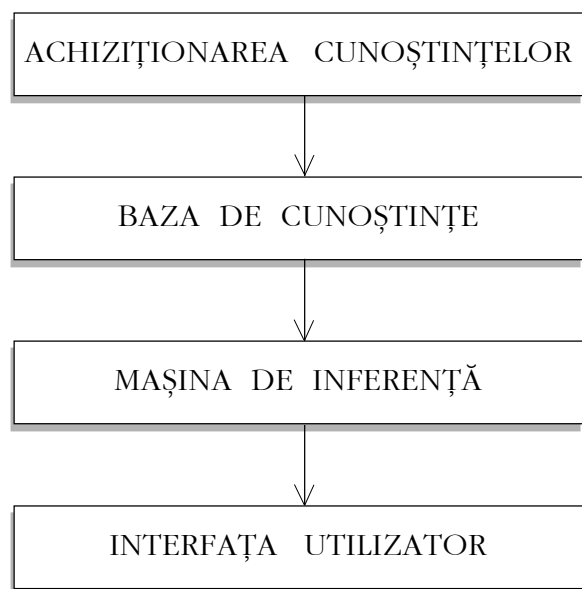


Fig.1 Componentele realizării unui sistem expert printr-un mediu

„Sistemul expert poate fi privit ca materializarea în cadrul sistemului de calcul a bazei de cunoștințe cumulate din experiența expertului, într-o formă în care sistemul este capabil să propună un răspuns inteligent sau să poată realiza o decizie inteligentă într-o situație reală. O altă trăsătură dorită și considerată fundamentală ar fi capacitatea de a explica punctul propriu de vedere sub o formă inteligibilă utilizatorului.” (British Computer Society). O modalitate de a obține aceste caracteristici poate fi realizată prin programarea bazată pe reguli. Urmărind această idee, sistemul expert ar avea patru componente (după cum este ilustrat în figura 1).

În acest caz, folosind un mediu (shell) pentru elaborarea unui sistem expert, primul lucru pe care trebuie să-l facă un utilizator potențial este introducerea cunoștințelor în sistem. Pentru atingerea obiectivului pot fi folosite mai multe procedee, în funcție de complexitatea sistemului expert. Dacă sistemul expert este de tipul celor bazate pe reguli, atunci cunoașterea poate fi introdusă sub forma unor mulțimi de reguli. Dacă sistemul expert este de tipul celor producătoare de reguli ("de producție"), atunci regulile pot fi deduse din datele avute la dispoziție. Colectarea, modelarea respectiv introducerea cunoașterii nu este o operație atât de simplă pe cât ar părea la prima vedere. Sistemul trebuie să dispună de seturi de reguli înlănțuite, cuprinzătoare, pentru acoperirea tuturor posibilităților. Baza de cunoștințe (și regulile la sistemele bazate pe reguli) reprezintă partea cea mai importantă a sistemului. Cunoștințele (și regulile) conținute sunt ierarhizate. La sistemele expert generate cu ajutorul mediilor de dezvoltare existente, cunoștințele, odată introduse și cuprinse în baza de cunoștințe, rămân nemodificate pe parcursul

funcționării sistemului. Adică, utilizatorul poate avea acces la cunoștințe prin chestionarea regulilor, dar regulile nu se modifică. Există situații în care regulile sunt actualizate, completate sau îmbogățite, dar aceste schimbări se realizează separat de interacțiunile obișnuite cu baza de cunoștințe.

Mașina de inferență —după cum sugerează și denumirea— este componenta răspunzătoare pentru modul de funcționare al sistemului. Mașina are acces la reguli, promovând astfel relațiile, legăturile, conexiunile și deducțiile necesare. Există două moduri principale, considerate ca fiind de bază, pentru raționamente: deducția (permite să se obțină concluzii pornind de la premisele regulilor verificate) și inducția (conduce la regulile de verificat, pornind de la combinații ale factorilor, chiar de la scheme predefinite ale acestora). Acestea pot fi aplicate analitic și/sau sintetic asupra domeniilor considerate. De fapt, pentru rezolvarea problemelor complexe, metoda clasică este prin segmentarea acestora. Un alt mod de raționare, apărut ceva mai târziu și destinat în principal eficientizării inducției, este abducția. Acest procedeu constă în realizarea și combinarea unor scheme de observare ipotetice pentru pornirea inducției. Natura și numărul regulilor nu influențează modul de funcționare al mașinii de inferență. Deci, dacă se impune, se pot modifica regulile fără ca aceasta să afecteze mașina de inferență, deși raționamentele (drumul care conduce prin baza de cunoștințe) pot suferi schimbări. Este foarte important ca toate sistemele să poată fi utilizate ușor. Erorile, de orice natură ar fi, trebuie să fie sesizabile și, în caz de nevoie, trebuie să existe un program de ajutorare (help). De regulă sistemele expert dispun de programe ajutătoare integrate, acestea făcând parte din interfața utilizator.

Sistemul expert poate furniza explicații referitoare la raționamentul efectuat de către el. Pe parcursul chestionării sistemului, utilizatorul poate solicita explicații referitoare la motivele drumului deductiv urmărit de către sistem, după cum poate fi solicitat un maestru de șah să-și explice mutările. Valoarea fundamentală a sistemului expert rezidă în capacitatea de a înlocui expertul uman. Ca urmare, sistemele expert sunt realizate, în principiu, pentru a furniza recomandări, sfaturi și, nu neapărat, răspunsuri. Sistemul expert ar putea fi exprimat sumar prin ecuația:

$$\textbf{sistem} = \textbf{cunoaștere} + \textbf{inferență}$$

Principală caracteristică a sistemelor expert este derivată din baza de cunoștințe împreună cu un algoritm de căutare specific metodei de inferență. Deci, un sistem expert poate fi definit ca "un program informatic capabil să reproducă raționamente umane". Ea poate astfel trata cu succes probleme pentru care nu există o soluție algoritmică clară.

Programele tradiționale, obișnuite, după cum am mai amintit, sunt algoritmi. Am menționat mai devreme că acestea funcționează foarte bine în cazul problemelor repetitive.

$$\textbf{program} = \textbf{algoritm} + \textbf{date}$$

Dacă datele reprezintă salariile angajaților, atunci prin aplicarea algoritmului se vor obține salariile brute și nete, respectiv impozitele. Sistemul expert însă funcționează altfel. Drumul care conduce prin baza de cunoștințe este determinat prin reacțiile utilizatorului la întrebările adresate de către sistem. Diferențele dintre sistemele expert și programele tradiționale pot fi rezumate după cum urmează:

Pachetul de programe tradiționale

manevrează date
aplicând algoritmi
parcurgând prelucrări repetitive și
se sprijină pe baze de date

Sistemul expert

manevrează cunoștințe
aplicând metode euristice sau reguli
parcurgând procedee prin inferență și
se sprijină pe baze de cunoștințe

În legătură cu sistemele expert, trebuie să mai precizăm un lucru. Denumirea de „sistem expert” se aplică de multe ori nu numai aplicației realizate ci și pachetului instrumental care ajută la elaborarea acestuia. Există o varietate relativ mare de medii accesibile pentru generarea unor sisteme expert (*"expert system shell"*, *"expert system toolkit"*) care permit dezvoltarea aplicațiilor chiar și pe microcalculatoare.

Tehnologiile de programare utilizate la dezvoltarea sistemelor expert pot fi privite și ca o evoluție a limbajelor calculatoarelor spre un nou nivel al exprimării (permițând utilizarea și reprezentarea cunoștințelor). Limbajele timpurii de programare, incluse de hardware-ul ce executa acestea, sunt dificil de utilizat pentru nespecialiști în calculatoare. Mai recente limbaje de programare (de nivel înalt, obiectuale, vizuale etc.) sunt binevenite pentru utilizarea și includerea diverselor forme de exprimare umană. Sistemele expert fac parte deci dintr-o gamă de instrumente indispensabile pentru realizarea unor sisteme automate sau interactive, capabile să efectueze sarcini complexe. Nu este cu puțință să se ignore toate raționamentele calitative și simbolice care formează baza proceselor decizionale. Aceste raționamente pot fi simulate prin sisteme expert, iar în această lucrare voi încerca să prezint și câteva aspecte specifice acestei probleme. Încă de la început sistemele expert au avut asociate instrumente de cercetare operațională care au facilitat realizarea unor sisteme de asistare foarte complexe (diagnostizare, interfețe inteligente, medii de producție ș.a.). În prezenta lucrare de doctorat încerc să abordez o parte din acest domeniu fascinant, nu numai din cauza curiozității caracteristice cercetătorilor, dar și deoarece consider că lucrarea este realizată într-un moment oportun pentru a se insera în eforturile susținute de modernizare pe de o parte a învățământului, iar pe de altă parte a producției și managementului. În general, acestea din urmă sunt întotdeauna legate în cazul inteligenței artificiale, iar sistemele expert și bazele de cunoștințe formează fundamentul acestora și sunt deja operaționale în mai multe domenii.

CAP.2 CONCEPTELE DE SISTEM EXPERT ȘI BAZĂ DE CUNOȘTIȚE

2.1 CERCETAREA INTELIGENȚEI ARTIFICIALE

Obiectivele cercetării inteligenței artificiale –după A. Sloman [172]– sunt următoarele:

- ❶ Analiza teoretică a explicațiilor posibile și aplicabile în practică, referitoare la înțelegerea comportării raționale.
- ❷ Înțelegerea profundă a capacităților umane.
- ❸ Realizarea unor produse inteligente.

Granițele cercetării inteligenței artificiale sunt greu de trasat. Însăși definiția științifică a noțiunii de inteligență este încă deschisă, nerezolvată. Nu există o asemenea "substanță" care să se numească inteligență. Însă există fără dubii și se pot demonstra experimental diferitele elemente ale noțiunii comune de inteligență: factori cognitivi (Spermann), diverse structuri și unități funcționale (R. Zazzo), mecanisme de funcționare (Piaget). Citându-l pe A. Sloman: "există numeroase <<priceperi>>, calități pentru a vedea diverse lucruri, pentru înțelegerea limbii, pentru planificare și testare, pentru rezolvarea problemelor, pentru conducerea activităților noastre și urmărirea lor..." [172]. Toate aceste calități presupun cunoștințe specifice unor domenii, pe care oamenii le folosesc (de multe ori inconștient) în cadrul unor activități profesionale sau în viața cotidiană. Definirea explicită și depozitarea acestor cunoștințe este una din realizările cercetărilor în domeniul inteligenței artificiale care au condus la apariția bazelor de cunoștințe și a sistemelor expert. Într-o bază de cunoștințe se găsesc, pe lângă faptele, noțiunile, relațiile din literatura (colecția) de specialitate a domeniului și cunoștințe de tip *know-how*, experiența, dar și manifestările proceselor intuitive de gândire ale specialistului care-și exercită meseria la un nivel calitativ înalt. De fapt, din punctul de vedere al posibilităților de înmagazinare în sistem, pot fi considerate orice capacități ale spiritului uman, oricare componentă a inteligenței naturale, cum ar fi de exemplu:

- capacitatea surprinderii senzațiilor subiective, sesizarea (auz artificial, vedere artificială),
- capacitatea de a gândi la ceva, atenția,
- capacitatea sesizării faptelor necăutate de la început, observația,
- capacitatea de a învăța, autoperfecționarea,
- capacitatea de cumpănire, decizie, planificare, acțiune, inclusiv capacitatea de comparare a obiectivelor și a mijloacelor,
- capacitatea de a realiza și manevra simboluri, modele, reprezentări,
- capacitatea de gândire propriu-zisă, raționarea (capacitatea ca folosind cunoștințele posedate să ne lărgim cunoașterea cu cunoștințe noi, concluzii).

Abilitatea de a învăța, conform specialiștilor, aparține și ea în majoritate conceptului de sistem expert, reprezentând astfel o a doua calitate importantă prin care cercetarea

inteligenței artificiale a contribuit la realizarea acestor sisteme. Printre sistemele expert existente astăzi, sunt unele care dispun de o abilitate pasivă de a învăța. Acestea pot acumula cunoștințe, reguli noi, prin reprogramare sau restructurare de multe ori profundă. În realizarea abilității de învățare, deocamdată se cunosc doar rezultate incipiente. Din acestea se poate însă observa deja că, pentru învățare (folosind experiența), sistemele au nevoie de o cantitate importantă de antecunoștințe specifice domeniului și nu numai.

O componentă importantă a sistemelor expert o reprezintă sistemele de raționare (mecanismele de inferență). Acestea sunt dezvoltate și folosite în principal pentru verificarea relațiilor logice. Există aplicații în care acestea sunt concepute pentru demonstrații matematice și logice, validări (de exemplu la configurarea resurselor hardware/software sau la construcția programelor) etc, fără a fi însă sisteme expert propriu-zise. Din ce în ce mai multe firme oferă suport "on-line" consumatorilor prin intermediul unor baze de cunoștințe (actualizate oportun) combinate cu mecanisme de inferență și interfețe prietenoase, evitând astfel factorii aleatori ce derivă din natura umană a unor angajați pe asemenea posturi.

De dezvoltarea sistemelor expert se leagă și alte domenii din cadrul inteligenței artificiale. Printre acestea trebuie să menționez în primul rând prelucrarea limbajelor naturale, precum și recunoașterea și sinteza vorbirii. Pornirea pentru acestea s-a născut din dorința utilizatorilor de a comunica prin viu grai cu mașinile. Deși există câteva rezultate remarcabile în acest domeniu, ele nu sunt încă suficient de răspândite. Astfel, în majoritatea sistemelor cu inteligență artificială, pentru procesarea textului, dialogul se derulează cu ajutorul monitorului și a tastaturii. Printre perspectivele (nu foarte îndepărtate) ale acestor cercetări putem menționa aplicarea rezultatelor în sistemele de dispecerat, în domeniile traducerilor, a pregătirii (învățării) asistate de calculator etc. Asemenea facilități vor conduce cu siguranță la creșterea eficienței sistemelor expert. Un alt domeniu important este cel al înțelegerii și prelucrării imaginilor. Obiectivul acestor cercetări este ca sistemul să posede capacitatea de recunoaștere și analizare a informațiilor vizuale. Din aceste informații sistemul ar putea trage concluzii, valorificându-le prin servicii de informare, comunicare sau comenzi (de exemplu roboți care "văd").

2.2 SCURT ISTORIC AL INTELIGENȚEI ARTIFICIALE ȘI TEHNICII DE CALCUL

Între calculatoare și oameni există diferențe cantitative și calitative. Calculatorul poate executa mai multe milioane de operații într-o secundă, mult mai multe decât utilizatorul uman. Performanța prelucrării, adică mai bine zis a capacității care face posibilă realizarea calculelor de acest gen, evident nu are limite. Cantitatea calculelor nu reflectă calitatea cunoașterii deținute de calculator. Între cunoașterea umană și cea a calculatoarelor există încă diferențe enorme. Acest fapt poate să surprindă cunoscând performanțele date ale calculatoarelor. Să ne imaginăm pentru un moment câteva activități umane obișnuite, care nu pot fi executate cu calculatoare: de exemplu curățirea unui ou fiert moale, recunoașterea unei șosete într-un sertar conținând șosete neîmperecheate, condimentarea alimentelor etc. Fiecare activitate enumerată presupune combinația următoarelor trei lucruri: trebuie

achiziționate informațiile din lumea ambientală, trebuie stabilite sarcinile de efectuat pe baza unei cunoașteri compuse înmagazinate și, în fine, trebuie tratată lumea exterioară prin mișcările acordate ale mâinilor și ale ochilor. Nu este exclus ca aceste activități să fie realizate cândva de către calculatoare (există deja roboți care dispun de control tactil și există roboți care pot urmări cu "vederea" (stereoscopică) obiecte în spațiul tridimensional, putând astfel evalua și distanțele până la acestea). Au fost deja și tentative experimentale de copiere a modalităților umane în rezolvarea problemelor. În anul 1950 a fost elaborat modelul general pentru rezolvarea problemelor (GPS). Scopul era realizarea unui model care să raționeze inteligent, rezolvând toate problemele puse. Mai târziu, un asemenea GPS ar fi dispus și de capacități intuitive pentru a putea învăța din succesele și eșecurile proprii. Din păcate această abordare a fost sortită eșecului, din cauza multitudinii exagerate a dorințelor.

Dezvoltarea sistemelor expert a început pe la sfârșitul anilor '50. Primele sisteme erau necoapte și n-au părăsit incintele laboratoarelor. Mulți oameni și multe instituții au renunțat la sistemele expert și la inteligența artificială, considerându-le instrumente neaplicabile în practică. Străpungerea s-a realizat după ce oamenii de știință au început să privească fiecare problemă ca temă separată, rezolvându-le individual. În anii 1960 și 1970 s-au dezvoltat acele sisteme cu baze de cunoștințe care erau aplicate pe câte un domeniu distinct. Printre aceste sisteme "mari" trebuie să amintim: **DENDRAL** (pentru identificarea structurii moleculare), **MYCIN** (pentru diagnosticarea îmbolnăvirilor contagioase și pentru elaborarea unor planuri de cură și tratament), respectiv pachetul **R1** (conceput de către firma „Digital Equipment” pentru stabilirea de către utilizatori a configurației proprii de calcul). Tematica inteligenței artificiale se compunea (deja în mod "tradițional") din mai multe domenii mai mici de cercetare, dezvoltare, aplicare, care se puteau controla mai bine. La sfârșitul anilor '70, urmând succesul înregistrat mai ales de către sistemul expert medical MYCIN, s-au relansat cercetările în domeniul acestor sisteme. În ciuda ecurilor favorabile în cadrul cercurilor științifice, aceste sisteme "mari" nu au reușit să pătrundă în viața comercială. Realizarea sistemelor expert era o treabă încă mult prea complicată, nu se dispunea de unelte suficient de corespunzătoare și confirmate. Elaborarea acestora necesita efortul susținut și de durată a unui colectiv destul de mare de specialiști, cu eforturi financiare considerabile și cu rezultate de multe ori contestate. Această situație s-a schimbat doar în anii 1980, prin apariția generatoarelor dedicate sistemelor expert. Un asemenea generator de sisteme expert este și KES (Knowledge Engineering System) din care s-a realizat MICRO-PS. Componenta „MICRO” din cadrul denumirii acestui sistem dorește să sugereze faptul că pachetul este destinat microcalculatoarelor, iar „PS” este prescurtarea de la *production system* (sistem de producție). Așa sunt denumite în literatura de specialitate sistemele expert care sunt clădite din reguli *if-then* (dacă-atunci). Se poate pune întrebarea: de ce aceste sisteme sunt denumite sisteme de producție și nu sisteme de reguli (*rule-system*)? Majoritatea literaturii de specialitate referitoare la sistemele expert folosește această denumire, care își găsește originea în lucrarea lui E. Post cu caracter de pionierat, scrisă în anul 1943, care caracteriza producția ca un proces general, computerizat. Unul din avantajele generatorului de sistem expert este acela că permite un mod de construire asemănător cu limbajul natural (cel englez). În MICRO-PS, de

exemplu, o regulă poate fi scrisă sub următoare formă:

```
if prompt = a,  
& drive = b,  
then step = type b: enter.
```

(tradus: dacă indicatorul = a,
și unitatea = b,
atunci pasul = tastează b: enter.)

Această regulă exprimată în limbajul LISP ar fi mai lungă și mai complicată, scrierea ei ascunzând multe posibilități de a greși (chiar și numai datorită nivelurilor de paranteze):

```
($and (eq prompt a)  
(eq drive b))  
(conclude (step type b: enter))
```

Regulile apropiate de limbajul natural permit autorului sistemului expert (*knowledge base author, KBA*) să-și îndrepte mai mult atenția asupra problemei de rezolvat și să nu se preocupe de problemele limbajului complex de programare care, și ea, ar necesita o atenție mare. Autorul nu trebuie să dispună astfel de experiență vastă în domeniul programării. Acest fapt este un avantaj deosebit, deoarece generatoarele de genul lui MICRO-PS nu necesită ca autorul sistemului să verifice de fiecare dată programul. Mecanismul de comandă inclus în generatorul de sistem scutește scriitorul programului de redactarea unei părți foarte mari a codului programului. Acest mecanism, denumit mașină sau motor de inferență (*inference engine*), folosește regulile scrise de către specialistul din domeniul cunoașterii pentru a stabili ce anume trebuie să-l întrebe pe utilizator. Specialistul din domeniul cunoașterii nu trebuie să se preocupe nici de modul în care vor apare întrebările adresate utilizatorului pe ecranul calculatorului, lucrul acesta fiind realizat tot de către generatorul de sistem, de regulă prin interfețe grafice prietenoase (*GUD*). Gestionând și această parte a intrărilor și ieșirilor, generatorul de sistem oferă practic un mediu de lucru (*shell*) atât pentru crearea, dezvoltarea, cât și pentru consultarea și exploatarea sistemului expert.

După succesul sistemelor cu baze de cunoștințe mari, au fost dezvoltate sisteme expert mai mici. Există păreri care consideră că prin crearea acestor sisteme mici se realizează o îndepărtare de inteligența artificială și se obțin niște sisteme care nu corespund de loc acestei noțiuni. Din această cauză ar fi nevoie probabil de o definire a noțiunii de inteligență. Inteligența este capacitatea de a rezolva probleme, creativitatea, comunicarea cu alții sau (printr-o exprimare mai tradițională) propria convingere că acțiunea conștientă ne face inteligenți și ne deosebește de lumea animalelor? Extinzând întrebarea, când putem afirma despre o mașină că acționează inteligent? Dacă dorim, putem utiliza *Testul Turing*, propus în anul 1954 de către matematicianul englez Alan Turing (înainte de moartea lui timpurie survenită la 41 de ani), un pionier al domeniului inteligenței artificiale și al tehnicii de calcul. El afirma că o mașină atunci se comportă inteligent când reușește să lase asupra utilizatorului (care comunică cu ea printr-un dispozitiv periferic) impresia că partenerul lui nu este o mașină. Într-o partidă de șah la distanță este greu să-ți dai seama dacă partenerul este om sau mașină. Oare poate fi considerat sistemul *Deep Blue*, care l-a învins

pe maestrul de șah Kasparov, ca fiind inteligent? În acest context, consider că partenerul ar trebui să convingă –despre faptul de a fi om– mai degrabă prin capacitatea de a pierde (bineînțeles fără mutări stupide) în jocurile de tipul celor jucate cu copii în vederea creșterii încrederii lor. Bănuiesc că este mai ușor de reprodus facultatea câștigării convingătoare decât știința pierderii convingătoare.

Sistemele actuale de inteligență artificială conțin cunoștințe ce sunt înmagazinate de calculator și pot fi accesate atât de către calculator cât și de către utilizator. Din păcate nu există o idee consistentă referitoare la modul de reprezentare a cunoașterii. Nu avem o teorie atotcuprinzătoare despre o asemenea cunoaștere, cum ar fi cea bazată pe ideile, gândurile unui om mare, ca de exemplu relativitatea pentru Einstein, gravitația pentru Newton sau gazele pentru Boyle. Putem accepta părerea că o mașină poate prosti oricând pe oricine, dar cum stochează ea cunoașterea care cauzează acest fapt? Există două curente de opinii care stau față în față. În primul rând există eleganții, care consideră că inteligența artificială are principii fundamentale structurate, generale, sistematizabile, formale, testabile, ce utilizează logica deductivă. Purtătorii de cuvânt ai acestora sprijină limbajul de nivel înalt denumit PROLOG, care se clădește prin declarații, stochează legăturile (relațiile) dintre obiecte și trage concluzii din ele. Față în față cu eleganții se găsesc cei îmbrăcați lejer, după părerea cărora inteligența umană funcționează după reguli ad-hoc sau euristice, mult mai lejere, are puține reguli fundamentale structurate (sau acestea sunt chiar inexistente). Abordarea lor se materializează în LISP, care este un limbaj destinat în special pentru gestionarea listelor și tratarea simbolurilor. LISP domină mai ales în S.U.A., pe când PROLOG este agreat în primul rând în Europa. Întrucât cunoștințele noastre referitoare la om, la sistemul lui de înmagazinare a cunoștințelor (memoria) și la inteligența lui nu sunt complete, probabil nu vom putea soluționa disputa dintre cei eleganți și cei îmbrăcați lejer. Între timp, în domeniul inteligenței artificiale oamenii lucrează, în majoritatea cazurilor la reguli, eventual clădesc sisteme expert fundamentate pe euristică urmând abordarea propusă de cei îmbrăcați lejer, deși –pentru a complica oarecum problema– unele dintre aceste sisteme sunt scrise în PROLOG. Proiectul inițiat de japonezi pentru construirea calculatoarelor de generația a V-a merită toată admirația, dar nu putem desconsidera nici eforturile imense anterioare, care în alte țări au dus la formarea producțiilor și piețelor existente, acestea neputând fi ignorate pentru a porni de la capăt cu alte principii. Răspunsul actual este o anumită combinație a structurii și a lejerității. Putem deveni eleganți în modelarea pe calculator în măsura în care putem să ne înțelegem egoul, convingându-ne asupra structurii și asupra regulilor fundamentale ale creierului, ale gândirii și ale inteligenței proprii.

În răstimpul în care cercetătorii au pus accentul pe gândire și inteligență, s-au cercetat și alte proiecții ale inteligenței artificiale. A fost îndreptată o cantitate importantă de atenție asupra dezvoltării roboților. A renăscut visul original de a realiza suplinitorul electronic, mecanic complet al omului. Această idee a fost neglijată temporar în interesul roboților mici. Acești roboți mai mici sudează componentele caroseriilor la automobile, fierb ceai sau cafea, ori dezafectează bombe. În fine, este posibil ca toate iscusințele manuale să apară și în formă umanoidă. Până atunci, însă, roboții își continuă munca precisă, repetitivă

(fără pauze de cafea, concedii sau greve).

Mai devreme am amintit generația a cincea a calculatoarelor. Ce sunt celelalte patru generații? Merită să aruncăm o privire asupra tehnologiei pentru că aceasta ne conduce la aplicații mai performante, bazate pe tehnologii mai performante; în aceeași timp aflăm desigur și limitele unor tehnologii. Visele dezvoltatorilor de soft sunt adesea spulberate de tehnologia care se târăște în urma așteptărilor acestora. În cazurile normale amândouă depășesc dorințele și necesitățile recunoscute ale utilizatorilor (tehnicieni, ingineri, manageri). Prin conectarea tuburilor electronice și a altor piese și subansamble s-au născut primele calculatoare la începutul acestui secol. Calculatoarele din prima generație erau foarte voluminoase, scumpe, predispuse la supraîncălzire și nefiabile. Din cauza dimensiunilor mari, a prețului și a utilizării restrânse, erau destul de puține asemenea aparate, cele existente aflându-se în instituții de învățământ superior sau în sectoarele de cercetare ale marilor corporații. Erau folosite pentru calcule aritmetice cu posibilități reduse de prelucrare și cu memorie accesibilă puțină. La începutul anilor 1950 dezvoltarea tranzistoarelor și a diodelor (a semiconductoarelor în general) a condus la apariția calculatoarelor din generația a doua. La început acestea aveau capacități de prelucrare reduse, dar cu ajutorul benzilor magnetice aveau acces la cantități imense de memorie, devenind astfel ideale pentru probleme repetitive (de exemplu calcularea salariilor, care presupune calcule simple dar pentru un număr de angajați care poate fi foarte mare). Majoritatea organizațiilor mari au cumpărat calculatoare și le-au folosit la prelucrarea evidențelor financiar-contabile (bonuri, salarii etc.). Aceste calculatoare erau scumpe și voluminoase, necesitând condiții ambientale bine controlate, din această cauză fiind implantate în centre de calcul. Răspunderea revenea administratorilor de sistem și colegilor acestora, de la ei așteptând și dirijarea. Calculatorul era inaccesibil publicului larg.

Spre sfârșitul anilor 1950 John F. Kennedy a declarat că americanii, în decurs de 10 ani, vor trimite om pe lună. Pentru atingerea scopului NASA trebuia să dezvolte nu numai tehnologia rachetelor dar și sistemele de comunicare, navigare. Filmele timpurii științifico-fantastice puteau prevesti rachetele, costumele spațiale, dar atenția lor a fost ocolită de cantitatea excepțională a tehnologiei care înconjoară astronautul în cabina sa. Programul spațial a dat elan dezvoltării calculatoarelor din a treia generație, bazate pe circuite înalt integrate (*LSD*). Aceste circuite integrate, cunoscute sub numele generic *chip* (sau *microchip*), realizate din plăci de siliciu cu tehnologii de vârf, pot conține elemente echivalente cu mii de tranzistoare. Dezvoltatorii și-au dat repede seama că aceste microprocesoare pot fi folosite în afara programului spațial, pentru realizarea unor calculatoare eficiente și ieftine. S-a ajuns astfel la calculatoarele din generația a treia, inimaginabil mai ieftine, mai mici și mai performante decât calculatoarele realizate cu douăzeci de ani înainte. De performanțele ridicate ale acestora au beneficiat aplicațiile timpurii de inteligență artificială, deși aceste sisteme erau încă exploatate și verificate central. În afara celor destinate cercetării din centrele universitare, doar în mâinile specialiștilor în prelucrarea datelor se aflau calculatoare. Între timp contabilii și câțiva oameni de afaceri au început folosirea calculatoarelor de masă. Pașii făcuți către circuitele VLSI (*Very Large Scale Integrated*) au asigurat capacitatea de prelucrare necesară construirii

calculatoarelor personale, care reprezintă o parte din calculatoarele din a patra generație. Aceste aparate sunt mici, eficiente și ieftine. Deoarece nu necesită condiții ambientale speciale, ele pot fi amplasate pe un birou, în locuințe sau la puncte mai îndepărtate, la diverse puncte de lucru, de exemplu în magazine, prăvălii, la ieșirea din supermarket (în ghișeul de casă), în hale de producție, pe șantiere etc. Paralel cu dezvoltarea aparatelor mici a crescut numărul programelor și aplicațiilor de utilizare ce se pot rula pe asemenea calculatoare. Printre acestea se găsesc editoare de texte, tabele de calcul, gestionare de baze de date, programe grafice și de publicistică și, bineînțeles, sistemele expert cu dimensiuni reduse. Aceste calculatoare nu mai erau cumpărate, programate și controlate de către administratorii de sisteme. Ele se aflau deja, în sensul adevărat, în mâinile utilizatorilor.

Calculatoarele timpurii nu erau prea performante, posibilitățile lor de aplicabilitate erau destul de restrânse, dar s-au întâmplat două lucruri. În primul rând calculatoarele personale au devenit din ce în ce mai eficiente, au devenit adecvate pentru rularea unor limbaje de programare cu necesități mari de memorie (cum ar fi PROLOG). Utilizarea discurilor cu suprafețe magnetice pe post de memorie a realizat accesul aparent instantaneu la cantități foarte mari de date. Calculatoarele de azi sunt adecvate pentru rularea unor pachete destul de mari de programe (de exemplu pentru exploatarea unei baze de cunoștințe sau pentru rularea unui sistem expert care poate juca un rol semnificativ în cadrul unei organizații). Pe de altă parte, calculatoarele personale pot fi conectate în rețele, realizând astfel posibilitatea conversării nemijlocite între oameni de afaceri, proiectanți etc., respectiv posibilitatea comunicării cu un calculator central care poate conține un sistem expert foarte mare sau o bază de cunoștințe extinsă (ca exemplu se pot aminti mai multe sisteme de diagnosticare/depanare accesibile prin internet). Evoluția tehnologică a determinat deci schimbări importante în poziția și rolul calculatoarelor în cadrul organizațiilor. La început ele erau centralizate în mare măsură și erau folosite pentru rezolvarea unui număr restrâns de probleme (de exemplu pentru realizarea evidențelor de salarizare). Dezvoltarea tehnologiei s-a materializat în calculatoare cu dimensiuni reduse, ieftine, performante, care puteau fi amplasate pe birouri. Conform unei analize efectuate s-a demonstrat că dacă automobilele Rolls-Royce s-au dezvoltat în aceeași măsură ca și calculatoarele, atunci ele ar fi dispus la începutul anilor '90 de un consum de combustibil de 3 l la 1000 de km, pe care i-ar fi parcurs „normal” cu o viteză de 800 km/h, prețul lor ar fi coborât sub 5 lire sterline, iar după gabaritul atins ar fi încăput într-o cutie de chibrite (nu se știe însă, cine ar fi avut nevoie de asemenea automobile...). Performanțele din ce în ce mai ridicate și prețurile din ce în ce mai scăzute au condus fără dubii la descentralizarea resurselor de calcul. Conducătorii și oamenii de afaceri sunt în deplinătatea cunoașterii posibilităților de aplicare. Roata s-a întors încet: puterea, forța au ajuns înapoi în centre, dar utilizatorii externi își pot conecta aparatele proprii la aceste centre. Aparatul central poate stoca cantități extraordinare de informații, accesibile utilizatorilor. Aceste informații pot fi date organizate și cunoștințe tratate prin sisteme de gestiune evaluate și sofisticate, sau chiar sisteme expert. Deci și în dezvoltarea acestui domeniu se poate observa un traseu sub formă de spirală (se repetă anumite trăsături, dar la un nivel calitativ și cantitativ diferit).

Cele mai bune sisteme expert sunt caracterizate prin manevrabilitate, utilizare ușoară și performanțe foarte ridicate. În asemenea sisteme expert sunt incluse baze de cunoștințe speciale cu care se pot rezolva în primul rând probleme neformulate numeric. Deși aceste sisteme nu pot concura întotdeauna cu adevărații experți umani, faptul că accesul la ele este ușor, utilizarea lor este facilă și sunt relativ ieftine, reprezintă o provocare serioasă la adresa experților umani. După cum descoperirea tipăririi cărților a revoluționat propagarea cunoașterii, așa revoluționează sistemele de baze de cunoștințe și sistemele expert valorificarea simplă, ușoară și ieftin accesibilă a cunoașterii în viața de toate zilele.

2.3 PRIVIRE GENERALĂ ASUPRA SISTEMELOR EXPERT

2.3.1 Roluri în cadrul sistemelor expert

Am amintit în partea introductivă elementele principale ale unui sistem expert, în contextul utilizării unui mediu de dezvoltare. De acestea se ocupă *shell*-ul, dar utilizatorul este preocupat de aplicație și nu de mediul în care se realizează acesta. Noțiunea de utilizator este folosită frecvent în vocabularul tehnicii de calcul, prin aceasta desemnând persoana situată la sfârșitul lanțului alcătuit din dezvoltator – producător – vânzător – cumpărător – utilizator. Deși calitatea interfeței utilizator (care este diferită de la *shell* la *shell*) este importantă, ceea ce determină în final utilizabilitatea și eficiența aplicației este calitatea bazei de cunoștințe. După cum am mai amintit, baza de cunoștințe poate fi privită ca mulțimea unor reguli. Această afirmație nu este adevărată pentru toate sistemele expert, dar se potrivește majorității sistemelor existente azi în exploatare, iar considerarea alternativelor ar lărgii poate exagerat volumul prezentei lucrări, diluând caracterul aplicativ al conținutului.

Rolul principal aparține bineînțeles utilizatorului (*user*), pentru că el este cel care (pentru rezolvarea problemei sau pentru atingerea scopului propus) așteaptă răspunsul sistemului expert. Este foarte important ca sistemul expert să fie prietenos cu utilizatorul. El nu-și poate permite să se plângă chiar dacă utilizatorul are doar idei aproximative legate de problemă sau de scopul propus. Sistemul adresează pe rând întrebări utilizatorului, sistematizează informațiile cele mai importante primite, explică neobosit rostul întrebărilor puse și scopul investigației. Pe urmă, după furnizarea recomandărilor referitoare la soluționarea problemei utilizatorului sau după atingerea scopului propus de către acesta, sistemul va motiva (în profunzimea solicitată de către utilizator) și va explica de ce a furnizat recomandările comunicate.

Expertul domeniului (*domain expert*) este acel specialist uman documentat, care (posedând cunoștințe profunde și experiență vastă) reprezintă sursa de informații pe cunoașterea căreia se clădește baza de cunoștințe. În unele cazuri se poate întâmpla ca un articol științific sau o carte de specialitate pertinentă să poată înlocui expertul uman al domeniului. De regulă, acesta reprezintă chiar subiectul dorit a fi substituit prin serviciile oferite de-a lungul consultării sistemului expert realizat.

Autorul bazei de cunoștințe, altfel spus expertul cunoașterii (*knowledge base author*), este persoana care pe baza cunoștințelor expertului domeniului realizează baza de cunoștințe. La generatoarele de sisteme expert, integrarea interfeței utilizator și a mașinii de inferență în mediu poate ajuta expertul domeniului să devină el însuși autorul bazei de cunoștințe, acest aspect presupune însă, de regulă, anumite cunoștințe suplimentare (din afara domeniului de specialitate) din partea celui în cauză.

Inginerul cunoașterii (*knowledge engineer*) este acel specialist în sisteme informatice și tehnică de calcul care proiectează și realizează generatorul de sisteme expert. Activitatea acestuia este foarte importantă deoarece astfel avem posibilitatea de a cunoaște sistemele expert și de a realiza aceste sisteme fără a ne afunda într-un cod de programare de nivel inferior (sau foarte apropiat de mașină). Până acum era nevoie de expertul domeniului, care furniza acele cunoștințe pe care expertul cunoașterii le traducea în regulile bazei de cunoaștere. Inginerul cunoașterii alcătuiește deci cealaltă componentă importantă a sistemului expert: interfața utilizator, mecanismul de inferență și baza de cunoștințe. Inginerul cunoașterii trebuie să stăpânească perfect, pe de o parte limbajul de programare ales pentru realizarea sistemului (LISP, PROLOG, C etc.), iar pe de altă parte (bineînțeles) teoria, practica și natura sistemelor expert.

2.3.2 Ce este baza de cunoștințe?

De unde știe omul anumite lucruri? Simplu: înmagazinăm foarte multe fapte, iar unele dintre acestea se leagă de situația în care ne aflăm. De aici apare următoarea întrebare. De unde știm care sunt faptele care se leagă și cum le aplicăm? Memoria umană conform unei viziuni simple se compune din cutii. Fiecare cutie are un conținut și o etichetă. Astfel, de exemplu, cea care are eticheta „zi pentru activități didactice” conține cunoștințele referitoare la zilele în care avem activități didactice. Acest model însă are o mare greșală. Se poate întâmpla ca gândurile (sau o discuție) referitoare la o anumită temă –de exemplu referitoare la mâncăruri– să genereze gânduri despre mâncarea de la bufetul facultății, din timpul zilei de activități didactice. Aparent orice zonă din memoria noastră poate activa orice altă zonă, atâta timp cât poate exista o legătură (fie ea cât de slabă) între ele. După un model mai sofisticat, memoria noastră este colecția extraordinară a informațiilor și a cunoștințelor. Nu ajunge să posedăm cunoștințe, trebuie să știm și la ce se referă ele (adică trebuie să cunoaștem ce anume știm). Ceea ce știm este cunoaștere superficială, care este structurată și organizată. Cunoașterea modului de organizare a cunoștințelor se numește cunoaștere adâncă. Conținutul informațional este important, dar este mai importantă relația dintre conținuturi. Aceste cunoștințe, înmagazinate sub forma unor reguli, se cheamă euristice (cuvântul având origine grecească, desemnând metoda educațională prin care învățăceii descoperă de la sine răspunsurile la problemele date).

Memoria umană poate fi privită ca o mulțime de cunoștințe euristice care nu pot fi identificate și definite ușor, dar dacă dorim crearea unei baze de cunoștințe pentru un sistem expert trebuie să procedăm mult mai exact. Aceste cunoștințe se înmagazinează sub forma unor reguli care conțin perechi condiție–eveniment (dacă condiția este

îndeplinită atunci se realizează operația –prin care evenimentul are loc– deci regula este „executată”), de exemplu:

dacă plouă,
atunci voi lua o umbrelă cu mine.

Condițiile pot fi exprimate în diferite moduri. Pot fi afirmații, de exemplu *dacă plouă*, comparații numerice (= , < etc.), pot fi aplicate asupra unor simboluri al căror conținut este colectat direct din lumea exterioară, pot fi diverse combinații (folosind operatori logici etc). Operațiile, evenimentele pot fi declarații, instrucțiuni și pot determina dirijarea evenimentelor lumii exterioare (de exemplu cazul dirijării producției). Oricare declarație poate fi folosită pentru activarea unei alte reguli. În afara perechii condiție–eveniment, fiecare regulă mai are două părți. În primul rând trebuie să dispună de o etichetă. Această etichetă poate fi una simplă (de exemplu: *regula R1*) sau una explicativă (de exemplu: *regula „ia-ți umbrela”*). Evidența regulilor este simplă într-un sistem compus din zece reguli, dar această preocupare devine din ce în ce mai greu de urmărit în cazul creșterii dimensiunilor bazei de reguli. Din această cauză multe pachete de generare dispun de dicționare de reguli, care permit autorului să marcheze anumite drumuri prin referirea regulilor considerate. Ca exemple pentru ilustrarea ordinelor de mărime: un sistem demonstrativ poate fi compus din aproximativ 50 de reguli; un sistem funcțional poate dispune de 1.000 de reguli. În timp ce un sistem mare are câteva mii de reguli, un om poate deține –după unele aproximări– circa 100.000 de reguli (inclusiv euristice). Deși sistemele expert pot fi foarte mari, și un sistem redus cu 50 de reguli se poate dovedi foarte util.

O altă componentă importantă a regulilor este explicația. Nivelul explicativ este una din diferențele cele mai evidente dintre sistemele expert și programele tradiționale. Un sistem timpuriu pentru calcularea salariilor parcurea stabilirea salariului fiecărui angajat, fără a oferi vreun ajutor utilizatorului. Este adevărat că sistemul a fost proiectat să fie autonom, din cauza aceasta rulează ușor, fără oprire. Pachetele mai noi de programe au un sistem de ajutorare inclus. Acesta oferă explicații referitoare la comenzile accesibile, dar nu oferă nici o explicație referitoare la ceea ce face (și de ce face) sistemul la un moment dat. Sistemele expert sunt capabile într-o anumită măsură să-și explice comportamentul, dar explicațiile trebuie clădite în sistem. Sistemele expert care există azi nu sunt capabile să realizeze propriul sistem ajutător pe baza regulilor din baza de cunoștințe. Acest serviciu este o posibilitate rezervată sistemelor viitoare. O regulă completă ar fi deci asemănătoare cu următoarea:

etichetă	R26 (luarea umbrelei)
condiție	dacă (plouă)
eveniment	atunci (luați umbrela)
explicație	deoarece folosind umbrela puteți evita să vă udați.

Se poate afirma că lumea nu poate fi descrisă simplu printr-un set de reguli. Fără îndoială

Însă, priceperea se materializează de obicei prin reguli. O mare parte a activității noastre umane poate fi privită ca reguli sau elemente euristice și poate fi încorporată în baza de cunoștințe a unui sistem expert. *Inteligența* sistemului expert este determinată în primul rând de procedeul prin care mașina de inferență leagă regulile.

2.3.3 Mașina de inferență

Exemplul anterior (cu umbrela) este util, dacă plouă, dar un domeniu de cunoaștere alcătuit doar din câteva reguli nu necesită un sistem expert. Dacă însă numărul regulilor crește, la zeci, sute sau mii, atunci crește și complexitatea. Nu numai din cauza numărului de reguli, dar și din cauza relațiilor dintre reguli. Extinzând puțin exemplul anterior al umbrelei (pentru ilustrarea deducției înlănțuite înainte și înapoi), putem stabili/considera câteva reguli:

- | | |
|-----|--|
| R26 | Dacă (plouă),
atunci (luați umbrela); |
| R34 | Dacă (cerul este înnorat și trotuarul este ud),
atunci (plouă); |
| R43 | Dacă (cerul este gri),
atunci (cerul este înnorat); |
| R67 | Dacă (luați umbrela),
atunci (uitați umbrela în tren); |
| R88 | Dacă (lăsați urme),
atunci (trotuarul este ud). |

Mașina de inferență dă sens acestor reguli prin legarea lor, pentru a putea adresa utilizatorului întrebările corespunzătoare și pentru a putea realiza concluziile corespunzătoare. Setul prezentat ne poate ajuta să înțelegem dacă avem sau nu nevoie de umbrelă. În acest caz, chestionarea ar începe prin aplicarea întrebării cuprinse în regula R43: *Cerul este gri?*, care ar activa regula R88: *Lăsați urme?*, urmând activarea succesivă a regulilor R34, R26 și R67. Consultarea regulilor ne dă rezultatul *Uitați umbrela în tren* (conform aplicării strategiei de urmărire înainte). Noi însă am dori să întrebăm ce condiții au dus la uitarea umbrelei în tren. Mașina de inferență va parcurge drumul înapoi de la regula R67 la R26, apoi la R34, R43, R88 și la întrebarea *De ce am uitat umbrela în tren?* va răspunde: *Pentru că lăsați urme și cerul este gri* (conform strategiei de urmărire înapoi). Acest exemplu este oarecum hazliu, dar cred că ilustrează diferența dintre deducția prin înlănțuire înainte (*forward chaining*) și deducția prin înlănțuire înapoi (*backward chaining*).

Dacă numărul răspunsurilor posibile este cunoscut și redus, atunci deducția prin înlănțuire înapoi este foarte eficace. Dacă, însă, numărul răspunsurilor posibile este foarte mare, atunci se recomandă deducția prin înlănțuire înainte. Cele mai multe sisteme expert înclină spre folosirea inferenței sub forma deducției înlănțuite înapoi, deoarece rezultatul este cunoscut și este nevoie de factorii care conduc la acel rezultat (cazul clasic al sistemelor de diagnosticare, de exemplu: un cercetător poate fi interesat de aflarea acelor factori care conduc la apariția cancerului pulmonar).

Baza de cunoștințe dispune de două dimensiuni. Această caracteristică determină capacitatea sistemului de a efectua pași nu numai înainte (prin înlănțuirea înainte) și înapoi (prin înlănțuirea înapoi) ci și lateral, adică să consulte regulile aflate la același nivel în baza de cunoștințe. Dacă sistemul este foarte mare, atunci utilizatorul nu poate efectua căutări complete în baza de cunoștințe, din cauza limitărilor de timp sau a performanțelor calculatorului. Acesta înseamnă o căutare relativ restrânsă, deci nu se verifică întotdeauna toate posibilitățile. Este foarte posibil ca mașina de inferență să execute căutarea întâi în adâncime. Se începe prin întrebarea de deschidere după care se avansează, prin reguli urmărind înlănțuirea lor înainte, până la obținerea unui răspuns. Acest răspuns, bineînțeles, se poate întâmpla să nu fie cel mai bun. Dacă mașina inferențială ar fi parcurs o altă serie de întrebări (și reguli) am fi putut obține un alt răspuns. Cele mai multe programe funcționează după acest model, dar în sistemele mai rafinate mașina inferențială permite utilizatorului să aleagă modalitatea chestionării.

Pentru a ilustra diferența dintre cele două abordări, putem să ne imaginăm un specialist, față de un om care posedă doar cunoștințe generale. Primul va pune întrebări legate de –din ce în ce mai multe– detalii, până când se convinge asupra rezultatului. Cel de al doilea va pune întrebările la întâmplare, într-un mod aparent nepotrivit, până la rezolvarea (într-un fel sau altul a) problemei. În cadrul unui sistem mic, unde există posibilitatea căutării fiecărei reguli, procedeul de întrerupere a căutării nu este necesar, deși procedeele destinate scurtării timpului de căutare (fără a renunța la găsirea rezolvării optime) nu se pot ignora. Unul din aceste procedee (care permite utilizatorului experimentat reducerea timpului de căutare) este furnizarea fără cerere expresă a unor informații sistemului. Aceste informații pot preîntâmpina folosirea unor grupuri mari de reguli (de exemplu, în cadrul unui sistem pentru planificarea concediului de vară, putem evita toate întrebările referitoare la condițiile meteorologice dacă precizăm de la început că acestea ne sunt indiferente).

2.3.4 Utilitatea sistemelor expert

Dacă examinăm câteva detalii ale sistemelor expert putem observa lesne avantajele acestei abordări față de abordarea tradițională în rezolvarea problemelor. Aplicațiile timpurii de calcul foloseau limbaje destinate manipulării valorilor numerice. În consecință, aplicațiile erau axate pe prelucrări numerice și erau potrivite pentru calcule ingineresti, scopuri științifice sau economico-financiare. Cum –prin dezvoltare– tehnica de calcul a devenit mai ieftină și mai eficace, au apărut procesoarele de texte și gestionarele bazelor de date. Următoarea etapă de dezvoltare a fost caracterizată de înmagazinarea, interpretarea și obținerea cunoștințelor, ce s-a materializat sub forma sistemelor expert. Acestea au fost inițial definite ca „programe de calculator care furnizează sfaturi experte”. Nu putem afirma că scopul sistemelor expert este înlocuirea completă a experților sau a minții umane, dar trebuie să recunoaștem că metodele de calcul utilizate sunt cu totul diferite față de cele cu care ne-am obișnuit.

Utilizatorii pachetelor de programe tradiționale, de regulă, nu au cunoștințe despre programele folosite –dacă acestea au fost realizate de către altcineva. Numai autorii

pachetului înțeleg complet funcționarea programului, respectiv ipotezele fundamentale. Sistemele expert sunt ceva mai transparente, ele îngăduind de regulă utilizatorului înțelegerea funcționării interne a sistemului. Ele sunt foarte îndepărtate de concepția „cutiei negre”. Aplicațiile prin sisteme expert trebuie să fie atât de comprehensibile încât să permită utilizatorului înțelegerea modului în care lucrează sistemul. Acest aspect este sprijinit puternic de nivelurile explicative accesibile. În timpul consultării se realizează o legătură interactivă –relativ strânsă– între sistem și utilizator, și din această cauză utilizatorul are impresia că sistemul este „mai comod”. Dacă apar probleme, utilizatorul poate consulta informațiile ajutătoare și explicative. Sistemul este capabil să urmărească pașii făcuți în baza de cunoștințe înapoi și să explice cum a ajuns la situația dată.

Cunoașterea principiilor fundamentale și posibilitatea obținerii explicațiilor determină caracterul comprehensibil al sistemului pentru utilizator, de aici derivând și încrederea mai mare a utilizatorului în sistem. Adjectivul „prietenos” (*user friendly*), folosit frecvent în trecut, nu făcea sistemele mai transparente pentru utilizator, desemna mai degrabă sisteme coborâte la nivelul utilizatorului (instrucțiunile din cadrul acestora fiind cele mai simple posibile). Utilizatorului trebuie să i se permită folosirea simplă a sistemului expert, dar și înțelegerea acestuia. Deoarece aceste sisteme sunt complet interactive (accentuând la ele caracterul cu adevărat „prietenos”), ele reprezintă instrumente ideale pentru realizarea sistemelor de învățare, instruire, cu ajutorul cărora cel care învață își poate însuși cunoștințele fără a fi încurcat, amenințat sau fără a avea impresia că este condus de o „cutie neagră”.

Programele tradiționale de aplicații erau create în urma unei analize efectuate de către un specialist, analiză urmată de proiectarea de către un alt specialist și programarea de către un programator –utilizatorul folosindu-l doar. Acest proces părea să fie eficient, dar în realitate îngreuna destul de mult viața. În aceste sisteme erau atât de multe relații, conexiuni, încât utilizatorul nu putea fi partener adevărat în procesul de dezvoltare. Orice schimbare a sistemului presupunea rescrierea unor părți considerabile de program, iar acesta influența și alte părți din sistem. Împreună cu creșterea sistemului crește și numărul posibilităților de aplicare. Sistemele expert sunt mai flexibile, cu mai multe fațete. Pot porni cu un domeniu mic, special, al cunoștințelor (cu o parte din cunoaștere), putând crește în timp. Nu numai sub aspectul extinderii bazelor de cunoștințe, dar și prin posibilitatea împlinirii acestora. Deoarece cunoașterea este un set fără limite de reguli, extinderea este nelimitată practic. Mașina de inferență tratează din ce în ce mai multe reguli, deoarece orice s-ar întâmpla cu regulile ea rămâne neschimbată. Numărul regulilor poate crește, regulile pot fi schimbate destul de ușor, putând astfel insera schimbările cunoașterii în sistem. Programul nu trebuie rescris (ceea ce ar putea fi și dificil și periculos, deoarece programele tradiționale nu sunt foarte inteligibile pentru cititorii din urmă). Schimbarea conținutului de reguli al unor părți nu trebuie să influențeze celelalte părți ale sistemului.

2.3.5 Limitările sistemelor expert

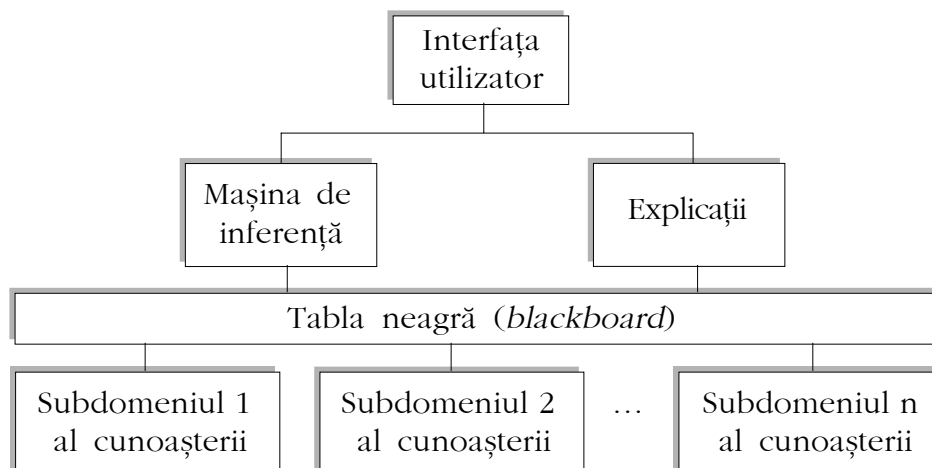
Dacă sistemele expert diferă în modul prezentat de programele tradiționale, de ce nu s-

au răspândit într-un cerc mai larg? Este evident că orice rezolvare își are limitele, nefăcând excepție de la această afirmație nici sistemele expert. Dezvoltarea *hardware*-ului și *software*-ului este mai rapidă decât capacitatea de asimilare a pieței. S-a afirmat în mai multe rânduri că sistemul expert este o aplicație care caută probleme. Există o doză de adevăr în această afirmație, mai ales dacă recunoaștem că sistemele expert s-au dezvoltat mai mult din munca efectuată asupra limbajelor de programare și asupra aplicațiilor inteligenței artificiale, și mult mai puțin din rezolvarea unor probleme reale. Părerea mea este că, deși pe piață se înregistrează o anumită mobilitate a acestor sisteme, ele nu au depășit încă complet stadiul dezvoltării. Mediile de generare (*shell*) accesibile azi sunt mult mai performante și mai dezvoltate decât cele de acum câțiva ani. Cred că încă nu se poate stabili dacă sistemele expert reprezintă o parte a procesului evolutiv, sau zorii tehnicii noi de calcul.

Dezvoltarea oricărui sistem necesită un timp considerabil, dar acest timp este scurt în comparație cu durata utilizării. Sistemul este practic oglindirea cunoștințelor unei anumite perioade. Dacă baza de cunoștințe este bine organizată și documentată corespunzător, atunci ea poate fi actualizată, extinsă, dar modelul va rămâne același pe toată durata de viață a sistemului. Fiecare bază de cunoștințe trebuie concepută astfel încât să se acomodeze mediului de generare pentru sisteme expert, și presupune o viziune proprie asupra lumii. Această viziune poate fi corectă azi, dar mâine se poate dovedi incorectă (de exemplu îmbunătățirea cea mai mare în situația sănătății populației a fost rezultatul apei potabile curate și a canalizării corespunzătoare, factori externi pentru știința medicală). De factorul timp se leagă și unul dintre instrumentele dibace ale sistemului expert. Mașina de inferență (instrumentul rezolvitor) este clădită în sistem, ea poate fi evidentă azi, însă mâine poate fi mai puțin utilă. De exemplu, dacă ne gândim la scară mare: a existat o trecere de la epoca de piatră la epoca bronzului ceea ce a avut ca și consecință înlocuirea cvasicompletă a tehnologiilor. Sau, la scară mult mai redusă: utilizarea microtehnologiei la mașinile de spălat a determinat înlocuirea pieselor electromecanice nefiabale anterioare. Sistemele expert au totuși, de regulă, avantajul că din cauza alcătuirii pot fi actualizate și modificate mai ușor.

Folosirea unui singur expert la realizarea setului de reguli presupune cunoașterea tuturor regulilor de către acel expert. Deasemenea, se presupune că ne-am clarificat asupra tuturor regulilor. Cu alte cuvinte, am putea declara că nu are importanță alegerea expertului uman, pentru că oricare ar fi el ar realiza același set de reguli. Acest punct de vedere este însă eronat. Probabil există o înțelegere largă între experții umani, dar ajungerea la un numitor comun (obținerea înțelegerii complete) între ei este foarte dificilă, aproape imposibilă. Cercetătorii au recunoscut problema unificării părerilor divergente și au dezvoltat diverse procedee tehnice pentru unificarea cunoștințelor experte.

De exemplu, prin anii 1970 a apărut conceptul procedeeului „tabla neagră” (*blackboard*), în care se inventariază toate opiniile divergente, încercând găsirea unor relații comune înainte de a cuprinde acestea în baza de cunoștințe. Dezvoltarea și perfecționarea acestui concept continuă și azi. Practic o asemenea tablă neagră a ajuns o bază mare și compusă de date, care ține evidența conținutului bazelor de cunoștințe ale domeniilor parțiale ce

Fig.2 Structura cu *blackboard*

formează domeniul dat (figura 2). Pentru tratarea complexității caracteristice acestui procedeu sunt însă necesare calculatoare ce suportă prelucrări paralele corespunzătoare.

Noțiunea de sistem expert presupune implicit existența unei soluții optime sau cel puțin viabile. Se pot însă imagina unele probleme care sunt nerezolvabile și fără răspuns la nivelul actual al cunoștințelor noastre. În aceste cazuri sistemele expert nu au cum să ne furnizeze răspunsuri. Se poate întâmpla ca răspunsul furnizat de către sistemul expert să nu fie utilizabil (de exemplu sistemul expert ne poate sfătui să drenăm albia râului în locul construirii unui pod) sau acesta să fie utilizabil însă considerat inacceptabil de către utilizator. Ce se întâmplă în acest caz cu credibilitatea sistemului? Fiecare sistem trebuie să se poată adapta elastic la viziunea proprie unică a utilizatorului. Sistemele expert presupun existența unei lumi perfecte, însă o asemenea lume nu există încă. De aceea ne străduim să obținem mai mult soluții acceptabile decât soluții optime, pentru că lucrăm cu resurse limitate. O soluție –chiar optimă– are puține șanse să fie la fel de bună pentru toți. În general nu există timp, bani, informații, instrumente sau forțe de lucru suficiente pentru obținerea soluțiilor perfecte.

2.4 PRIVIRE GENERALĂ ASUPRA BAZELOR DE CUNOȘTINȚE

2.4.1 Reprezentarea cunoștințelor

Problema reprezentării cunoștințelor din domeniul de expertiză considerat stă în centrul preocupărilor proiectanților de sisteme expert, această reprezentare determinând dezvoltarea sistemului și interferându-se cu strategiile de control. Pot fi utilizate mai multe maniere de reprezentare pentru aceleași cunoștințe, alegerea modalității potrivite poate fi în funcție de natura problemei, de instrumentația disponibilă, de oportunitate, de intuiție etc., reprezentarea căzând în sarcina inginerului de cunoștințe. Prima fază din activitatea inginerului de cunoștințe este achiziția cunoștințelor. Pe parcursul acestei achiziții se identifică conceptele fundamentale ale domeniului de expertiză, se asimilează regulile, strategiile și raționamentele expertului uman. Următorul pas este structurarea acestor cunoștințe (schematizarea fluxului de informații, legăturilor dintre concepte, înlănțuirea

raționamentelor, ordinea de aplicare a regulilor etc.), după care se poate alege un sistem de reprezentare al cunoștințelor. Acesta va depinde de regulă, în cea mai mare măsură, de sistemul disponibil atunci când cel care elaborează sistemul expert va dispune de software-ul necesar. Dacă nu există sistem disponibil, alegerea va depinde de termenul de realizare impus și de resursele financiare ale celui care a comandat sistemul. Cunoștințele pot fi adaptate —mai mult sau mai puțin ușor— și unor șabloane deja existente. Cu cât șablonul este mai potrivit, cu atât vor fi mai convingătoare performanțele sistemului, elaborarea va fi mai rapidă, iar întreținerea mai facilă.

Faptele sunt reprezentate de regulă (din punct de vedere informatic) ca niște șiruri structurate de caractere. Un procedeu răspândit este folosirea unor *obiecte cu valoare atribuită*, de genul:

<atribut> <obiect> <valoare>

unde atributul este o proprietate a obiectului. Această reprezentare a devenit clasică prin succesul sistemului EMycin (derivat din Mycin). Se poate utiliza și sistemul echivalent în esență, dar mai simplu (care permite și folosirea unor variabile pe post de operator):

<obiect> <operator> <valoare>

Predicatele sunt folosite de regulă pentru exprimarea *relațiilor* dintre obiecte. Ele pot fi de ordine diferite. Considerând un exemplu imaginar: pentru a exprima faptul că firma CONS deține 20 de unități din capitalul societății CELUX, putem scrie sub forma unei funcții cu două variabile:

Parte_capital (CONS, CELUX) = 20

sau folosind forma obiectelor cu valoare atribuită:

<CONS-POSEDĂ> <CELUX> <20>.

O funcție de n variabile se poate scrie oricând sub forma unui predicat de ordinul $n+1$, unde argumentul al $n+1$ -lea va fi valoarea funcției. Deci, relația exprimată mai înainte printr-o funcție cu două argumente se poate scrie sub forma unui predicat de ordinul trei:

Structură_capital (CONS, CELUX, 20).

Reducând această expresie sub forma unor predicate de ordinul doi se obține:

Posedă (CONS, \$1)
Este_posedat (CELUX, \$1)
Unități (20, \$1)

unde am introdus un obiect fictiv (\$1) care poate fi interpretat ca un act pentru care factorii

implicați sunt detaiați cu ajutorul predicatelor de ordinul doi (această reducere de predicate de ordin superior la predicate de ordinul doi se întâlnește și în SNARK). O funcție se poate restrânge deci la predicat, iar predicatele de ordinul doi pot exprima predicate de ordin superior. Aceasta explică faptul că majoritatea sistemelor existente se restrâng la predicate de ordinul doi și la relații binare. Calculul predicatelor ca mijloc de reprezentare a cunoștințelor este o idee caracteristică limbajului PROLOG.

Pentru a regrupa în mod adecvat numeroase informații legate între ele, se pot utiliza ceea ce Minsky a numit *frame*-uri. Un *frame* (în limba română se mai utilizează denumirile: *cadru*, *formă-obiect* etc.) este o unitate de informație care regrupează un număr de *slot*-uri (altfel spus: *rubrici*, *locașuri* etc.). Cadrele reprezintă o modalitate agreabilă de gestionare a *valorilor implicite*, adică a valorilor asociate unei rubrici, atunci când utilizatorul nu are informații precise pentru a indica o valoare. La fiecare rubrică sunt atașate proceduri care gestionează valoarea rubricii (într-o asemenea rubrică pot figura și apeluri către proceduri externe de calcul). Astfel fiecare cadru „știe” care sunt valorile dependente situate în alte cadre care trebuie schimbate atunci când conținutul unei rubrici se modifică. Unele actualizări de acest fel pot declanșa o procedură externă sau „demon”, care este atașată unui cadru. În plus, fiecare rubrică posedă procedurile sale care sunt activate la citirea/solicitarea (*if-needed*), adăugarea (*if-added*), suprimarea (*if-removed*) unei valori. În reprezentarea cunoștințelor prin cadre, noțiunea propriu-zisă de regulă dispare. În timpul sesiunii, rețeaua de cadre este parcursă (și modificată dacă este cazul) din aproape în aproape cu ajutorul procedurilor atașate rubricilor. Această cvasi-dispariție a regulilor nu contribuie la transparența bazei de cunoștințe și nu ușurează scrierea sistemului. De aceea, sistemele bazate pe cadre reintroduc adesea regulile (sub forma unor cadre particulare). De multe ori se întâmplă să coexiste regulile „clasice” și cadrele într-un sistem expert.

În ciuda lipsei amintite de transparență, reprezentarea prin cadre are două avantaje (existente și la câteva forme de reprezentare diferite): realizează organizarea unei rețele relaționale prin dependențele dintre fapte, și partiționează (sau compartimentează) aceste fapte – operație utilă în faza de restricționare sau de filtrare. Utilitatea clasificărilor apare mai ales la organizarea regulilor sau la introducerea metaregulilor. Noțiunea de *rețea* sau altfel spus *graf de obiecte* provine din analiza limbajului, de unde și denumirea de *rețea semantică* (legată de înțelesul frazelor) atribuită unei asemenea rețele din obiecte. Frazele limbajului natural, compuse din obiecte (grupul subiect+verb, grup complement etc.), se pretează foarte bine acestei reprezentări. Aceste reprezentări sunt utilizate de mai mult timp de către analiști, tehnologia sistemelor expert reluând doar idea pentru structurarea cunoștințelor de specialitate. O bună structurare a rețelei semantice evită pe cât posibil circuitele în graf, tinzând spre o organizare ierarhică a obiectelor. O astfel de organizare se traduce printr-o arborescență de fapte sau de fapte primordiale (elementare), ceea ce ușurează structurarea domeniului.

Mergând pe această cale, cea din urmă idee este reprezentată de arborescența clasificării sau *taxonomie* (similară cu cele existente în botanică sau zoologie). O taxonomie este

o arborescență în care există proprietăți de *moștenire*. Clasificarea se face de regulă de la general la particular (de exemplu: *grinda* face parte din *cadrul* care este o *structură*). Noțiunea de taxonomie se utilizează în unele cazuri și pentru grafuri ierarhizate – care nu sunt neapărat structuri arborescente – în care există proprietăți de moștenire (dar pot exista mai multe moduri de a ajunge la același obiect, mai ales pentru nivelurile inferioare ale ierarhiei). O taxonomie bine concepută poate să reducă considerabil numărul caracteristicilor atașate în mod explicit unui obiect.

De fapt, orice arborescență din reprezentarea faptelor sau regulilor poate fi folosită de către mașina inferențială. Chiar dacă arborescența se aplică inițial doar la fapte, structurarea aleasă acționează în mod inevitabil și asupra regulilor (alternativele de reprezentare ale regulilor și ale faptelor sunt strâns legate). Rețelele celor mai multe sisteme bazate pe cadre sunt organizate sub formă ierarhică cu moștenire.

Este important de menționat că toate aceste sisteme de reprezentare a conceptelor (obiectelor) sunt inspirate mai mult sau mai puțin din caracteristicile umane. Acest fapt este confirmat și de psihologia cognitivă, care are ca obiect de studiu reprezentarea cunoștințelor la om. Astfel, existența unităților de cunoaștere care comportă mai multe elemente, fie la nivel descriptiv fie la nivel operațional, este atestată în literatura de specialitate. Majoritatea acestor noțiuni (de exemplu: model, obiect, plan, prototip, scenariu, schemă, script, spațiu, unitate etc.) sunt destul de asemănătoare cu cea a cadrului. Este adevărat însă că noțiunea de *prototip* în psihologia cognitivă desemnează un obiect reprezentativ al unei categorii. În această accepțiune, celelalte obiecte ale categoriei sunt instanțieri diferite ale prototipului (sau ale cadrului tipic).

2.4.2 Reprezentarea regulilor

Nu întotdeauna este evidentă diferența dintre fapte și reguli, distincția acestora depinzând în mare măsură de reprezentarea adoptată. Problema esențială a sistemelor bazate pe reguli este modul de structurare al ansamblului de reguli pentru obținerea unei eficacități maxime în procesele de filtrare și control. Atunci când există o structurare arborescentă a domeniului (sau o taxonomie) se poate utiliza acesta pentru a construi o *rețea de reguli* (sau o *rețea de inferențe*, un *graf de precedență a regulilor*) care indică ordinea de aplicare a regulilor. Rețelele de inferență furnizează o reprezentare grafică agreabilă care ajută expertul să-și vizualizeze structura cunoașterii pentru a urmări raționamentele (ca în sistemul NEXPERT-OBJECT). Eventual, crearea unor asemenea legături relaționale între reguli ne poate scăpa de operațiile de filtrare hazardate, ameliorând performanțele sistemului. Pe de altă parte *regruparea regulilor* în pachete servește ca suport pentru operațiile restrictive. Însă, în majoritatea sistemelor comercializate în prezent, regruparea regulilor în pachete este mai puțin indicată și se utilizează doar dacă este impusă de reprezentarea cunoștințelor adiacente sistemului. În sistemele de tip *blackboard*, regulile sunt regrupate în module *KS* (*Knowledge Source*), cum ar fi de exemplu în HEARSAY-III și AGE. În OPS 5 însă, regulile sunt însoțite de numele unei clase care este destinată să faciliteze operația de filtrare.

Noțiunea de graf semantic și regruparea regulilor este utilizată și în sistemul ARGUMENT, idea principală în acest sistem fiind dependența funcțională între concepte, denumită *schemă*. Graful ierarhizat este astfel un *arbore de scheme*. Noțiunea de schemă generalizează conceptul de funcție pentru obiecte nenumerice. Regulile realizează definirea punct cu punct a acestei funcții (sau a grafului corespunzător). Prin această reprezentare, partea cea mai importantă a reprezentării cunoștințelor rezidă în construcția rețelei semantice sau a arborelui de scheme. Cea mai mare parte a controlului este efectuată asupra acestui arbore de scheme și nu asupra regulilor, ceea ce accelerează filtrarea, mărinde performanțele sistemului. Rezultatul este același cu cel obținut prin precompilarea regulilor, cu observația că rețeaua este semantică, adică are un sens pentru utilizator (nu are numai un sens pur informatic). Regruparea regulilor în scheme asociază două idei fundamentale în reprezentarea cunoștințelor: ierarhizarea și asociativitatea. Asociativitatea apare în dependența funcțională ca noțiune universal înțeleasă. Ierarhizarea este prezentă deoarece cunoștințele umane, atât cele generale cât mai ales cele speciale (ale experților), par să fie, într-o măsură mai mare sau mai mică, organizate în acest fel.

Există în multe cazuri și reguli pentru controlarea pachetelor de reguli. Aceste reguli, care reprezintă practic niște cunoștințe despre reguli, se numesc *metareguli*. O metaregulă este scrisă în general ca o regulă „obișnuită”. De obicei gestiunea metaregulilor se bazează pe o mașină inferențială specială. Seturile de reguli care sunt dirijate de metareguli sunt în mod obișnuit definite de către expert (ele fac parte din baza de cunoștințe). De fapt, toate organizările de reguli și/sau de obiecte reprezintă metacunoștințe. Această deosebire între nivelurile de cunoștințe a condus la redescoperirea noțiunilor de niveluri de reprezentare. Acestea sunt dificil de fixat și de inventariat pentru că depind în mare măsură de domeniul ales și de problema pusă. Singurul lucru de care putem fi convinși este că aceste niveluri există și probabil nu pot fi limitate ca număr (deși deocamdată ne ajunge să tratăm doar două). De asemenea, este dificil de explicat prin ce diferă metacunoștințele de cunoștințe, dacă nu sunt privite din punctul de vedere al rolului tehnic din cadrul unui sistem informatic.

Există și metode de reprezentare destinate în special reprezentării cunoștințelor la nivelul bunului simț, cunoștințelor vagi sau excepțiilor. Dacă ne interesează rezolvarea unei probleme reale, experiența a demonstrat că metodele amintite pot fi satisfăcătoare și suficiente. Multe din problemele teoretice, precum excepțiile sau non-monotonia, au fost rezolvate sau pot fi ocolite de către expert. De exemplu, în majoritatea proceselor temporale non-monotonia nu reprezintă o problemă dacă mecanismul de inferență acționează mai rapid decât timpul necesar modificării fizice semnificative a sistemului. Problemele de excepție pot fi stăpânite cu succes de către expert, noțiunea de excepție neavând de altfel sens decât în raport cu contextul (excepția depinde de funcția de utilitate a celui care o consideră). Alegerea metodei de reprezentare a cunoștințelor depinde și de criterii ca: transparența (lizibilitatea) bazei de cunoștințe, facilitatea ingineriei cunoașterii, disponibilitatea și costul unui instrument de elaborare. De cele mai multe ori aceasta este o problemă de obișnuință și de oportunitate.

2.4.3 Cunoștințele incerte și aproximative

Prima modalitate de considerare a incertitudinilor și/sau aproximațiilor în cadrul sistemelor expert este introducerea acestora în obiectele cărora li s-a atribuit o valoare. Acestea pot avea atât valori numerice (sub formă de fracții, procente etc.) cât și valori lingvistice (în acest ultim caz ele fiind evaluate de regulă indirect). Se pot introduce probabilități și statistici fără probleme atât în fapte cât și în reguli. În fond, este de ajuns să discretizăm scările de probabilități și de valori (adică să considerăm un număr finit de „trepte”) pentru a putea lua în calcul toate situațiile posibile, scriind un număr finit de reguli. Aceasta presupune introducerea unor cunoștințe de genul:

$$\begin{aligned} \text{Dacă } 0,35 \leq \text{PROB} \leq 0,45 \\ \text{Atunci } \text{PROB} = 0,4. \end{aligned}$$

Abordarea se poate simplifica prin utilizarea variabilelor. Aceste considerații simpliste (dar operaționale) nu sunt întotdeauna satisfăcătoare, în special în domeniul diagnosticării. Acesta este motivul pentru care mai multe sisteme expert (de ex. Mycin, Prospector etc.) au introdus noțiunea de *inferență incertă*. Să considerăm spre exemplu următoarea inferență:

$$\text{Dacă } F_1 \text{ și } F_2 \text{ și } F_3 \text{ atunci } E_k$$

unde F_1 , F_2 și F_3 sunt fapte (observate sau demonstrate) și E_k este un diagnostic (sau un fapt dovedit). Atunci când evenimentul $F = \{ F_1, F_2, F_3 \}$ este observat, probabilitatea ca E_k să fie concluzia cea potrivită dintre $E_1, E_2, \dots, E_n$ concluzii disjuncte posibile, este dată de formula lui Bayes:

$$P(E_k | F) = \frac{P(F | E_k) \cdot P(E_k)}{\sum P(F | E_i) \cdot P(E_i)}$$

În această expresie $P(F | E_k)$ este probabilitatea de observare a lui F atunci când este formulat diagnosticul E_k (când E_k este adevărat). Este o probabilitate „a priori” care poate fi dedusă din serii statistice sau dintr-o estimare precedentă a expertului uman. Atunci când probabilitățile se bazează pe estimarea unui expert uman, este vorba de *probabilități subiective*. Formula lui Bayes poate fi interpretată și ca transformarea probabilității apriorice $P(E_k)$ în probabilitatea posterioară $P(E_k | F)$ prin observarea lui F . Acest model însă, prezintă la rândul lui mai multe deficiențe criticabile (și criticate):

- precizarea probabilităților apriorice $P(E_k)$ este dificilă, cu atât mai mult cu cât subiecții de cele mai multe ori nu raționează în domeniul incertitudinii cu probabilități subiective (după cum dovedesc experimentele);
- nu se poate calcula $P(F | E_k)$ din $P(F_i | E_k)$ dacă nu se acceptă ipoteza independenței condiționale a faptelor F_i . Această ipoteză este aproape întotdeauna admisă în practică, deși are puține șanse de a fi satisfăcută în realitate.

Anumite sisteme au încercat să evite utilizarea formulei lui Bayes sau au adaptat-o. În sistemul Mycin există un coeficient de verosimilitate (o valoare numerică cuprinsă între 0 și 1) aplicat asupra regulilor. Asupra faptelor s-a introdus o incertitudine care nu ține seama de modelul lui Bayes, fiecare fapt fiind prevăzut cu un coeficient de certitudine.

O altă abordare se bazează pe noțiunile de *posibilități*. Aceasta se poate considera o generalizare a noțiunii de probabilitate [58]. Obiectivul acestor modele este să schimbe logica de „totul sau nimic” conținută în *modus ponens*. În acest scop se definesc *modus ponens* generalizate.

În fine, rămân cunoștințele aproximative exprimate în valori lingvistice, de genul: *viteza este mare*. Dacă este peste 100 km/h ea este considerată mare, sub 80 km/h nu este mare. Între aceste valori se poate atribui un *grad de apartenență* faptului că viteza este mare, conform teoriei mulțimilor vagi a lui Zadeh. Manipularea unor astfel de graduri (funcții) de apartenență în sistemele expert permite evitarea efectelor de prag (câteodată neplăcute în dirijarea procedurii). În practica dirijării (controlării) procedurilor, pragurile –mai mult sau mai puțin arbitrar– există, pentru că întotdeauna există un moment în care operatorul decide cum să acționeze (fără să fi făcut ceva până atunci), această acțiune fiind de multe ori discretă. Prelucrarea cunoștințelor vagi și a incertitudinilor sub forme declarative nu este încă foarte răspândită din cauza modelelor care utilizează coeficienți de verosimilitate și de certitudine (dar își face simțită prezența din ce în ce mai mult).

2.4.4 Limbajele de programare și reprezentarea cunoștințelor

Limbajul care poate fi utilizat, pentru a scrie programele unui sistem expert, nu depinde neapărat de forma adoptată pentru reprezentarea cunoștințelor. Au fost și sunt elaborate sisteme expert în limbajele BASIC, COBOL, FORTRAN, APL, PL/1, PASCAL, LISP, PROLOG, C etc. Cu toate acestea, există două limbaje care ar putea revendica rolul de limbaje de inteligență artificială: LISP și PROLOG.

Limbajul PROLOG (PROgrammation en LOGique) este cel care a fost ales de către japonezi pentru programul de dezvoltare a calculatoarelor din generația a cincea. A fost dezvoltat mai întâi la Universitatea din Marsilia (în perioada 1970–1980), apoi la Universitatea din Edinburgh. Este un limbaj bazat pe logica de ordinul întâi, logica predicatelor. Acest limbaj este prototipul limbajelor declarative, prin opoziție cu programarea imperativă sau procedurală (limbajele imperative sunt cele în care programatorul descrie pas cu pas un algoritm, precizând prin înșiruirea instrucțiunilor calea exactă –de regulă invariabilă– a programului). Programarea declarativă constă în furnizarea dezordonată a datelor și regulilor în cadrul programului, iar programul „se descurcă” cu ajutorul mecanismului de inferență încorporat. Există mai multe versiuni ale limbajului (variante cu restricții, extensii ale limbajului, combinații cu alte limbaje etc.), precum și adevărate medii bazate pe acest limbaj, cu strategii de control diferite.

Limbajul LISP (LISt Processing) a apărut la sfârșitul anilor 1950 la M.I.T. (S.U.A.). În acest

limbaj se manipulează *atomi* și *liste*. Fiecare listă este organizată sub formă de arborescență binară —de exemplu, operația $3 \times (2 + 4)$ se va scrie $(x\ 3\ (+\ 2\ 4))$. Primul element al listei este interpretat ca un operator (sau ca o specificație de funcție) care se aplică elementelor următoare din listă. Scrierea regulilor este ușoară, ca de altfel și prelucrarea și compararea listelor, ceea ce este foarte util pentru manipularea arborescențelor și pentru operația de filtrare. Limbajul permite o modulare ridicată, ceea ce reprezintă un atu pentru realizarea sistemelor expert. Performanțele mediocre ale interpretoarelor și compilatoarelor LISP pe calculatoare tradiționale (din punctul de vedere al timpului de răspuns) au determinat realizarea unor calculatoare dedicate. Aceste calculatoare dedicate (conținând "procesoare LISP") cedează însă progresiv locul stațiilor de lucru care vizează publicul larg și care —datorită progresului microprocesoarelor— au performanțe comparabile cu cele ale calculatoarelor LISP. Din cauza vitezei de execuție, a portabilității și a difuziunii sistemului de operare UNIX, limbajul care domină în prezent realizarea sistemelor expert este C —un limbaj orientat pe obiecte (manipulează entități autonome de aceeași tip). Cercetările continuă și azi pentru dezvoltarea limbajelor orientate pe obiecte și cât mai interactive. Cel mai vechi din acest domeniu este probabil SmallTalk, realizat între anii 1970 și 1976, pornind de la limbajul SIMULA (ulterior au apărut mai multe versiuni: SmallTalk-72, SmallTalk-76, SmallTalk-80). Limbajele orientate pe obiecte sunt foarte modulare (grație autonomiei obiectelor), ele se apropie de cadre și posedă proprietățile de moștenire și de ierarhizare ale claselor. Modularitatea este asigurată și de ușurința introducerii și distrugerii claselor și a obiectelor. Aceste limbaje sunt lizibile ușor pentru că toate aspectele referitoare la o clasă sau la un obiect sunt regrupate izolat. Combinația reprezentării prin cadre și a limbajelor orientate pe obiecte a dat naștere unor limbaje destinate în special dezvoltării sistemelor expert (LOOPS, MERING, KOOL, LRO etc.). Anumite medii pentru dezvoltarea sistemelor expert sunt construite în jurul acestor limbaje (de exemplu Bull propune limbajul KOOL). Există și limbaje născute din extinderea limbajului LISP sau din combinarea acestuia cu limbaje orientate pe obiecte (FLAVORS, OBJVLISP, CEYX, ROSACE etc.). Au apărut chiar medii orientate pe obiecte realizate în C (precum C++, OBJECTIVE-C etc.).

Reprezentarea cunoștințelor într-un sistem expert trebuie să satisfacă în principiu două exigențe contradictorii. Prima exigență este legată de sistemul de calcul. Calculatoarele înțeleg un limbaj sărac și sunt cu atât mai eficace cu cât acest limbaj este mai apropiat de limbajul mașină (prima versiune a lui NEXPERT pe Macintosh era realizată în Assembly, acum este în C). A doua exigență este transparența (lizibilitatea) bazei de cunoștințe. O bază de cunoștințe trebuie să poată fi citită, completată, modificată, întreținută de către un expert în domeniul desemnat (în general neinformatician). Limbajul cel mai bun pentru a răspunde acestei cerințe ar fi probabil limba maternă a expertului, la care s-ar adăuga jargonul de specialitate. Problema este că, din păcate, calculatorul nu înțelege limbajul expertului, și invers. Reprezentarea cunoștințelor constă în găsirea unei terminologii intermediare, ceea ce nu poate fi decât un compromis (acesta explică faptul că, pe de o parte sistemele expert sunt numeroase, pe de alta ele nu sunt niciodată unanim aprobate...). Conform celor două exigențe majore amintite, există două tendințe.

Prima tendință este folosirea limbajelor de programare (de regulă independente de interpretor sau de calculator). Astfel, la scrierea unui program în PASCAL, se poate spune că programul va funcționa pe orice calculator și cu orice compilator PASCAL adaptat calculatorului (cu excepția câtorva aspecte legate în majoritate de intrări/ieșiri). În ceea ce privește limbajele, ordinea lor de preferință ar fi: LISP, PROLOG și limbaje orientate pe obiecte. Acestea însă sunt adaptate mai degrabă informaticienilor și calculatoarelor decât expertului uman. Cealaltă tendință este apelarea la instrumente de dezvoltare și generatoare de sisteme expert (instrumente special concepute pentru a scrie baze de cunoștințe). După cum indică și denumirea „instrument”, acestea nu au o destinație universală, spre deosebire de limbajele care permit exprimarea mai multor lucruri. Aceste pachete de programe (care într-o anumită măsură sunt ca niște limbaje limitate) sunt puternic dependente de interpretor (în cazul de față acesta este chiar mașina inferențială). Cele două tendințe tind să se reunească, în măsura în care limbajele oferă medii de dezvoltare, adică mai multe instrumente asociate, care furnizează aproape aceleași funcționalități ca și generatoarele de sisteme expert.

2.5 CÂTEVA ASPECTE LEGATE DE GENERAREA SISTEMELOR EXPERT

2.5.1 Mecanisme de inferență

Un sistem expert poate fi privit și ca un sistem de producții capabil să reproducă raționamentele unui expert într-un domeniu bine definit (eventual chiar în mai multe domenii bine definite). Ce face un expert? El pleacă de la o situație descrisă de un număr de fapte sau de date pe care le analizează și pentru care dă verdicte. În paranteză fiind zis, aceste verdicte nu trebuie înțelese ca și soluții, ele sunt de fapt rezultate firești ale raționamentelor expertului uman. Deci, plecând de la fapte, experții elaborează înlănțuiri de raționamente, raționamente care pot fi reprezentate sub forma unor reguli de producție de tipul:

Dacă condiția inițială este satisfăcută, **atunci** să se execute o acțiune.

Varietatea contextelor acoperite de acest formalism este foarte largă. Printre regulile de acest gen, un loc aparte este ocupat de *regulile deductive* (inferențe), care sunt de tipul:

Dacă premiza este adevărată, **atunci** concluzia este adevărată.

Regulile de producție de acest tip se află la baza majorității sistemelor expert. Ele fac parte din tipul general definit mai sus, în sensul în care condiția inițială este satisfăcută dacă premiza se găsește în baza de fapte adevărate (execuția regulei conduce la adăugarea concluziei în aceeași bază de fapte adevărate). Acest tip de regulă deductivă a fost folosit și de greci în antichitate și aparține domeniului logicii. Regulile gramaticale dintr-un limbaj indică cum pot fi construite corect frazele, plecând de la cuvinte. Regulile de producție care se întâlnesc în cadrul sistemelor expert sunt reguli de acțiune (ele permit fie să se acționeze, fie să se îmbogățească informația înainte de a se acționa).

Într-un sistem expert, ca de altfel în toate sistemele de producții, este necesar să existe o stare a bazei de fapte prin care să poată fi recunoscute regulile aplicabile, operație care este numită *filtrare* (pattern matching). Filtrarea este o operație destul de lungă, pentru că sistemele recurg în general la preselecția regulilor care urmează să fie filtrate. Altfel spus, în loc să fie selectate regulile aplicabile din mulțimea tuturor regulilor, se consideră o submulțime de reguli din care se selectează seturi aplicabile. Această operație de preselecție dinaintea filtrării se numește *restricționare*. După ce regulile aplicabile au fost recunoscute, trebuie aleasă o strategie de control (operație definită ca *selectare*). După ce regula a fost selectată ea trebuie să fie și executată. Modulul central al sistemului expert care realizează aceste operații (restricționarea, filtrarea, selectarea și executarea) se numește *motor de inferență* (*mecanism, mașină* etc.) sau *interpretor*.

Filtrarea constă în determinarea regulilor aplicabile potențial pentru o stare dată a bazei de cunoștințe. Sistemul de control poate decide ca o parte din reguli să nu fie testate la acest nivel (prin restricționare). Partea care trebuie testată dintr-o regulă se numește *declanșator*. Acesta este de regulă o formulă, alcătuită corespunzător în sensul logicii propozițiilor sau logicii de ordinul întâi, care se referă la fapte oarecare (conținând fapte și conectori). Sintaxa acestor formule este standardizată, iar numărul conectorilor acceptați este în mod obișnuit limitat.

Problemele de reprezentare ale formulelor logice joacă un rol important în programele pentru demonstrarea teoremelor. Motorul inferențial al unui sistem expert, bazat pe logica propozițiilor (*motor 0*), nu este capabil să testeze toate formele logice ale unei reguli, și majoritatea acestor mecanisme deductive nu gestionează negația. Adică acestea nu tratează faptele care nu dispun de informații referitoare la veridicitatea faptelor, ele se ocupă doar de faptele adevărate și nu efectuează deducții decât pentru cele adevărate. Deci, sistemele expert nu trebuie confundate cu sistemele destinate demonstrării teoremelor.

Motoarele bazate pe logica de ordinul întâi (a predicatelor) sunt mai avansate decât cele bazate pe logica propozițiilor. La acestea, ordinea de instanțiere a variabilelor în predicate poate să influențeze performanțele sistemului. Din această cauză se recomandă să se instanțieze variabilele care sunt singure într-un predicat, apoi acelea care sunt câte două, și așa mai departe. În practică problema se pune de regulă pentru cel mult două variabile, pentru că ponderea predicatelor este limitată la 2 (ceea ce –după opiniile specialiștilor– permite deja reprezentarea oricărui predicat și a oricărei funcții). Totuși, după părerea mai multor autori, la ora actuală nu există motoare inferențiale care să lucreze sub logica de ordinul întâi în totalitate, iar motoarele de acest gen, care acceptă variabile în fapte, sunt destul de rare.

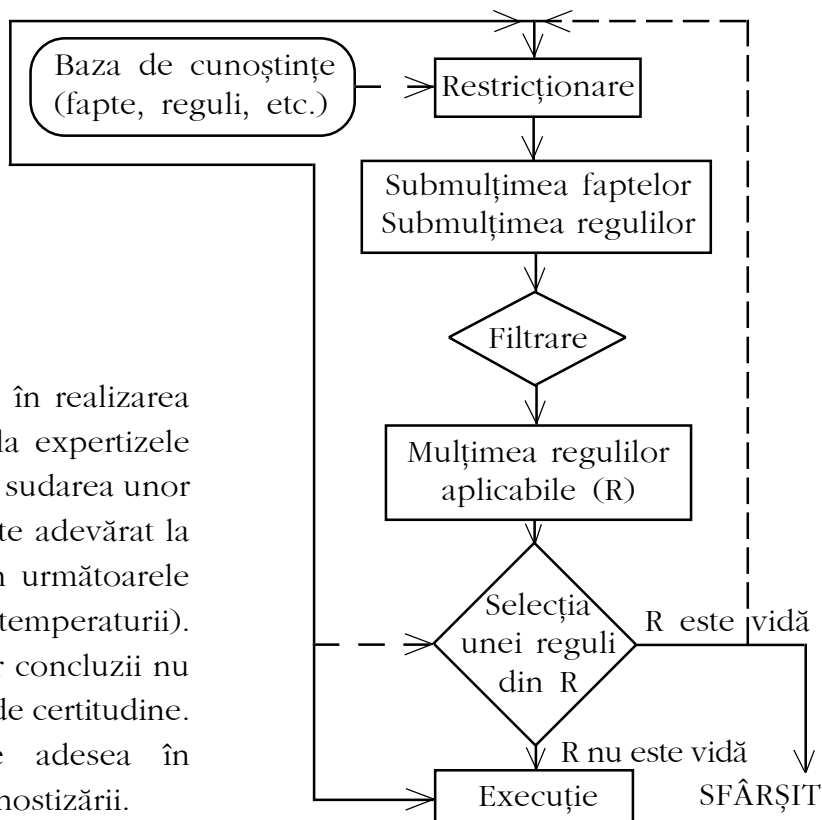
Un motor de inferență care funcționează după logica predicatelor, cu variabile în fapte și aplicând algoritmi de unificare generală (nu numai prin substituirea constantelor), ar putea fi numit *motor 1*. Un motor inferențial care nu are variabile în fapte și care permite numai unificarea prin constante a regulilor se poate numi *motor 0+*. Singurul limbaj care

se bazează în întregime pe calculul predicatelor este PROLOG. Acest limbaj dispune de un motor inferențial implicit, prin componentele pe care le are. Variabilele sunt acceptate în fapte, iar procesul de unificare a regulilor face parte din nucleul central al sistemului. Din alt punct de vedere, filtrarea este o operație greoaie în sistemele bazate pe logica de ordinul întâi, care poate conduce la o explozie combinatorică a numărului de teste. Pentru a ocoli acest impediment, se poate aplica o metodă „naturală” care constă din reducerea în amonte a numărului de reguli filtrate, prin aplicarea procedurii de restricție. La nivelul filtrării se poate ameliora procesul examinat (referitor la declanșatorul fiecărei reguli), dând prioritate predicatelor care au mai puține variabile libere. Un alt mod de a proceda este păstrarea în memorie a rezultatelor filtrărilor precedente pentru fiecare regulă sau pentru fiecare parte a declanșatorului (de exemplu: se pot reține unificările posibile pentru fiecare parte a declanșatorului). După fiecare ciclu al motorului de inferență se actualizează conținutul acestei zone de memorie. Această metodă de filtrare este denumită în literatura de specialitate *filtrare prin propagare*. Greutățile în cadrul operațiilor de filtrare apar mai ales în timpul execuției (când motorul de inferență este solicitat de către utilizator). De cele mai multe ori se poate reduce timpul de execuție prin realizarea unei precompilări/indexări a bazei de reguli (înainte de folosirea motorului inferențial). Această compilare prealabilă crează o rețea protejată (prin pointeri) de interdependențe între reguli. Principiul acestei rețele constă în crearea unor legături care, de exemplu, precizează ca o regulă a cărei concluzie conține „evaluarea proiectului” să nu fie filtrată atâta timp cât „timpul de revenire” nu a fost evaluat, stabilindu-se astfel ordinea de aplicare a seturilor de reguli. În timpul execuției, procesul poate fi accelerat dacă propagarea faptelor adevărate se efectuează pe baza acestei rețele pregătite. Această metodă de precompilare a regulilor a fost folosită prima dată pentru ameliorarea performanțelor sistemului OPS 5 (sub denumirea „algoritmul RETE”). În cazul prezentat mai sus, rețeaua creată rămâne invizibilă, ascunsă pentru utilizatori.

O altă alternativă constă în crearea unei rețele identice, dar încărcate cu semnificație. În acest caz rețeaua va fi denumită *rețea semantică*. Construirea rețelei se realizează prin analizarea cunoștințelor necesare pentru buna desfășurare a raționamentelor, acesta devenind factorul principal pentru accelerarea filtrării (un exemplu bun pentru o asemenea rețea semantică este furnizat de sistemul ARGUMENT).

Dacă, după faza de filtrare, există un număr de reguli recunoscute ca fiind aplicabile, structura de control va fi cea în măsură a stabili care dintre ele să se aplice (execute). Concluzia regulii (sau premiza în cazul sistemelor guvernate de scopuri) are ca efect modificarea bazei de cunoștințe prin crearea, modificarea sau ștergerea unor înregistrări (fapte). Bineînțeles, se pot imagina și sisteme în care concluzia unei reguli să modifice mulțimea regulilor pe care nu le-am analizat, însă un asemenea sistem nu ar fi un sistem de producții în accepțiunea clasică a definiției. Mai multe sisteme expert se rezumă, sub acest aspect, doar la adăugarea de fapte în baza de fapte. Aceste sisteme sunt denumite *monotone*. Pe drept cuvânt, denumirea de monotonie presupune că faptele care se adaugă nu sunt contradictorii cu cele care se găsesc deja în baza de fapte. Altfel spus, aceste sisteme nu țin seama de loialitatea unui fapt recunoscut în prealabil ca adevărat. Această

Fig.3 Funcționarea unui sistem cu restricție și selecție



limită este de altfel un obstacol în realizarea bazelor de cunoștințe relative, la expertizele variabile în timp (de exemplu, la sudarea unor profile laminate, un fapt care este adevărat la un moment dat, poate fi fals în următoarele momente datorită schimbării temperaturii). Uneori se folosesc reguli a căror concluzii nu sunt adevărate decât cu un grad de certitudine. Această situație se întâlnește adesea în sistemele expert destinate diagnostizării.

În general se poate afirma că, într-un sistem de producții, strategiile de control permit trecerea unei baze de fapte într-alta (sistemul trece dintr-o stare în alta). Fiecărei stări a bazei de fapte îi corespunde o submulțime de reguli de producție bine determinate (*reguli aplicabile*). Pentru a reduce neajunsul referitor la numărul de teste de efectuat, se poate lua decizia ca numai o parte din reguli, sau fapte, să fie testate. Deci, înaintea operației de filtrare este necesară o fază de restricționare. Sistemul va funcționa, în acest caz, conform schemei reprezentate (figura 3). Modulul de selecție trebuie să dispună de capacitate de acțiune și în cazul în care mulțimea regulilor aplicabile este vidă (dacă restricția este prea puternică, se poate încerca din nou cu alta mai puțin restrictivă). După execuție se poate relua o regulă din mulțimea R, până la epuizarea tuturor elementelor din mulțime, sau se poate reface întreg ciclul. Trebuie să facem distincție între cele două faze de alegere care intervin: faza de alegere dinaintea filtrării (restricția) și faza de alegere de după filtrare (selecția). Faza de selecție se poate numi și *rezolvarea conflictului*, în acest caz mulțimea regulilor aplicabile la acest nivel numindu-se *mulțimea conflictelor*. Strategiile de control aplicabile acestor două faze pot fi diferite. Ansamblul operațiilor ilustrate prin schema din figura 3 formează ciclul unui motor de inferență.

Pentru a reduce mulțimea de fapte sau de reguli prin operația de filtrare, au fost propuse mai multe modalități. Într-un sistem guvernat de fapte, restricționarea constă din luarea în considerare a faptelor adevărate. Filtrarea constă din reținerea tuturor regulilor a căror premiză este adevărată, ținând seama de faptele (adevărate) prezentate în baza de fapte. Se spune că un astfel de sistem funcționează prin înlănțuire înainte (*forward chaining*). Dacă sistemul este guvernat de scopuri, restricția constă din luarea în considerare doar a scopurilor (acestea pot fi privite și ca fapte ce trebuie stabilite). Filtrarea constă deci din reținerea regulilor a căror concluzii conțin scopuri care se găsesc în baza de fapte.

Se spune că un astfel de sistem funcționează prin înlănțuire înapoi (*backward chaining*). Dacă sistemul este guvernat de fapte și scopuri se poate spune că sistemul are înlănțuire mixtă (*mixed chaining*).

Restricția poate fi comandată și explicit, de către expert, cu ajutorul regulilor (este vorba în acest caz de *metareguli*, adică reguli aplicabile regulilor). O metaregulă poate indica starea următoare a bazei de cunoștințe/fapte prin setul regulilor de considerat pentru filtrare. Metaregulele pot indica grupele de reguli prioritare sau pot defini o ordonare a submulțimii regulilor. Restricționarea poate respecta și legi sau reguli care nu sunt comandate prin metareguli ci care rezultă din structurarea problemei. În acest caz se spune că există o *lege de precedentă*. Acesta înseamnă că regulile pot fi organizate într-un *graf de precedentă*, care ilustrează ordinea de parcurgere (și implicit de aplicare) a regulilor.

Cunoștințele pot fi organizate și sub forma unei rețele analoage rețelelor semantice utilizate în analiza sintactică a limbajelor. Acest mod de organizare este eficient atunci când se poate constitui cu adevărat o organizare ierarhizată a cunoștințelor. Un alt mod de a comanda restricționarea poate fi prin organizarea regulilor în pachete sau *scheme*. Plecând de la această organizare se poate construi o structură arborescentă din reguli. Toate aceste metode depind însă, în primul rând, de felul reprezentării adoptate pentru cunoștințe.

După faza de restricționare urmează filtrarea, putând ajunge în situația în care sunt aplicabile mai multe reguli. Este necesară efectuarea unei alegeri din mulțimea acestor reguli. După ce regula găsită a fost executată, programul trebuie să știe unde să se întoarcă. În cazul în care există cel puțin o regulă aplicabilă (mulțimea rezultată din filtrare nu este vidă), există două strategii posibile: O explorare în *lățime înainte*, care constă din considerarea unei reguli aplicabile (de obicei se ia prima disponibilă) care se execută. Altfel spus, se execută succesiv toate regulile aplicabile, înainte de a testa dacă condiția de oprire este satisfăcută (dacă nu, se trece la un nou ciclu al motorului de inferență). Acest tip de *explorare exhaustivă* este folosită și în sistemul Mycin. O altă strategie este găsirea unei reguli aplicabile urmând o metodă euristică, după care se trece la execuție (dar execuția acestei reguli nu modifică baza de fapte). Apoi, se verifică dacă condiția de oprire este satisfăcută. Dacă nu este satisfăcută, se trece la un nou ciclu al motorului de inferență. Cea de a doua strategie se cheamă explorare prin *adâncime înainte*. Alegerea la acest nivel se supune unor reguli, ca la faza de restricționare. Euristică nu se bazează neapărat pe o funcție de evaluare, ea poate fi dirijată prin diferite metode.

Dacă nici o regulă nu este aplicabilă (mulțimea rezultată din filtrare este vidă), atunci majoritatea sistemelor adresează întrebări utilizatorului, iar dacă utilizatorul nu știe sau nu dorește să răspundă, sistemul se poate opri. Se poate deci lua în considerare faza de restricționare și fără schimbarea ciclului motorului inferențial.

2.5.2 Medii și generatoare pentru sisteme expert

În paragrafele precedente am descris deja principiile de funcționare ale sistemelor expert.

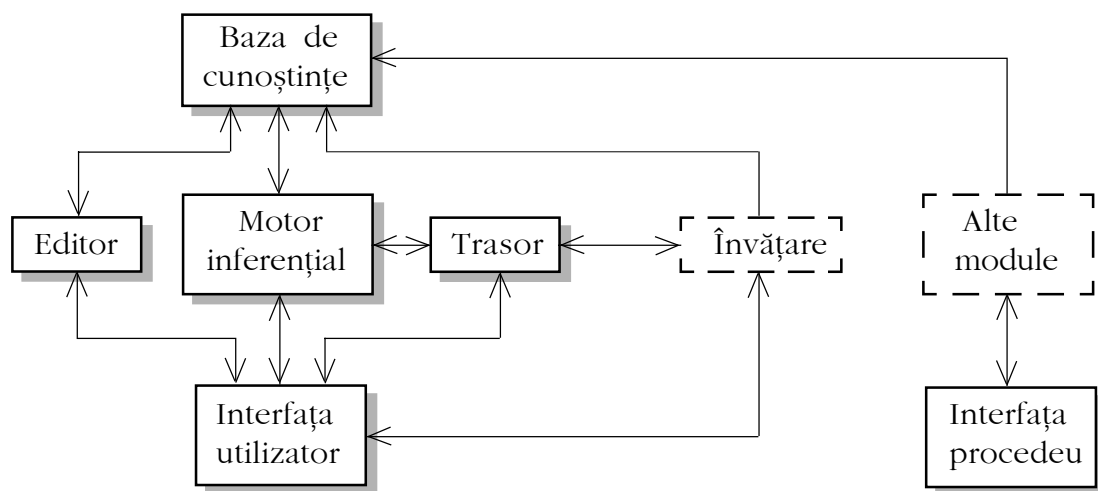


Fig.4 Arhitectura simplă a unui instrument de elaborare (GSE)

Dar un sistem expert —oricât ar fi de performant— nu este numai un motor de inferență, ci necesită mai ales o acumulare masivă de cunoștințe care trebuie introduse și gestionate în sistem. De asemenea trebuie interpretate și înțelese rezultatele oferite de către sistem. Toate aceste funcțiuni se găsesc în afara atribuțiilor și posibilităților motorului inferențial, ele fiind realizate cu instrumente auxiliare numite *utilitare*. Noțiunea de sistem expert se poate întâlni frecvent și referitor la aceste instrumente (care sunt pachete de programe fără baze de cunoștințe dar care sunt capabile să primească și să exploateze cunoștințe organizate într-o bază). Aceste instrumente se pot numi mai potrivit *generatoare de sisteme expert* (GSE) sau medii (din traducerea termenului englez: *shell*). Un asemenea generator conține, în afara motorului de inferență, utilitare de elaborare și de exploatare necesare.

În figura 4 este ilustrată arhitectura generală a unui GSE (săgețile indică sensurile fluxurilor de informații). În centru se află motorul de inferență care este conectat la baza de cunoștințe, de unde poate extrage informațiile, revenind apoi și, eventual, modificând datele. Baza de cunoștințe trebuie încărcată și alimentată cu fapte și reguli, această operație fiind realizată cu ajutorul editorului (toate sistemele evolute posedă un asemenea editor pentru achiziția cunoștințelor). Alături de acest modul indispensabil există și alte utilitare: trasorul, motorul de elaborare, modulul de învățare și/sau autoinstruire, eventual alte module profilate pe explicații, comentarii și calcule. Cele din urmă nu sunt prezente în toate sistemele expert, ele putând fi înlocuite de interfețe sau funcții specializate, după natura domeniului.

Figura prezintă doar un singur modul de interfață, însă fiecare modul poate fi echipat cu propria interfață de dialog. Pentru sistemele expert industriale, de regulă trebuie să existe și o interfață-procedeu pentru achiziționarea automată a datelor de la captatoare. În domeniul gestiunii, interfețele sunt destinate mai ales pentru extragerea datelor din baze și înregistrarea noilor informații. De asemenea, în această schemă figurează un singur motor de inferență, dar în prezent există tendințe spre arhitecturi cu mai multe motoare. Multiplicarea motoarelor rezultă fie din existența motoarelor separate pentru reguli și metaregulă, fie din faptul că sistemele pot fi de tipul multi-expert (destinate pentru mai multe domenii de cunoaștere date).

Editorul, prin interfața care asigură și dialogul, trebuie să permită schimburi de cunoștințe (comunicare) într-un mod cât mai accesibil utilizatorilor potențiali (de exemplu: în mod text cât mai apropiat de limbajul natural, în mod grafic utilizând semne și simboluri cât mai sugestive). Pentru ca utilizatorul să se facă înțeles de către calculator, este necesar să se familiarizeze și să se conformeze cu o sintaxă predefinită a comunicării. Voi încerca să prezint în continuare ceea ce poate realiza un editor "clasic" (în mod text) în cadrul unui *GSE*, pentru facilitarea utilizării sistemului:

- Poate ușura înțelegerea afișând pe ecran câmpuri ce trebuie completate (dacă premisa sau concluzia unei reguli conține un obiect cu valoare atașată, poate să propună lista valorilor posibile sau valoarea implicită).
- Poate afișa lista obiectelor selectate pentru a permite crearea sau actualizarea rețelei semantice.
- Poate controla încadrarea corectă a unei reguli într-o rețea de inferență.
- Poate afișa regulile care au aceleași premise sau concluzii, permițând reperarea eventualelor incoerențe sau contradicții.
- Poate verifica corectitudinea (ortografia, recunoașterea existenței) cuvintelor cheie, obiectelor sau cadrelor utilizate în reguli.
- Poate verifica dacă utilizarea conectoarelor este corectă.
- Poate afișa rubricile unui cadru cu procedurile asignate sau să afișeze legăturile unui cadru.

În măsura în care structura cunoștințelor permite, editorul poate verifica dacă faptele și regulile au forma cerută. De exemplu, în modulul de achiziție a cunoștințelor la sistemul Mycin, editorul verifică dacă o regulă referitoare la identitatea unui organism are în premise următoarele trei elemente: tipul infecției, locul culturii și modul de contaminare. Sistemele care posedă o structură semantică de genul rețea de cadre sau clase pot verifica coerența între rețea și datele furnizate de către utilizator. Dacă utilizatorul dorește să adauge obiecte sau reguli, editorul îl va chestiona referitor la modul de integrare al acestora în rețelele existente. Deci editorul poate controla dacă cunoștințele introduse corespund structurării predefinite a cunoștințelor. În schimb, el nu este capabil să detecteze toate tipurile de incoerențe. O mare parte dintre contradicții se poate elimina în faza de testare, în momentul manifestării acestora. Cercetarea aprofundată a incoerențelor ar putea fi rezolvată, eventual, de un modul special – o asemenea funcție nu este obligatoriu să fie o performanță oferită de către editor.

Editorul gestionează și dicționarul faptelor, obiectelor sau al cadrelor, al regulilor (posedând lista completă a acestora) și confidențialitatea acceselor la bazele de cunoștințe. De multe ori poate fi utilă și asocierea unor comentarii la unele fapte și reguli, realizarea și ordonarea acestora fiind asigurată tot prin modulul de editare. În anumite sisteme, editorul își încheie sesiunea prin compilarea regulilor, ceea ce permite accelerarea activității inferențiale a motorului, și detectarea erorilor de sintaxă din cadrul regulilor. Interfața editoarelor este de multe ori impresionantă, ca urmare a efortului imens depus de către proiectanții acestor

sisteme. Pot fi folosite culori, meniuri, ferestre multiple, există butoane active (pentru mouse sau ecrane tactile), mai mult decât în alte module.

Modul normal de funcționare al sistemului este cel de exploatare. Pe măsura derulării sesiunii de lucru, motorul inferențial va solicita (prin intermediul interfeței sale) utilizatorului faptele necesare care pot fi utile pentru elaborarea raționamentului și completarea bazei de cunoștințe inițiale. Utilizatorul, de regulă, nu se mulțumește să răspundă la câteva întrebări asupra naturii faptelor, el dorește în general să urmărească și raționamentul motorului. Din această cauză, există un modul special, conceput pentru a răspunde la motivul adresării unei întrebări sau la modul în care a ajuns sistemul la o anumită concluzie: *trasorul*. Pentru a utiliza cu încredere un sistem expert, trebuie să fim siguri că acesta este adecvat și de bună calitate. Din această cauză, sistemele, înainte de a trece în faza de exploatare, trec printr-o fază de dezvoltare și implementare. Modul de dezvoltare reprezintă deci un mod de a utiliza sistemul, care permite îmbogățirea și validarea acestuia. Modulul folosit pentru dezvoltare este de obicei tot *trasorul*. Acesta poate fi ajutat și de către proceduri sau module care nu servesc în cazul exploatării normale (pentru a desemna ansamblul acestor instrumente, se utilizează expresia: *motor de elaborare*).

Trasorul este deci un element esențial al sistemului expert, care joacă un rol important atât în modul de exploatare, cât și în etapele de dezvoltare. Prima lui sarcină este urmărirea șirului raționamentelor care se efectuează prin motorul de inferență. Trasorul are, de asemenea, sarcina de a actualiza baza de fapte, pe măsură ce faptele sunt dovedite. În cadrul modului de exploatare trebuie să fie posibilă citirea faptelor care sunt stabilite la un moment dat al sesiunii. Putem dori modificarea acestor fapte –la care trebuie să avem acces în cadrul sesiunii– mai ales dacă avem impresia că ne-am înșelat la un răspuns anterior. Când modificăm valoarea unui fapt, trasorul regăsește toate deducțiile care au fost făcute pornind de la acest fapt și le șterge. Sesiunea se poate relua cu noua valoare (și implicit cu noua stare). Asemenea încercări de modificare permit evaluarea sensibilității rezultatului final raportat la valoarea faptelor sigure (certe). Această funcțiune, numită *studiu de sensibilitate*, este una din posibilitățile importante oferite utilizatorului de către generatoarele performante. Orice modificare a faptelor antrenează actualizarea imediată a bazei de fapte și continuarea consultării cu noile date. La sfârșitul sesiunii este disponibil ansamblul regulilor care au servit la demonstrarea unui scop, adică este accesibil traseul complet, precum și înlănțuirea raționamentelor.

2.5.3 Motorul de elaborare și învățarea

După cum am amintit, această noțiune acoperă proceduri asociate trasorului și care se folosesc de regulă numai în modul de dezvoltare. Motorul coordonează activitatea trasorului și a editorului, permițând facilitarea dezvoltării. Această dezvoltare înseamnă printre altele îmbogățirea bazei de reguli (prin adăugări), validarea regulilor (verificarea corectitudinii raționamentelor). Într-un motor cu coeficienți de verosimilitate trebuie realizată și ajustarea coeficienților. Calitatea unei baze de reguli poate fi măsurată prin considerarea următoarelor criterii: completitudinea, coerența și –în sistemele cu coeficienți de certitudine–

neredundanța. Completitudinea înseamnă că sistemul nu riscă să se oprească datorită inexistenței unor reguli în baza de cunoștințe. Ea este foarte importantă la sistemele industriale cu buclă închisă, altfel fiind necesară intervenția utilizatorului. Coerența reprezintă proprietatea unei baze de reguli de a produce fapte necontradictorii, pentru orice bază de fapte necontradictorii. Trebuie să menționez însă că nu am găsit informații despre metode informatice care să permită verificarea coerenței unei baze de reguli. Nu există un moment precis pentru testarea coerenței, coerența putând să apară după executarea unui număr nedeterminat de reguli. Modalitatea cea mai sigură de a scrie o bază coerentă de reguli constă, în primul rând, în structurarea regulilor în rețele și pachete, iar în al doilea rând, în testarea variată și semnificativă a acestora cu baza de fapte. Atunci când un expert introduce o regulă nouă, el poate apela la noțiuni care nu figurează încă printre cunoștințele sistemului. În asemenea cazuri motorul de elaborare trebuie să recunoască noul concept, să-l plaseze printre cadre sau scheme, eventual să-l potrivească într-o rețea. Pentru acesta trebuie generat un dialog corespunzător între expert și sistem, astfel încât să se evalueze și să se găsească locul noului concept. Aceasta este una din căile pentru evitarea incoerențelor grosiere.

Există două noțiuni de învățare care sunt asociate sistemelor expert: achiziționarea noilor reguli și reperarea celor euristice performante. Pentru reguli, problema se poate pune în funcție de scopul urmărit. Putem încerca să adăugăm reguli într-o structură de cunoștințe deja stabilită sau putem să dorim achiziționarea regulilor prin niște exemple (inducție). În primul caz ar fi vorba mai ales de îmbogățire, fapt ce ține de motorul de inferență. În al doilea caz, pentru exemplele bine structurate, se regăsește noțiunea de *tabelă de decizie* (o tehnică răspândită deja în anii șaizeci, redevenită modernă odată cu sistemele expert). O asemenea tabelă poate fi interpretată și ca o matrice de exemple sau de situații. În sistemele obișnuite (cu inducție) se folosesc asemenea tabele pentru generarea ansamblelor de reguli care să conducă la aceleași concluzii ca și matricea, pentru situațiile pe care aceasta le conține.

Există și alte tehnici, mai complicate, dar care se bazează pe structurarea avansată a tipurilor de reguli. Anumite sisteme sunt capabile să înlocuiască într-o regulă o constantă cu o variabilă sau să abandoneze o parte a premisei pentru obținerea unei reguli mai puternice (generalizare). Învățarea poate să se refere și la structura de control. Aceasta înseamnă posibilitatea asocierii unor legi euristice performante sau a unor metareguli stărilor din baza de fapte, după ce sistemul a constatat că, conjuncția acestor stări și a acestor euristici ar putea conduce în mod rapid la rezultat.

Merită menționat faptul că în acest domeniu există probabil perspectivele cele mai promițătoare. Spre sfârșitul anului 1998 a fost lansat pe piață jocul "Creatures" care are ca și subiect principal crearea și urmărirea evoluării unor ființe virtuale (creaturi) care pot achiziționa și utiliza un bagaj limitat de cunoștințe.

Totuși, sistemele de învățare sunt considerate în general încă obiecte de laborator, achiziționarea cunoștințelor făcându-se de obicei fără învățare propriu zisă. De altfel, dacă

ne referim la om, ar fi de domeniul hazardului să vorbim despre învățare fără *efect retroactiv*. Ca să existe acest efect, sistemul ar trebui să știe când a raționat în mod corect (ceea ce ar reprezenta un pas important în direcția conștientizării), dar așa ceva deocamdată înseamnă să-i cerem prea mult. Pentru moment, când un expert este în dezacord cu concluzia unui sistem, nu se pune problema ca sistemul să schimbe în mod automat unele reguli, cu atât mai puțin strategia sa de control. Anumite programe însă dispun de capacitatea de a utiliza manevre pentru probleme test, pentru a comunica de câte ori a fost implicată o anumită regulă în obținerea rezultatelor eronate în sesiunile de lucru (acesta permite reperarea celor mai discutabile reguli). Este cert că, învățarea și programele pentru achiziționarea cunoștințelor reprezintă o cale promițătoare pentru dezvoltarea sistemelor expert. Deocamdată, însă, se consideră că este mai ușoară introducerea cunoștințelor cu mâna (sub forma unor reguli) decât construirea unui modul de învățare eficace.

CAP.3 SISTEME EXPERT ÎN CONSTRUCȚII

3.1 STRUCTURA ȘI DIRECȚIILE DE DEZVOLTARE ALE SISTEMELOR EXPERT

Sistemele expert înmagazinează cunoștințe specifice prin datele și programele pe care le conțin. Această bază de cunoștințe a sistemului expert este compusă frecvent, pe de o parte din supoziții (posibilități) specificate corespunzător, pe de altă parte din mulțimea cauzelor, exprimate de obicei ca reguli, legate de combinațiile acestor posibilități. Din punctul de vedere al metodei, aceasta se realizează cel mai uzual prin înlănțuirea relațiilor de tipul "dacă...atunci..." referitoare la domeniul predefinit al posibilităților. Reprezentarea de cunoștințe bazată pe reguli nu este însă singura posibilitate. Există sisteme, chiar dacă într-un număr mai redus, care funcționează prin rețele semantice, cadre etc.

Stabilirea și definirea regulilor și a consecințelor este sarcina specialistului din domeniul abordat. Realizarea mecanismului necesar pentru structurarea și gestionarea cunoștințelor, precum și a regulilor, cade în sarcina specialistului din domeniul inteligenței artificiale. Această activitate este denumită generic "ingineria cunoașterii" - după opinia mai multor autori nu reprezintă altceva decât analiză de sistem la un nivel elevat... Cunoștințele de specialitate, ce completează datele în cadrul sistemului, se regăsesc sub denumirea de *bază de cunoștințe*. Metaregulile, împreună cu relațiile și regulile asupra cărora acționează, fac parte tot din această mulțime, în literatura de specialitate fiind de multe ori referite separat ca *bază de reguli*. Complexitatea sistemelor crește odată cu numărul regulilor. Dacă la început au existat sisteme cu cca. 2.000 de reguli, care au fost considerate complexe, azi se pot întâlni sisteme care conțin peste 20.000 de reguli. Dar nu numai numărul de reguli prezintă relevanță în aprecierea unui sistem. Arhitectura interioară a sistemelor expert poate fi și ea destul de complicată. După Sloman [172] componentele care trebuie avute în vedere la aprecierea unui sistem expert sunt următoarele:

- Structuri independente, cum ar fi:
 - ❶ mediul;
 - ❷ biblioteca conținând date factice, relații sigure (dependențe) și posibilități (opțiuni);
 - ❸ biblioteca ce conține resurse (fapte, mijloace, unelte, limbaje, sisteme de simboluri, procedee învățate anterior, programe);
 - ❹ catalogul resurselor (ce contribuie la "autocunoașterea" sistemului);
 - ❺ așa zisa bibliotecă de motivații;
 - ❻ indexul procedeu-scop.
- Proceduri cu caracter permanent pentru utilizarea inteligentă a cunoștințelor existente și a informațiilor noi achiziționate, cum ar fi:
 - ❶ proceduri pentru realizarea gestionării centrale;
 - ❷ diverse monitoare;
 - ❸ proceduri de analiză retrospectivă (prin înlănțuire înapoi: *backtracking*)

pentru supervizarea presupunerilor, propunerilor, procedurilor, proiectelor anterioare.

Sistemele expert care dispun de structuri simple, la o privire mai superficială, se aseamănă foarte mult cu tabelele decizionale. Există însă câteva note de discordanță semnificative. Față de o tabelă de decizie, un sistem expert poate:

- da explicații, poate motiva recomandarea sa;
- învăța;
- oferi nu numai procedee deterministe ci și euristice (acolo unde asemenea procedee sunt necesare sau pertinente), tratează probleme cu probabilități cunoscute (asumate) sau incerte (vagi);
- depășind simplele relații de tipul "dacă... atunci...", poate rafina relații apelând și la cadre, rețele semantice, logică și mulțimi vagi –în acest ultim caz cu ambele sensuri ale incertitudinii: fie a percepției, fie a semnificației.

În privința specificării resurselor materiale ale sistemelor expert, în tendințele mondiale se puteau întâlni două curente, reprezentând interese oarecum contradictorii. Concepția japoneză despre dezvoltarea sistemelor expert în anii '90 prevedea dominația sistemelor bazate pe reprezentarea cunoștințelor. Detașându-se impetuos de principiile arhitecturii clasice de tip Neumann, dezvoltarea *hardwer*-ului considerat ca a V-a generație de echipamente a pornit cu avânt. Bazat în principal pe tehnologia *VLSI* cu interfețe om-mașină inteligente, respectiv realizarea și pregătirea apariției pe piață a calculatoarelor deductive-rezolvitoare de probleme și utilizatoare de baze gnoseologice, a necesitat eforturi financiare și științifice deosebite. Fantezia dezvoltatorilor americani și a celor vest-europeni s-a lovit mai tare de problema uzurii morale a investițiilor masive (la nivel de miliarde de dolari) existente în calculatoare și echipamente tradiționale. Din această cauză concepțiile lor sunt caracterizate mai mult prin strădania menținerii compatibilității sistemelor noi cu calculatoarele anterioare, aparținând generației a IV-a. Cu toate acestea, distanța dintre cele două concepții se micșorează din ce în ce mai mult. Există și în cadrul dezvoltărilor americane și europene echipamente speciale dedicate (de exemplu: calculatoare LISP, PROLOG). De fapt, după cum remarcă un specialist competent al tematicii (T. Vámos [187]), din cele patru componente principale necesare pentru realizarea unui sistem expert:

- ❶ determinarea problemei,
- ❷ crearea bazei de cunoștințe,
- ❸ realizarea mecanismului deductiv (sistemul de reguli),
- ❹ alegerea uneltelor și a mijloacelor matematice și de programare, realizarea unui sistem "prietenos" pe cât posibil în timp real,

doar ultima, a patra, se leagă strict de calculatoare. Pentru realizarea bazelor de cunoștințe și a sistemelor expert există din ce în ce mai multe unelte, accesorii și medii de lucru, care permit elaborarea și întreținerea facilă. Cererea pe piață pentru aceste produse este în

creștere, nu în ultimul rând pentru că au un caracter independent: pot fi separate de baza de cunoștințe a oricărui sistem expert concret, fiind astfel posibilă utilizarea lor, chiar repetată, în proiecte diferite. Există și componente sau subansamble specializate, cum ar fi *mașinile deductive* (motoarele de inferență) cu care viteza obținerii deducțiilor este accelerată de 10-50 de ori prin înglobarea unor soluții hardware. Pentru menținerea utilajelor tradiționale se pot imagina și soluții de completare cu agregate de inteligență artificială ce se pot cupla în funcție de tipul problemei abordate.

Cercetările inteligenței artificiale au condus la recunoașterea faptului că influențele și dependențele reciproce dintre fenomenele sociale, economice, chiar psihologice, sunt mult mai complicate decât cele întâlnite în legile științelor naturii. Pentru tratarea lor nu dispunem de utilitate de valoare a celor oferite de analiza matematică pentru descrierea și analiza fenomenelor fizice. De aici derivă faptul că aproape toate sistemele expert realizate până în anii '90 au fost create în domenii tehnico-naturo-științifice, deoarece în problemele din cadrul acestor domenii înțelegerea între specialiștii de marcă, referitoare la existența soluțiilor considerate corecte, se realizase relativ ușor, majoritatea acestora putând fi fundamentată pe legile cunoscute din științele naturii, pe relații de cauzalitate. În situațiile de decizie și în modelele din sfera economică respectiv umană, incertitudinea este considerabilă, cunoștințele sunt incomplete. Există mai multe posibilități de comportament și de acțiune pentru alegeri considerate corecte sau bune. În cele mai multe cazuri specialiștii acestor domenii ajung foarte greu la consens în privința considerării priorității regulilor. În ciuda acestor probleme, s-a reușit crearea și a unor sisteme expert, deja consacrate, pentru conducere [138]. Cercetările teoretice ale inteligenței artificiale și ale ramurilor științifice adiacente ne apropie de înțelegerea mai profundă și definirea productivă a multor fenomene fiziologice, sufletești, respectiv sociale. Dintre aceste cercetări putem remarca următoarele:

- definirea unor logici noi (logica temporală, logica modală, logica de acțiune etc.),
- unele direcții ale cercetărilor lingvistice (semanticile generative și interpretative derivate din gramaticile de tipul Chomsky, limbajul logic al lui Montague și Kripke [70]),
- studiul proceselor cognitive: percepția, înțelegerea, gândirea, deducția pentru o înțelegere mai profundă (aici se poate încadra și domeniul proceselor dezvoltării sufletești, învățarea, predarea).

În oglinda noilor descoperiri și știința de calcul primește o înțelegere mai largă. Ea străpunge granițele teoriei sistemelor tradiționale și ale ciberneticii, care se ocupau cu procese caracterizabile prin indicatori cantitativi și structuri relativ neschimbate. Au apărut astfel pe lângă aparatul matematic (singular până atunci), procesele și noțiunile mai puțin formale, nenumărate (analiza sentențială, interpretarea, recursia reciprocă, efectele colaterale, potrivirea texturilor etc.), deoarece de cele mai multe ori relațiile dintre procesele și elementele care compun fenomenele multiplu compuse amintite, nu se încadrează într-o ordine ierarhică. Se poate întâmpla ca procesul *A* să facă parte din procesul *B*, iar

procesul *B* să facă parte din procesul *A*. Cercetarea acelor mijloace mentale, care fac posibilă gândirea despre procese multiplu înlănțuite cu influențe reciproce, nu a pornit de mult timp. Este evident faptul că aparatele matematice ale fizicii nu pot înlocui aceste mijloace noi, ba mai mult, folosirea lor poate conduce la obținerea unor reprezentări nereușite sau înșelătoare ale unor asemenea domenii. Limitele teoriilor matematico-informaționale, dezvoltate de către Shannon și Weaver în domeniul problemelor amintite, s-au conturat în mod asemănător. După cum remarcă A. Sloman [172]: "comunicarea umană este total diferită: noi nu decodificăm ci interpretăm, folosind pentru acesta foarte-foarte multe cunoștințe de fundal, precum și capacitate de rezolvare". După părerea unui specialist de vârf din Europa de vest [6] sistemele expert pot juca rolul tehnicii rachetelor din anii '60. Ca și atunci, și în acest caz se cumulează o gamă largă a unor tehnologii noi într-o aplicație specifică. Prin această caracteristică, asemenea aplicații pot constitui motorul de dezvoltare al întregii informaticii.

3.2 SISTEME EXPERT, MEDII DE DEZVOLTARE PE MICROCALCULATOARE

Sistemele de software destinate realizării sistemelor expert sunt unelte de cele mai multe ori independente de problema urmărită, care folosesc diverse mecanisme deductive și scheme de reprezentare prestabilite pentru cunoștințe. Funcționează deja destul de multe asemenea sisteme, deasemenea există multe în curs de dezvoltare sau de perfecționare. Acestea sunt destinate unei game largi de microcalculatoare, începând cu cele mai modeste până la cele mai performante. Există sisteme care necesită doar 64kb memorie și procesor din generația 8086 (puterea acestora se poate extinde prin mărirea memoriei accesibile), precum și sisteme care nu se descurcă fără calitățile multimedia ale calculatoarelor personale de azi.

În cele ce urmează voi prezenta sumar caracteristicile unor sisteme mai cunoscute, devenite deja clasice (fiind elaborate între anii '70 - '90), amintind și domeniile lor efective de aplicare. Pentru realizarea acestei liste, am pornit de la articolul lui Paul E. Lehner și Stephen W. Barth, intitulat: Expert Systems on Microcomputers (publicat în revista Expert Systems Nr. 4, vol. 2, octombrie 1985), ajungând până la multe resurse actuale accesibile prin internet.

AL/X (Advice Language) [153; 161] a fost printre primele sisteme expert mai importante care au rulat pe microcalculatoare. Unitatea de inferență, limbajul pentru descrierea regulilor, precum și sistemul consultativ s-au elaborat după modelele oferite de PROSPECTOR [59] și EMYCIN [188]. Sistemul este destinat găsirii și tratării erorilor la sondele de extracție de pe câmpurile petrolifere, dar el poate fi aplicat și pentru orice altă problemă a cărei mulțime de reguli se poate defini prin limbajul descriptiv al sistemului AL/X. Acestea sunt precizate sub forma unei rețele deductive (figura 5), unde pw și nw reprezintă măsurile Bayes (pozitive și negative) asociate încrederii în faptul elementar E5. Ipotezele finale (aflate la nivelul cel mai înalt) pot fi reduse înapoi la ipoteze parțiale diferite, care la rândul lor pot fi descompuse mai departe până la faptele elementare aflate pe nivelul de bază –așa numitele evidențe. Săgețile marchează legăturile de tip condiție-

consecință cuprinse în reguli. Ca și exemplificare, cele din E1 și E2 înspre A1 corespund regulii următoare:

DACĂ E1 ȘI E2 ATUNCI cu o anumită certitudine A1.

Fiecărui vârf al rețelei îi aparține un parametru inițial de încredere (prior degree of belief) care se precizează dinainte de către utilizator prin evaluarea faptului reprezentat de acel vârf. Certitudinea condițiilor se calculează după reguli stabilite (ilustrate în figura 6).

În etapa consultativă sistemul evaluează certitudinea ipotezelor finale, urmărindu-le înapoi la certitudinile faptelor elementare ale căror valori le află din răspunsurile utilizatorului. Această etapă se compune în principal din răspunsuri date la întrebările furnizate de către sistem, eventual din primirea unor explicații referitoare la procesul deductiv. În manualul de utilizare [153] se poate găsi tratarea amănunțită a unui exemplu.

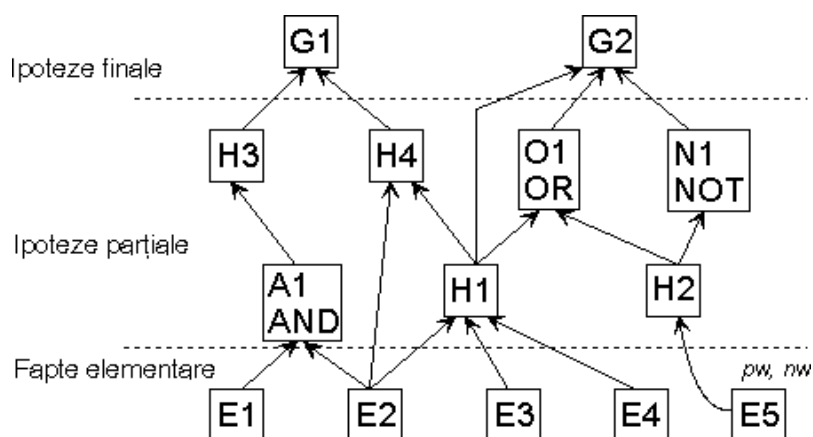


Fig.5 Structura unei rețele deductive în AL/X

Regula	Calculul certitudinii
AND	$\text{Degree}[A1] = \min[\text{Deg}[E1], \text{Deg}[E2]]$
OR	$\text{Degree}[O1] = \max[\text{Deg}[H1], \text{Deg}[H2]]$
NOT	$\text{Degree}[N1] = -\text{Deg}[H2]$
Bayes	$\text{Degree}[H2] = W + \text{Deg}[E5]$, unde W și $\text{Deg}[E5]$ rezultă din interpolarea încărcărilor pozitive pw și negative nw între $E5$ și $H2$.

Fig.6 Regulele din AL/X pentru calculul certitudinii consecințelor

versiuni și pentru sisteme Apple II respectiv LSI-11/02 în UCSD Pascal; pentru Cromemco Superbrain în Pascal/M și pentru IBM PC. În cazul unei memorii de 64kb, AL/X într-o etapă consultativă poate trata aproape 100 de reguli (pentru a rula cu asemenea resurse minime, bazele de cunoștințe mai voluminoase –după o descompunere adecvată– pot fi înmagazinate în mai multe pachete, fragmentând astfel și etapa consultativă).

ERS (Embedded Rule-based System) [29] permite nu numai realizarea unui sistem consultativ simplu, dar și a unui sistem expert ce poate funcționa ca o "interfață inteligentă". Astfel, pentru evaluarea vârfurilor ce reprezintă faptele, ERS poate apela de la sine funcții ajutătoare specifice aplicației, fără a întreba utilizatorul despre faptele

Inițial AL/X a fost dezvoltat în Pascal standard pentru calculatoare PDP 11/34 sub sistemul de operare Unix. Mai târziu au fost realizate

elementare. Aceste funcții ajutătoare pot furniza informații actuale din diverse baze de cunoștințe.

Sistemul decizional **DART** (Duplex Army Radio/Radar Targeting Aid) [68], funcționând ca interfață inteligentă pe calculatoare VAX 11/80, are motorul de inferență foarte asemănător cu cel al lui ERS. Dezvoltarea bazei de cunoștințe a sistemului DART se începe prin folosirea în mod consultativ a sistemului vid. Acest lucru face posibilă încărcarea bazei de cunoștințe, fără considerarea interfețelor necesare. Funcțiile ajutătoare ale vârfurilor, ce reprezintă situațiile faptelor elementare, se conturează treptat, fiind din ce în ce mai bine definite pe măsura utilizării.

Deci, în construirea sistemelor ce funcționează ca interfețe (pentru susținerea cărora a fost realizat ERS) se parcurg două faze esențiale. Prima dată trebuie realizată dezvoltarea bazei de cunoștințe –prin modul consultativ, după aceea trebuie dezvoltate funcțiile ajutătoare, pentru a putea cupla sistemul cu o bază deja existentă (sau cu un sistem decizional creat anterior). Precum AL/X, și ERS a fost realizat în Pascal. În raportul lui Lehner și Donnel [120] se prezintă o aplicare a sistemului ERS ca sistem expert pentru proiectare de mentenanță.

DELTA (Diesel-Electric Locomotive Troubleshooting Aid) [38] este un sistem expert realizat pentru a facilita depistarea defecțiunilor la locomotive și pentru furnizarea soluțiilor necesare remedierii. Sistemul dispune de un mecanism decizional hibrid: raționare prin avansare de la fapte la consecințe (înainte) și în direcție inversă, permițând urmărirea și în sens opus, adică deducerea faptelor din consecințe (înlănțuire înapoi). Modul general de gândire al sistemului se bazează pe paradigma eliminării erorilor, adică repetarea de un număr suficient de ori a ciclului de tipul: "Caută o greșeală, elimin-o, verifică dacă au mai rămas probleme". Cunoștințele de specialitate sunt reprezentate prin reguli ȘI/SAU, forma simplificată a unei asemenea reguli fiind:

DACĂ: motorul nu dezvoltă putere
ȘI
nivelul presiunii combustibilului este scăzut
ȘI
indicatorul de presiune a combustibilului este corespunzător

ATUNCI: sistemul de alimentare cu combustibil este defect.

Partea condițională a regulilor este alcătuită din condițiile necesare și suficiente pentru depistarea defecțiunilor. Regulile sunt înmagazinate separat, conform apartenenței lor la diferite domenii de probleme (exemplu: componente mecanice, electronice). Pentru ca întotdeauna să se activeze doar regulile esențiale din punctul de vedere al mulțimii actuale de indici, la grupele de reguli se atașează indecși de către așa numite *metareguli*. Sistemul DELTA folosește mai mult de 500 de reguli, din care aproximativ 300 servesc cunoștințele de diagnosticare, iar restul folosesc sistemului de ajutor (Help) –răspund întrebărilor puse de către utilizatori și oferă alte informații complementare. Interfața conectabilă la disc optic

poate oferi ajutor în procesul de învățare, de efectuare a reparațiilor și în colectarea răspunsurilor date la întrebările furnizate de către sistem. Deși original DELTA a fost conceput și realizat în LISP, a fost transcris ulterior în FORTRAN pentru a putea fi exploatat mai bine pe microcalculatoare. Rulează pe PDP 11/23 necesitând 10Mb spațiu pe disc, cititor pentru discuri optice (CD-ROM) și terminal cu monitor grafic.

GEN-X (Generic Expert System) [135] a fost realizat pentru sprijinirea dezvoltării sistemelor expert pe microcalculatoare. Dintre domeniile de aplicare se pot aminti: sistem expert pentru descoperirea defectelor motoarelor la aeronave, sisteme expert pentru dirijarea proceselor industriale etc. GEN-X nu este un sistem de tipul lui AL/X: se compune dintr-un gestionar al bazei de cunoștințe, programe interpretatoare și generatoare de cod. Utilizatorul poate opta, alegând diverse forme de reprezentare ale cunoștințelor (grafuri, rețele de succesiune, arbori de decizie). Gestionarul bazei de cunoștințe oferă instrumente grafice bazate pe dialog, pentru realizarea, editarea și corectarea conținutului. Interpretoarele conduc etapele consultative pe baza specificațiilor existente în baza de cunoștințe sau primite prin interfața utilizator. Sistemul generează separat baza de cunoștințe și sistemul deductiv, prin etapele consultative corespunzătoare. Dacă dezvoltarea bazei de cunoștințe atinge un nivel satisfăcător, atunci generatoarele de cod realizează traducerea în programe (folosind limbajele C, Ada, Fortran, Pascal). Deși GEN-X a fost realizat inițial pe calculatoare mai mari (în C), rulează și pe microcalculatoare, iar scopul declarat la realizarea lui a fost generarea codului pentru aplicații pe microcalculatoare. Pe un calculator personal (IBM PC compatibil) sunt suficienți deja 32kb memorie (din 256kb recomandate) pentru o bază de cunoștințe ce conține 1000 de reguli de tipul ȘI/SAU.

APES (Augmented Prolog for Expert Systems) [95; 168] este un mediu destinat realizării sistemelor expert care se bazează pe Micro-PROLOG, în care utilizatorul poate alege modul în care să fie tratate incertitudinile. Sistemul permite alegerea unor moduri de lucru cu tehnici deductive asemănătoare cu cele din EMYCIN sau de tipul Bayes (ca în AL/X). Partea probabilistică a motorului de inferență poate fi combinată cu capacitatea deductivă logică implantată în PROLOG. Iată câteva dintre domeniile în care există aplicații APES: stabilirea dreptului la ajutor social, alegerea și clasificarea terenurilor pentru construirea barajelor, stabilirea tratamentului corespunzător de urmat la anumiți bolnavi incurabili, diagnosticarea corodării conductelor. Printre aceste aplicații există sisteme ce conțin mai mult de 400 de reguli.

Există foarte multe versiuni și variante ale limbajului PROLOG (Arity/PROLOG, Micro-PROLOG, PROLOG-2, Quintus-PROLOG, etc.). Una dintre cele mai cunoscute variante, deja tradițională în această parte a Europei, este MPROLOG (Modular Programming in Logic) dezvoltată în Ungaria. Inițial APES a fost un mediu independent (practic o colecție de unelte și utilitare) destinat dezvoltării sistemelor expert, construit în jurul limbajului Micro-PROLOG. În prezent însă aproape fiecare sistem PROLOG are deja elaborat un asemenea mediu propriu (denumit tot APES) care poate fi folosit la dezvoltarea de sisteme expert.

ACLS (Analog Concept Learning System) [100; 160] este în stare să genereze individual regulile de clasificare caracteristice domeniului abordat, din exemple diverse de aplicare. Diferă astfel de sistemele prezentate în prealabil, necesitând o participare diferită (și într-un anumit sens o participare la un nivel mai scăzut) din partea specialistului la elaborarea bazei de cunoștințe. În sistemele de genul AL/X, realizarea bazei de cunoștințe însemna de fapt achiziționarea regulilor necesare, gata pregătite, de la specialistul (expertul) domeniului ales. În comparație cu acesta, la dezvoltarea unui sistem expert sub ACLS, la început se definește mulțimea atributelor primitive și a claselor necesare organizării corespunzătoare a domeniului abordat. După această etapă, expertul uman comunică exemple evidențiate sistemului (precizând valorile atributelor și regulile aparținătoare). Pe parcursul prelucrării acestora, ACLS induce individual regula generalizată pentru exemplele primite sub forma unui arbore de decizie. Atributele, comunicate de către specialistul uman, care se dovedesc mai târziu irelevante, nu ajung să fie inserate în regula generală. În final expertul uman poate evalua dacă regula indusă de către sistem reflectă, sau nu, corespunzător cunoștințele sale într-un mod suficient de exact. În cazul unui eșec, el poate efectua corecturi prin precizarea unor exemple și/sau atribute noi.

ACLS poate funcționa atât sub formă de pachete de prelucrare (batch) cât și sub formă interactivă. În modul interactiv, utilizatorul poate verifica și examina din pas în pas arborele de decizie generat, urmărind modificarea acestuia după prelucrarea fiecărui exemplu. Sistemul poate genera și un cod executabil (folosind sintaxa limbajului Pascal), reprezentând arborele de decizie prin reguli decizionale structurate (IF-THEN-ELSE). ACLS a fost elaborat în UCSD Pascal pe calculatoare Apple II și IBM PC. Există și o versiune îmbunătățită care dispune de o interfață grafică cu meniuri și este în circulație sub numele Expert-Ease.

RuleMaster [139] conține algoritmul inductiv de învățare prezentat la ACLS, dar implantat într-un mediu mai extins de dezvoltare, fiind capabil în același timp să organizeze ierarhic arborii de decizie induși cu regulile, respectiv să prezinte explicații referitoare la deducțiile realizate în cadrul sistemului. Regulile induse sunt înmagazinate automat în module, prin intermediul limbajului Radial. Acest limbaj dispune de mecanisme pentru realizarea modulelor, verificarea și urmărirea aplicării regulilor, accesarea resurselor externe de informații, operatori probabilistici și fuzzy, module explicative. Predarea controlului între module se realizează prin apeluri ierarhice.

Aplicabilitatea sistemului RuleMaster a fost demonstrată de două exemple practice. Sistemul Willard [139], folosit pentru presemnalarea furtunilor puternice, este compus din 27 de module (fiecare din aceste module fiind un arbore de decizie indus în mod individual). Compararea sesizărilor furnizate de către acest sistem cu cele de la Centrul Național de Urmărire al Furtunilor (S.U.A.) a demonstrat eficiența extraordinară a sistemului. O altă aplicație (amintită în [139]) s-a dovedit foarte eficientă la diagnosticarea transformatoarelor mari pe baza examinării uleiului din ele. RuleMaster a fost realizat în C și poate rula sub sistemul de operare Unix pe calculatoare IBM PC, Sun Micro sau DEC Vax.

TIMM (The Intelligent Machine Model) [107] conține un sistem de achiziționare cunoștințe capabil să definească regulile automat, cuplat cu un motor de inferență fuzzy. Expertul uman definește la început problema, prin precizarea mulțimii de atribute și valorile posibile ale acestora. După această etapă, expertul cooperează cu sistemul pentru realizarea regulilor de producție: sistemul generează diferite situații pe baza definițiilor date pentru domeniul abordat, situații ce sunt evaluate de către specialistul uman. În etapa de consultare, sistemul va furniza utilizatorului curent răspunsuri și sugestii luând în considerare această evaluare (factorul decizional precizat de către expertul uman). Pe lângă această evaluare, expertul uman poate marca valorile atributelor ce caracterizează situațiile deosebite și poate fixa pentru acestea valoarea exactă a factorului decizional.

În TIMM este posibilă și examinarea globală a mulțimii de reguli. Sistemul poate depista situațiile insuficient acoperite de reguli prin valorile atributelor. Căutând regulile ce dispun de condiții asemănătoare, dar care au efecte contradictorii, sistemul poate efectua și o verificare a consistenței bazei de cunoștințe.

Aranjând atributele ce descriu condițiile regulilor (pe baza unor valori asociate) se obține un registru special în domeniul reprezentării cunoștințelor. Folosindu-se de acesta, motorul deductiv aranjează doar parțial condițiile regulilor, permițând ca partea condițională a unei reguli să nu se potrivească exact la o situație dată. Privit astfel, se poate afirma că mecanismul deductiv al sistemului TIMM este analogic, fiind apreciat că aplică întotdeauna regula cea mai potrivită situației date. Diversele variante dezvoltate pentru IBM PC-uri ale sistemului TIMM se ocupă cu preponderență de analize financiare și asigurări pentru ambarcațiuni. Sistemul TIMM/Tuner [107] a fost utilizat la configurarea și reglarea performanțelor optime la sistemul de operare VMS pe calculatoare DEC VAX (deja în cadrul testărilor beta s-a obținut o creștere medie a performanțelor sistemelor VAX, apreciată la 15%). Sistemul TIMM a fost realizat în FORTRAN și poate fi rulat pe calculatoare compatibile IBM PC (care dispun de coprocesor matematic incorporat) sau pe sisteme mai mari. Pe IBM PC-uri necesită 640kb RAM și un spațiu de 10Mb pe disc.

Personal Consultant era instrumentul conceput de către Texas Instruments, destinat proiectării și dezvoltării sistemelor expert pe calculatoarele profesionale TI. Acest sistem conține, în esență, proceduri alcătuite în principal din comenzi și este înzestrat cu structuri de reguli asemănătoare cu cele din EMYCIN (din această cauză putând fi utilizat și pentru transpunerea aplicațiilor din EMYCIN în mediul aparatelor profesionale TI).

Personal Consultant a fost realizat în LISP. Configurația recomandată era alcătuită dintr-un TI Professional Computer cu sistem de operare MS-DOS (chiar și versiunea 2.1 era suficientă), dispunând de 512kb memorie internă și 10Mb spațiu pe disc. Într-o asemenea configurație permitea deja manipularea unui sistem alcătuit din 400 de reguli.

M.1 poate fi deasemenea folosit pentru transpunerea aplicațiilor EMYCIN pe microcalculatoare. A fost utilizat în cadrul pachetului educațional TI, ca sistem demonstrativ pentru instruirea specialiștilor în sisteme expert. Merită amintită una din variantele acestui sistem,

denumită **SACON** (Structural Analysis Consultant [32]), dezvoltată cu scopul de a oferi sugestii în utilizarea unui program de simulare pentru analize structurale complexe.

M.1 a fost realizat în MICRO-PROLOG, configurația minimă necesară pentru a-l utiliza fiind 128kb memorie internă sub sistemul de operare DOS (versiunea 2.0 sau mai mare la MS-DOS) pe un calculator compatibil IBM PC cu monitor color. În această configurație minimă, sistemul poate gestiona aproximativ 200 de reguli.

K:base oferă o altă modalitate de reprezentare a cunoștințelor față de sistemele prezentate, care funcționau mai mult sau mai puțin folosind aceleași scheme de reprezentare. Mai exact, pretindeau din partea expertului uman cunoștințe corespunzătoare euristicii folosite de către acesta, traduse în reguli individuale (separate), sau erau în măsură să inducă asemenea reguli, pe baza unor exemple furnizate de către experții umani.

Deși contemporan cu celelalte sisteme prezentate anterior, K:base reprezintă oarecum o cotitură față de acestea deoarece reprezentarea cunoștințelor în acest sistem s-a realizat prin cadre (frames) – o structură apărută prin programarea orientată pe obiecte. Aceste cadre erau în esență structuri de date statice, care uneau cunoștințele declarative referitoare la obiecte cu cunoștințele procedurale despre folosirea și manevrarea acestora [141]. Această abordare a reprezentării părea mai potrivită pentru bazele de cunoștințe complexe, în raport cu reprezentarea bazată pe reguli, ușurând semnificativ și reprezentările grafice pe terminale (monitoare) a cunoștințelor implementate în sistem. Inițial, domeniul de aplicare a sistemului K:base a fost analiza diferențelor dintre ratele dobânzilor între două grupuri financiare. Sistemul a fost scris în LISP pentru mașina LISP Symbolics 300 și pentru calculatoare IBM PC cu cel puțin 512kb RAM.

După cum reiese din varietatea și succesul sistemelor descrise anterior, sistemele expert elaborate și dezvoltate pe microcalculatoare oferă posibilități din ce în ce mai importante pentru utilizatori. Avantajele principale ale acestor sisteme corespund cu avantajele generale pentru care celelalte aplicații pe microcalculatoare sunt atât de căutate: accesabilitate, costuri reduse și portabilitate. Primele două permit copierea aceluiași sistem în mai multe exemplare, ceea ce constituie o condiție fundamentală pentru răspândirea cu costuri reduse. În ceea ce privește portabilitatea: dimensiunile reduse ale microcalculatoarelor, precum și întreținerea ușoară (fără pretenții deosebite), împreună cu rezistența la condițiile de exploatare, sunt argumente serioase care permit exploatarea lor și în spații reduse, în condiții de confort redus (un exemplu elocvent este utilizarea sistemului AL/X pe platformele de forare din Marea Nordului). Dezvoltarea spectaculoasă a pieței microcalculatoarelor, atât prin evoluția prețurilor cât și prin accesibilitatea și dezvoltarea resurselor, a permis răspândirea aproape generală a acestora: în instituțiile publice, în cercetare, proiectare, producție, învățământ, acasă etc.

Avantajul cel mai important al acestor sisteme rezidă în faptul că utilizatorii lor pot efectua încercări cu un sistem expert "domestic" asumându-și riscuri materiale minime, putând evalua și singuri eficiența previzibilă datorată utilizării acestei tehnologii în propriul

domeniu de activitate. La ora actuală există domenii publice (pe internet și nu numai) dedicate sistemelor expert și mediilor destinate dezvoltării acestora, deci aproape oricine își poate încerca talentul în realizarea sau în testarea și folosirea acestor sisteme. Companiile cele mai renumite în crearea softului (realizatori de sisteme de operare, pachete consacrate de aplicații, jocuri, etc.) recurg în mod curent la sistemele pe bază de cunoștințe pentru menținerea și deservirea "liniilor fierbinți" prin internet, destinate sprijinului tehnic și suportului oferit consumatorilor.

CLIPS (C Language Integrated Production System) este unul dintre cele mai recente și cele mai celebre instrumente pentru dezvoltarea sistemelor expert. Acesta oferă la ora actuală un mediu complet pentru realizarea sistemelor bazate pe reguli și/sau obiecte. Este folosit pe scară largă mai ales în S.U.A., de la publicul interesat până la diverse ramuri ale NASA, birouri federale, antreprenori guvernamentali, universități și diverse companii. Prima versiune a acestui sistem a fost realizată, ca și prototip, în primăvara anului 1985 de către Secția de Inteligență Artificială a Centrului Spațial Johnson din NASA, urmărind asemănarea cu sistemul ART realizat de către firma Inference Corporation. Ultima versiune a sistemului (6.1) datează însă din vara anului 1998. Printre caracteristicile și facilitățile fundamentale ale acestui sistem merită să amintesc următoarele:

Din punctul de vedere al reprezentării cunoștințelor, este o unealtă coezivă pentru tratarea unei varietăți extinse de cunoștințe cu suport pentru trei paradigme diferite de programare: pe bază de reguli, orientat obiect și procedural. Programarea pe bază de reguli permite reprezentarea cunoștințelor euristice prin specificarea unui set de acțiuni ce trebuie întreprinse pentru o situație dată. Programarea orientată obiect permite ca sistemele complexe să fie modelate prin componente modulare (care pot fi reutilizate ușor pentru a modela alte sisteme sau pentru a crea componente noi). Posibilitățile oferite de CLIPS în domeniul programării procedurale sunt similare celor din limbajele C, Pascal, Ada și LISP. CLIPS a fost scris în C, pentru portabilitate și viteză de execuție, și a fost implementat pe mai multe sisteme de calcul fără a fi nevoie de schimbarea codului. Printre altele, a fost testat pe calculatoare compatibile IBM PC sub sisteme de operare DOS, Windows 95, pe Macintosh sub sisteme de operare MacOS și Mach. Având în vedere că CLIPS se livrează cu sursa completă (care poate fi modificată și ajustată după nevoile utilizatorului), se poate implementa pe orice sistem de calcul care dispune de un compilator C corespunzător ANSI. Versiunea standard a sistemului oferă un mediu interactiv de dezvoltare (orientat text) cu unelte de depanare, help on-line și editor integrat (sau dezvoltat interfețe cu meniuri pentru medii Macintosh, Windows 95 și X Window). Există deasemenea facilități pentru verificarea și validarea sistemelor expert (permite proiectarea modulară și partiționarea bazei de cunoștințe; verificarea valorilor și argumentelor de funcții precum și a constrângerilor, atât static cât și dinamic; analiza semantică a seturilor de reguli pentru inconsistențe) și dispune de o documentație extinsă (Ghid de Utilizare, Manual de Referință, exemple etc.). În prezent sistemul este menținut ca și software în domeniul public de către principalii autori (care nu mai sunt angajați la NASA), iar utilizarea acestuia în scopuri educaționale este gratuită.

3.3 SISTEMELE EXPERT ÎN DOMENIUL CONSTRUCȚIILOR

3.3.1 Introducere

După toate aparențele, schimbările în curs de realizare datorate aplicării inteligenței artificiale vor fi extinse și foarte semnificative și în domeniul construcțiilor. Ceea ce azi poate să pară doar un curent la modă (chiar "de dragul artei" după părerea multora), are toate șansele să devină începutul unei revoluții majore în ceea ce privește modul în care luăm deciziile. În prezent, sistemele expert furnizează arhitecților, inginerilor și tehnicienilor din domeniul construcțiilor o gamă de instrumente noi, pentru a-i asista în deciziile pe care trebuie să le ia în multe situații pe parcursul carierei profesionale. Este de regretat faptul că lipsa (deocamdată a) unui sistem expert comercial major în ingineria construcțiilor face ca aria cercetărilor să pară prea teoretică pentru majoritatea inginerilor practicanți.

Cercetările legate de sisteme expert, precum și legate de aplicarea acestora în construcții, sunt axate pe mai multe direcții. Dacă la prima conferință dedicată exclusiv domeniului calculului construcțiilor: CIVIL-COMP (organizată în 1985), au fost prezentate doar patru lucrări bazate pe probleme de inteligență artificială, la ediția a patra a acestei conferințe (*The Fourth International Conference on Civil and Structural Engineering Computing*, 1989) au fost publicate deja 36 de lucrări legate de această problemă, organizate pe următoarele secțiuni:

- resurse umane și perspective profesionale,
- reprezentarea cunoștințelor și raționamente,
- planificarea și organizarea în construcții,
- management în construcții,
- proiectarea construcțiilor,
- analiza structurală,
- proiectarea structurală,
- ingineria seismică,
- ingineria geotehnică,
- tehnologia lucrărilor de construcții și robotică,
- educația și sistemele expert.

Se pare că cel mai îmbrățișat domeniu, din punctul de vedere al aplicațiilor de sisteme expert, este managementul în construcții. Dar în ultimul timp se poate observa o dezvoltare spectaculoasă a cercetărilor și a realizării aplicațiilor de inteligență artificială în ingineria seismică, ingineria apelor și hidrotehnică, ingineria mediului, ingineria materialelor, optimizarea structurală, legislația construcțiilor, ingineria transporturilor etc. În cele ce urmează, voi prezenta succint unele aspecte mai importante ale stadiului actual al cercetărilor și al aplicațiilor realizate în domeniul construcțiilor, pornind de la a patra conferință CIVIL-COMP amintită.

3.3.2 Sisteme pentru resurse umane și perspective profesionale

În acest domeniu se poate remarca lucrarea lui C. Mitchell (de la firma *C. & T. Mitchell Ltd., Kensington, London*): *Expert Systems and Engineering Management* [183] în care sunt explorate implicațiile sistemelor expert din perspectiva managementului resurselor în construcții. Autorul examinează cerințele sistemelor expert legate de două resurse importante în afaceri: oamenii și banii (resurse umane și financiare), tratând în particular dezvoltarea și întreținerea acestor sisteme. În lucrare se demonstrează că, dacă se adoptă strategii convenabile pentru amortismente, se pot obține reduceri semnificative la costurile de menținere atât a resurselor umane cât și a resurselor financiare. Asemenea strategii pot fi implementate în sisteme expert, iar utilizarea acestora ar deschide oportunități mai favorabile pentru întreprinderi și pentru întreprinzători. În lucrarea *Knowledge Based Systems: Potential in Design and Management* (autor: S. Rowlinson, *Department of Civil & Structural Engineering, University of Hong Kong, Hong Kong*) [183] se prezintă căile de succes spre dezvoltarea sistemelor bazate pe cunoștințe în domeniul construcțiilor, fiind citate și câteva exemple din Japonia, Regatul Marii Britanii și Statele Unite ale Americii. De asemenea, în această lucrare este prezentată sub forma unui schelet o taxonomie a tipurilor de domenii, prin care se poate discuta complexitatea sistemelor în dezvoltare. Sunt revăzute anumite scopuri în cercetare și abordări practice în dezvoltarea sistemelor expert, comentându-se impactul lor în obținerea succesului și a performanței în domeniul construcțiilor. Spre final se discută procesul de dezvoltare și necesitatea îmbunătățirii dezvoltării metodologiilor, această discuție fiind urmată de o sumară recapitulare a naturii răspunsurilor din partea industriei construcțiilor, la problemele ridicate de sistemele cu baze de cunoștințe, fiind descrise și două proiecte în curs de elaborare din Hong Kong. Lucrarea se încheie cu considerente asupra inițiativelor din cadrul dezvoltărilor viitoare.

Luând ca exemplu Japonia, poate fi instructivă observația referitoare la numărul și tipurile sistemelor utilizate în practică. Ishii K. și Hayami S. [102] raportau 34 de sisteme expert în exploatare practică, din 121 de sisteme observate, în anul 1988! Din aceste 121 de sisteme, 14 erau în domeniul construcțiilor, dar dintre acestea doar unul singur era în exploatare curentă: un sistem expert pentru proiectarea caselor (și implicit a locuințelor) prefabricate, folosit de către *Sekisui Chemicals*. Deși se consideră că Japonia ar avea o populație "în grafitezare", cele mai multe baze de cunoștințe realizate se referă la îmbunătățirea proiectării și a productivității, contrar oarecum tendințelor de reducere imediată a costurilor, de arhivare a cunoștințelor și de diagnosticare (acestea din urmă fiind caracteristicile dominante ale bazelor de cunoștințe realizate în vest). În Japonia, dezvoltarea sistemelor expert în domeniul legislației pentru construcții și-a pierdut avântul, sisteme complete dovedindu-se ineficiente în raport cu experții umani (datorită nu atât ambiguității și interpretării diverselor acte legislative și normative de multe ori contradictorii, cât mai degrabă datorită intervențiilor directe guvernamentale în construcții, deseori cu legi și regulamente excesive).

H. Adeli prezintă în [12] o listă cuprinzând domeniile funcționale în care pot fi utilizate sisteme cu baze de cunoștințe (de exemplu: diagnosticare, detectarea defectelor,

presemnalări, interpretări, monitorizări, instrucție, planificare și proiectare, se mai pot adăuga la această listă și alte domenii funcționale, cum ar fi: luarea deciziilor etc.). Această listă este urmată de o listă a disciplinelor, precum și de exemple concrete de domenii în care există dezvoltări de sisteme bazate pe cunoștințe (KBS= *Knowledge Based System*). Figura 7 prezintă o taxonomie propusă pentru bazele de cunoștințe cu

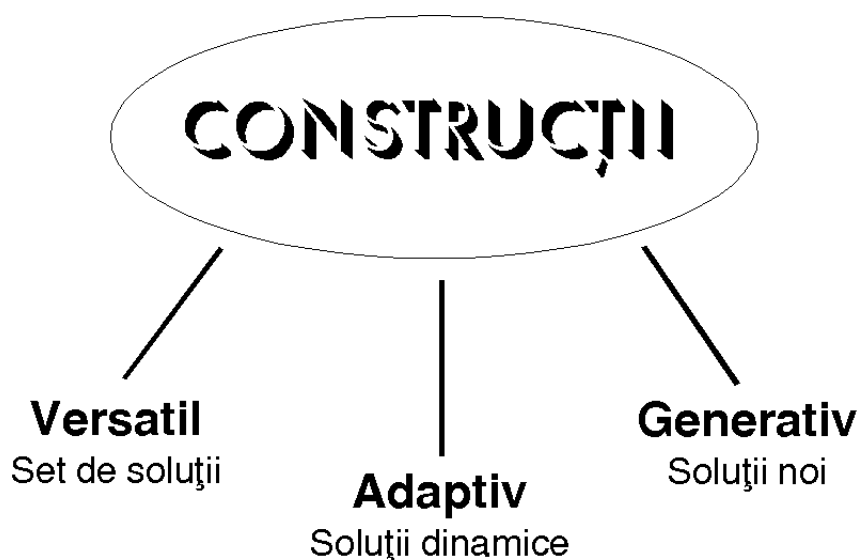


Fig.7 O taxonomie propusă pentru baze de cunoștințe în construcții

aplicabilitate în construcții. Aceasta se întinde de la capătul versatil al spectrului, parcurgând sistemele adaptive, până la sistemele generative. Sistemele versatile sunt considerate cele care satisfac condițiile de adaptabilitate prin coduri, proceduri sau scenarii definite, cum ar fi sistemele destinate analizei structurale (ex.: [13]), pre- și post-procesoarele (ex.: [173]) la programe mai complexe, și câteva sisteme simple de diagnosticare sau din domeniul legislației. Acestea produc de regulă o serie de seturi cu soluții. Sistemele adaptive au ca scop producerea unor soluții croite după cerințe individuale și situații inedite, cum ar fi învățarea asistată de calculator (ex.: [157]), sisteme "on-line" (ex.: [155]), sisteme estimative (ex.: sistemul **PRC** dezvoltat la Universitatea Tsing Hua sub conducerea profesorului Lu), respectiv sisteme de planificare și supraveghere (ex.: [163]; [20]; [62]). Astfel soluțiile produse pot fi descrise ca dinamice (dar nu neapărat și originale). Sistemele generative se consideră cele care produc într-adevăr soluții noi (originale) la probleme, cum ar fi pachete specializate de proiectare (ex.: [66]), sisteme complexe legislative (ex.: [26]) și pachete pentru sistematizare, alegerea și studierea amplasamentelor (ex.: [182]).

Această taxonomie prezintă astfel un nivel de dificultate crescândă de la stânga la dreapta. Prin această creștere a dificultății, se majorează și resursele necesare de investit în realizarea și dezvoltarea sistemelor. Nu e deloc surprinzător că majoritatea sistemelor care au înregistrat succese se găsesc la partea stângă a acestui spectru. După opinia lui H. Adeli, sistemele expert există mai ales în domeniile unde judecata și experiența joacă un rol important și nu există o teorie ca suport, sau teoria este inadecvată. Totuși, utilizarea cea mai eficientă a judecății și a experienței în KBS, după părerea lui Adeli, s-ar găsi prin alegerea între teoriile concurente și soluții, în diferite contexte, iar dezvoltarea bazelor de cunoștințe este dezavantajată acolo unde teoriile sunt inadecvate. Ca și exemplu care susține aceste afirmații, poate fi amintit proiectul **RICS "Alvey"** [42] de la *Salford University*, în cadrul căreia au fost obținute rezultate excelente cu trei module ale unui sistem de planificare strategică a proiectelor, dar dezvoltarea celui de al patrulea modul s-a dovedit

a fi deosebit de dificilă datorită absenței cercetării fundamentale coerente și a lipsei consensului asupra noțiunii. Dr. M. Alshawî, în lucrarea sa intitulată: *The Potential of Object Oriented Programming for The Development of Expert Systems in The Construction Industry* [183] ilustrează metodele variate de reprezentare ale cunoștințelor, scoțând în evidență tehnica orientată pe obiecte. Dezvăluie formele, avantajele și impactul viitor al acestei tehnici în dezvoltarea sistemelor expert și a bazelor de cunoștințe. Etalează mai multe exemple pentru ilustrarea conceptelor din spatele noțiunilor de "obiect", "cadru", "moștenire" și referitoare la instrumente hibride de dezvoltare.

3.3.3 Reprezentarea cunoștințelor și a raționamentelor în construcții

În ceea ce privește reprezentarea cunoștințelor și a raționamentelor, trebuie amintită lucrarea *Metaknowledge Reasoning in Civil Engineering Expert Systems* (autori: A. Oki și Dr. D. Lloyd Smith, *Department of Civil Engineering, Imperial College, London*) în care se prezintă o metodă simplă - dar care oferă un potențial puternic - pentru disponibilizarea raționamentelor prin metacunoștințe în cadrul unui mediu de programare logică/procedurală, destinat dezvoltării aplicațiilor de tipul sistemelor expert în construcții, efectuând corecții și ajustări minime asupra unor structuri de date existente.

În cazul reprezentării bazate pe reguli, orice implementare serioasă a sistemelor expert, ar conține probabil sute de reguli. Un mediu de programare proiectat necorespunzător, ar tinde să selecționeze prea multe reguli irelevante în cadrul căutării soluției la orice problemă specifică adresată sistemului. Davis a definit în [52] preponderența unor asemenea reguli în cadrul procesului de căutare, ca și *saturație*. Chiar și într-un mediu de programare bine proiectat (cum este XLOG în prezentarea lui Dr. Bruno Feijo [65]) se poate manifesta acest fenomen de saturare, în ciuda existenței unui mecanism intern pentru definirea criteriilor de căutare. Dacă însă sistemul ar dispune de cunoștințe despre propria bază de cunoștințe și ar putea raționa folosind aceste metacunoștințe, efectul saturației ar putea fi considerabil redus.

Raționarea cu ajutorul metacunoștințelor, sau altfel spus raționarea prin metareguli, nu reprezintă o metodă nouă în inteligența artificială, totuși implementările uzând de această metodă sunt puține în domeniul construcțiilor. Metoda prezentată în lucrarea mai sus amintită se bazează în esență pe ideile lui Davis [52], [53], dar acestea sunt aplicate într-o manieră mai simplă. O asemenea abordare poate fi utilizată atât pentru selectarea individuală a regulilor cât și pentru disponibilizarea alegerii traseelor de raționare (după cum se arată în [96]). Deși la lucrarea amintită mediul de programare folosit era XLOG, tehnica prezentată poate fi transpusă în orice alt mediu de programare (de la limbaje cum ar fi C, LISP, PROLOG, până la ansambluri alcătuite din instrumente și unelte hibride de inteligență artificială, ca ART și KEE). După convingerea autorului lucrării menționate, mediile strâns aliniate la logica matematică oferă fundamente corespunzătoare pentru dezvoltarea unor aplicații ingineresti în construcții, în special utilizând raționări prin metacunoștințe (asemenea sisteme fiind bazate în esență pe reguli).

În lucrarea *A Prolog Based Object Oriented Engineering DBMS* (autori: Dr. A.S. Watson și S.H. Chan, *Department of Civil Engineering, University of Leeds*), publicată în 1989, sunt prezentate pentru prima dată conceptele primare ale **PBASE** (prototipul unui sistem de gestiune baze de date, orientat obiect). Acest sistem a fost destinat pentru susținerea aplicațiilor ingineresti, în mod special în domeniul calculului structural. Pentru a se conforma cerințelor ingineresti, modelul orientat obiect care a fost folosit în PBASE încorporează mai multe îmbunătățiri față de alte sisteme similare, cum ar fi: structura evoluată a bazei de date, obiecte compozite, metode declarative și administrarea versiunii. Autorii consideră că în proiectarea construcțiilor evoluția este trăsătura de bază a procesului de proiectare. Un sistem cu bază de date, care suportă asemenea aplicații, trebuie să dovedească și abilitatea de a suporta evoluarea proiectului. Aceasta se întâmplă în mod normal dacă baza de date poate evolua incremental și logic împreună cu procesul de proiectare. Pentru a reflecta procesul de evoluție al proiectării este necesară o flexibilitate considerabilă în definirea dinamică și în modificarea structurii bazei de date. Bazele de date tradiționale alcătuite pe baza unor modele de date hierarhice sau distribuite în rețea, impun o structură relativ rigidă care nu poate fi modificată ușor. Schimbările majore în asemenea cazuri presupun oprirea sistemului. În PBASE s-a folosit o modalitate compusă pentru evoluția structurii. Schimbarea claselor și a rețelei de clase alcătuite au condus la conversia tuturor termenilor afectați pentru a se conforma noii scheme. Suportarea obiectelor compozite face parte din sistemul de relații dintre ansamblu și componente.

În construcții entitățile sunt grupate frecvent împreună pentru a forma o entitate la un nivel mai înalt. Pentru a susține eficient dezvoltarea aplicațiilor destinate domeniului construcțiilor, baza de date trebuie să dispună de abilitatea acceptării definirii și manipulării unor seturi de entități ca o entitate logică unică. Abstractizarea obiectelor prin cuprinderea lor într-un obiect compus se numește *agregare*. În forma generală, modelul orientat obiect este suficient numai pentru a reprezenta o colecție a obiectelor aflate într-o anumită relație. În PBASE conceptul obiectelor compozite este integrat în modelul orientat obiect pentru a sprijini agregarea. Suporturi pentru agregare, în limbajele de programare orientate obiect, au fost raportate și în Loops [177], CommonLoops [36] și SmallTalk [35]. Atât în Loops cât și în CommonLoops, suportul se cheamă obiect compozit, în timp ce în SmallTalk este descris ca o relație parte-întreg. Noțiunea sub formă de obiect compozit există și în baze de date orientate obiect ca suport pentru agregare, ca de exemplu în sistemul Orion [28]. De fapt, suportul pentru agregare prezentat în PBASE este similar cu cel din Orion, fiind utilizate noțiuni identice.

Rolul bazelor de date clasice a fost restrâns la înmagazinarea datelor, ele jucând un rol aproape nesemnificativ în folosirea acestor date. Cunoștințele referitoare la folosirea datelor erau înmagazinate în programele folosite ca aplicații sau chiar în utilizatori. Această separare netă a datelor și a cunoștințelor despre date nu era naturală și s-a dovedit și neeconomică. Eforturile de a integra cunoștințe în bazele de date, au condus la baze de date semantice, sau - mai recent - la sisteme de date expert (*EDS*) sau la sisteme de gestiune a bazelor de cunoștințe (*KBMS*). În PBASE s-a propus un concept al metodelor declarative

pentru abordarea separării semantice, prin încorporarea cunoașterii declarative în cadrul modelelor datelor, orientate obiect. Metodele declarative integrează paradigmele logice și orientate obiect, prin metode de înlocuire ale obiectelor cu programe logice și considerând lansarea unui mesaj către un obiect ca invocarea unui scop în cadrul acelui obiect. Astfel, metodele declarative conferă obiectelor capacități deductive și de raționare, transformând baza de date a obiectelor abstracte într-o bază de date a obiectelor cu cunoștințe. Astfel, includerea cunoștințelor în obiecte oferă o cale efectivă și practică pentru modelarea cunoașterii.

În cadrul procesului de proiectare, nu numai în construcții ci și în general, există mai multe procese tentative și iterative. Astfel, înainte de a obține un proiect mulțumitor se ajunge de multe ori la generații ale variantelor proiectelor, la combinații ale alternativelor și la revizii. Alternativele la proiect și reviziile vor conduce inevitabil la mutații ale proiectului, toate acestea fiind legate și de timp. În multe domenii de proiectare, urmărirea unei istorii a proiectării și disponibilizarea accesului la informații istorice joacă un rol important. În sistemele de baze de date destinate ingineriei, istoricul proiectării este urmărit și înregistrat în mod curent, acest suport fiind pomenit sub denumiri ca: versiune, controlul versiunii, evoluția proiectului sau administrarea versiunii [117], [30]. Această componentă permite urmărirea versiunii obiectelor și a alternativelor, în măsura evoluării lor pe parcursul procesului de proiectare. PBASE a fost realizat în PROLOG cu intenția declarată de a fi dezvoltat mai departe folosind un sistem de baze de date relaționale pe o stație de calcul SUN, sub sistem de operare Unix. A fost testat și aplicat la modelarea structurală a unor construcții metalice simple de înălțime mică și medie.

3.3.4 Sisteme pentru planificare și organizare în construcții

În domeniul planificării și organizării construcțiilor, în lucrarea intitulată *Knowledge-based Regulation Processing for Site Development* (autor: D.T. Shaw, *Department of Civil Engineering, University of Illinois, Urbana, Illinois, U.S.A.*) s-a descris proiectul conceptual al unui program pentru dezvoltarea situ-urilor municipale. Acest sistem, purtând denumirea **SITE CODE**, a fost elaborat pentru a studia modalitatea încorporării regulamentelor în sistemele destinate proiectării asistate de calculator a amplasamentelor. Dezvoltarea acestui proiect își trăgea începuturile din studiile privind normativele referitoare la organizarea șantierelor din Austin (Texas), San Diego (California) și Fairfax County (Virginia), realizate cu sprijinul material al Fundației Naționale pentru Știință din Statele Unite ale Americii. Funcția primară a programului SITE CODE era de a extrage din sistem toate cerințele de proiectare aplicabile în cazul unui proiect particular. A fost dezvoltat un sistem organizatoric pentru regulamente, cu scopul de a facilita extragerea cerințelor de proiectare bazate pe:

- ❶ districtul sau regiunea geografică pentru care este valabilă regula,
- ❷ perioada de timp pe durata căreia este efectivă, și
- ❸ procesul administrativ sau juridic (zonare, subdivizare teritorială, reconsiderarea amplasamentelor și a limitelor de proprietate etc.) cu care este asociată regula.

SITE CODE era alcătuit dintr-o bază de date hibridă (bază de reguli/reprezentare orientată obiect) a reglementărilor referitoare la amplasamente, denumită bază de cunoștințe normative. În timpul sesiunii era creată în mod dinamic o bază de cunoștințe orientată obiect a cerințelor, care conținea informații de proiectare specifice extrase din normative și reglementări. Concepția sistemului a urmărit exemplul oferit de J.H.Garrett și S.J. Fenyves în 1987 [73]. Aceștia au folosit, ca fundament pentru realizarea unei rutine de proiectare automată, un model denumit "multi-component" al standardelor referitoare la proiectarea structurală, purtând numele **SPEX**. Programul dezvoltat în cadrul acestui sistem, prin care se realiza proiectarea modelului, se compunea din următoarele părți:

- ❶ o rețea alcătuită din tabele de decizie care reprezentau porțiunile logice și substantive ale codificării,
- ❷ un sistem organizatoric (de indexare) care era folosit pentru extragerea eficientă a cerințelor ce prezentau interes privind aplicarea într-o situație particulară de proiectare în cadrul sistemului.

Sistemul organizatoric din SPEX clasifică fiecare cerință, în raport cu tipul și alcătuirea fiecărui obiect structural (ex.: stâlp metalic cu zăbrele), starea de tensiuni (ex.: compresiune axială) și limita efortului (ex.: flambaj) căruia îi corespunde cerința. SITE CODE a fost implementat ca bază de cunoștințe în 1989, rulând pe un calculator Apple Macintosh IIX cu 4Mb memorie. Platforma folosită pentru dezvoltare a fost NEXPERT OBJECT (realizat de către *Neuron Data Inc., Palo Alto, California, U.S.A.*), un sistem avansat pentru reprezentarea și procesarea cunoștințelor, suportând atât modelarea bazată pe reguli cât și cea orientată obiect. Pachetul AI VISION (*Neuron Data Inc.*) a fost utilizat pentru crearea interfeței utilizator, rezultând ferestre grafice atractive cu zone senzitive pentru mouse (butoane etc.). Aceste ecrane interactive sunt utilizate și pentru ferestrele cu răspunsuri generate de către sistem la întrebările utilizatorului pe parcursul sesiunii de consultare, ele permițând și cuplarea cu domenii grafice sau cu text prin legături și butoane adiționale (*hyperimage/hypertext screens*).

Există foarte multe prototipuri de sisteme expert referitoare la normative și reglementări în domeniul construcțiilor. Deja în 1985, M.A. Rosenman și J.S. Gero au reușit să programeze cu succes părți din normativul *Australian Model Uniform Building Code*, realizând un sistem expert și demonstrând totodată posibilitatea alegerii pentru modelare a unei baze de reguli în scopul reprezentării reglementărilor din normativul amintit. În urma publicării rezultatelor obținute, au atras atenția asupra necesității unor cercetări în următoarele domenii:

- strategii și metode îmbunătățite de control pentru accesarea bazelor de cunoștințe care cuprind reglementări și normative, în scopul evitării cerințelor excesive de calcul și memorie,
- mecanisme de indexare a bazei de cunoștințe, pentru a mări interacțiunea cu utilizatorii,
- structuri de reprezentare/modelare care ar permite încorporarea mai multor

cunoștințe din domeniul abordat, pentru explicarea și interpretarea informațiilor codificate.

În 1987, în Noua Zelandă, Hosking și colectivul său de cercetare au dezvoltat o bază de cunoștințe pe calculator, destinată tot domeniului construcțiilor, modelând normativul local de protecție împotriva incendiilor [99]. Ei au ajuns însă la concluzia că, deși marea parte a cunoașterii cerute într-un asemenea domeniu de către un sistem expert este conținută în textul codului realizat, experții umani sunt la fel de necesari pentru a interpreta multe informații incomplete sau ambigue din codul realizat.

O altă aplicație notabilă a bazelor de cunoștințe destinate proiectării amplasamentelor era **SEPIC**, un program care oferea asistență în activitatea de sistematizare zonală în Texas, SUA. În acest sistem, regulamentele locale și statale, precum și standardele de proiectare erau exprimate sub forma unor reguli grupate în module distincte. Această bază de cunoștințe era suficient de mică pentru a nu necesita scheme de reprezentare mai sofisticate în arhitectura originală a sistemului. Totuși, W.J. și S.G. Hadden raportau [91] că datorită adoptării de către consiliul orășenesc din Austin a unei ordonanțe comprehensive noi pentru protejarea decantoarelor (ordonanța din discuție înlocuind șase ordonanțe precedente referitoare la aceeași problemă), a fost necesară modificarea bazei de cunoștințe din SEPIC pentru a putea ține cont de faptul că un amplasament, prin virtutea istoriei sale juridice, putea fi:

- subiectul noii reglementări,
- guvernat de una din cele șase hotărâri anterioare, sau
- exceptat de la protejarea decantoarelor.

În reglementările referitoare la sistematizarea teritorială, asemenea dependențe sunt extrem de comune din cauza mecanismelor legale destinate protejării intereselor dezvoltării investițiilor în timpul tranziției la o nouă ordonanță implementată.

Prin prototipul **CONSIT** [92], [93], [94] s-a implementat un sistem care în baza de cunoștințe realizată încorporează atât euristica, judecata cât și experiența experților umani, referitoare la organizarea amplasamentelor (și a șantierelor) de construcții. Problema presupune căutarea unei soluții fezabile pentru un aranjament bidimensional al funcțiunilor (porți și căi de acces, magazii, utilități etc.), unde performanța este în funcție de mai multe criterii, unele chiar necuantificabile. Acest sistem a fost testat prin rezolvarea câtorva probleme specifice și s-a dovedit a furniza soluții satisfăcătoare. CONSIT a fost elaborat și dezvoltat în cadrul *University of Texas (Austin, U.S.A.)*, inițial rulând pe calculatoare *Texas Instrument Explorer* cu ajutorul mediului KEE (*Knowledge Engineering Environment*), care pentru reprezentarea cunoștințelor face uz de reguli, cadre și diverse tehnici de programare orientate obiect.

K. Kähkönen de la Centrul de Cercetări Tehnice ale Finlandei (VTT) situat în Espo, a publicat în 1989 o lucrare, intitulată *A Knowledge Based System Approach to Determine*

Durations of Site Activities, în care prezenta o bază de cunoștințe realizată pe baza sistemului finlandez al fișelor de producție (*Ratu-files*), denumit **RATU-AJ**. Acest sistem era structurat ierarhic, combinând duratele de execuție ale articolelor de construcții situate pe niveluri inferioare în durate de activități situate la niveluri mai înalte. Formele ecranelor standardizate, formularele electronice și utilizarea simbolurilor grafice facilitau în mare măsură utilizarea sistemului. Baza de cunoștințe realizată în cadrul programului RATU-AJ era rezultatul unui proiect pilot de computerizare a fișelor Ratu de producție, pornit încă din 1987 [186]. Fișele în cauză au fost elaborate manual pe parcursul a 15 ani, în 1989 acoperind deja aproape toate activitățile critice în planificarea calendaristică tipică a activităților de construcții. Ele au fost realizate de către specialiștii din cadrul Centrului de Cercetare Tehnică al Finlandei și au fost publicate de către Centrul de Informare în Construcții al Finlandei. Aceste fișe conțin peste o sută de cartele metodice (conținând descrierea metodelor de lucru), o altă sută de cartele cu resurse (conținând unități de timp și rate de materiale), opt cartele cu informații despre costuri și șase cartele cu problemele legate de manipularea materialelor. Informațiile multiple conținute în aceste fișe sunt prezentate în cadrul sistemului RATU-AJ, în trei moduri:

- tabele de calcul (unități de timp și factori de influență),
- grafică bitmap (desene scanate și grafice completate cu explicații),
- reguli cuprinse într-o bază de cunoștințe (metode și cunoștințe pentru estimarea duratelor).

Ca mediu de realizare s-a folosit shell-ul cunoscut NEXPERT destinat realizării sistemelor expert, rulând pe un calculator Apple Macintosh II. Tabelele de calcul conținând datele experimentale despre durate au fost întocmite utilizând Microsoft Excel, iar unele facilități de raportare au fost realizate cu ajutorul limbajului de programare PASCAL. Sistemul este încă utilizat în mod curent în industria finlandeză de construcții, mai ales la planificarea lucrărilor extinse și complexe de cofrare, demonstrând cu succes posibilitatea traducerii informațiilor multiforme integrate într-o singură aplicație, fără a avea de suferit calitatea acestora.

Cercetările și realizările axate pe dezvoltarea instrumentelor și tehnicilor destinate estimării duratelor activităților referitoare la amplasamente și la organizarea șantierelor de construcții pot fi împărțite în următoarele categorii:

- ❶ programe de simulare,
- ❷ rețele stohastice,
- ❸ sisteme expert.

❶ *Programele de simulare* pe calculator imită procesele din lumea reală, folosind ca bază pentru presupuneri un model particular. Primele aplicații de simulare pentru construcții au fost elaborate încă prin anii '60. Cele mai multe erau privite ca "jucării", fiind folosite în principal în context educațional. Prin anii '70 și '80 au apărut foarte multe programe de simulare, inclusiv pentru planificarea și urmărirea proiectelor de construcții. Obiectivul

comun din spatele acestor dezvoltări de programe era în principal desemnarea unor tehnici care să poată produce previziuni corespunzătoare referitoare la modul și la durata de realizare, în ceea ce privește proiectele de construcții. Rezultatele uzuale conțineau duratele activităților, timpii totali, deviații de la termene, costuri și riscuri probabile. Dintre aceste programe merită să amintesc **DYNASTRAT** [46], **PRODUF** [17], **SIREN** [108] și altele. Calculul duratelor activităților în aceste sisteme se baza pe două categorii de informații: durata prevăzută pentru activitate în cazul condițiilor optime și factorii posibili de interferență. În timpul simulării, duratele activităților și interferențele care pot produce prelungirea duratelor erau selectate din distribuțiile relevante ale acestora, folosind numere pseudo-aleatoare.

❷ *Rețelele stohastice* sunt modele ambițioase care încearcă să surprindă majoritatea datelor ce descriu activitățile dintr-un proiect sub forma de distribuții. În același timp sunt și modele care descriu anumite tipuri de dependențe între activitățile de pe șantier (de exemplu efectul de întârziere al unei activități asupra unei alte activități). Diferența principală între programele de simulare și rețelele stohastice constă în faptul că cele din urmă sunt modele probabilistice comprehensive, pe când programele de simulare conțin un anumit grad de simplificare. Modele de rețele stohastice cu aplicabilitate în planificarea din domeniul construcțiilor au fost realizate de către numeroase grupuri de cercetare.

❸ *Sistemele expert* prezintă cele mai bune promisiuni pentru determinarea duratelor activităților de pe șantier, oferind totodată și punctul de plecare pentru construirea sistemelor de baze de cunoștințe. Motivele pentru un asemenea optimism ar putea fi considerate următoarele:

- Bazele de cunoștințe permit aplicarea metodelor sistemice și detaliate de rezolvare într-o modalitate mult mai rapidă decât posibilitățile manuale.
- Bazele de cunoștințe pot oferi asistență în determinarea duratelor activităților de pe șantier.
- Priceperea necesară pentru acoperirea unor lucrări complexe poate fi capturată și preservată în cadrul unei baze de cunoștințe.
- Priceperea odată capturată poate fi analizată și formalizată într-un mod care să permită autodestinarea pentru forme euristice de rezolvare (cu alte cuvinte, elaborarea bazelor de cunoștințe poate fi folosită ca instrument de cercetare și dezvoltare).

MASON este un sistem expert realizat prin anii '80 pentru determinarea duratelor lucrărilor de zidărie. El încorporează estimarea duratelor într-un model ierarhic care poate fi utilizat pentru a produce durate estimate în două variante: estimarea inițială a duratei și estimarea finală a duratei. Estimarea finală se bazează pe un mediu computațional mai detaliat și mai dezvoltat, care cuprinde elemente, ca de pildă factorul de învățare.

Există de asemenea unelte derivate din abordarea dezvoltării sistemelor expert, cum ar fi de exemplu cele pentru furnizarea automată a propunerilor de planuri la proiectele

de construcții. Aceste unelte, pe lângă altele conțin și tehnici pentru estimarea duratelor activităților. În sistemul **PLATFORM I** [123], [124] se iau în considerare factori diferiți de risc în calculul duratelor activităților de construcții. În comparație, calculul duratelor în sistemul **CONSTRUCTION PLANEX** [196] este bazat numai pe ratele medii de productivitate, deși poate ține seama și de orele suplimentare de lucru considerate. **TIME** [86] realizează de asemenea planificări pe baza ratei medii de productivitate. **CONSAS** [101] poate fi folosit și pentru a analiza propunerile de planificare, unul din modulele acestui sistem fiind destinat evaluării acurateței estimării duratelor activităților.

3.3.5 Sisteme pentru management în construcții

În domeniul managementului, J. Hudson, J.P. Murray, R.N. Gameson și B. Toft au prezentat mai multe baze de cunoștințe dezvoltate, precum și diverse aspecte legate de cercetările efectuate la *University of Reading*, publicând lucrarea *Expert Systems for Briefing in the Construction Industry*. Considerând că eșuarea comunicării satisfăcătoare între client (beneficiar) și grupuri profesionale, în stadiile incipiente ale proiectului de construcții, poate conduce la probleme considerabile în stadiile finale ale proiectului și ale terminării construcției, autorii mai sus amintiți au examinat fezabilitatea utilizării bazelor de cunoștințe pentru a ajuta procesul de comunicare. Cercetările amintite aveau drept scop construirea unui model al comunicării în procesul de *briefing*, care să poată fi apoi utilizat ca suport pentru o bază de cunoștințe destinată asistării beneficiarilor și grupelor profesionale în acest proces.

Într-o altă lucrare, publicată în 1989 sub titlul *Knowledge Based System for Recognition of Work Required for Construction*, autorii A. Sirajuddin și M.J. Mawdesley descriu activitățile întreprinse la Departamentul de Construcții din *University of Nottingham*, referitoare la automatizarea producerii planurilor calendaristice și de lucru pentru proiectele de construcții. În acest articol, conturarea metodei date este urmată de discutarea detaliată a modului de elaborare, sarcinile sistemului și evaluarea muncii necesare, pornind de la schițele și desenele de construcții. Este descrisă baza de date utilizată și sunt prezentate rezultatele experimentărilor de-a lungul testării sistemului, ideile inițiale fiind publicate încă din 1987 cu ocazia celui de al 4-lea Simpozion Internațional de Robotică și Inteligență Artificială în Construcții, organizată la Haifa, Izrael [132]. În principiu, lucrarea publicată în 1989 este împărțită în patru părți: descrierea metodei **FEPP** (*Finite Element Project Planning*) care s-a utilizat, descrierea sistemului folosit pentru procesul de recunoaștere al construcției, descrierea rezultatelor experimentale obținute prin utilizarea sistemului și indicații pentru direcțiile viitoare ale investigărilor. Metoda prezentată a fost realizată și implementată sub sistemul de operare GEM în Prospero PASCAL. Sistemul realizat a fost aplicat efectiv atât la proiectul unui pod din beton armat cât și la proiectul unei clădiri cu structura în cadre din beton armat.

"Folosirea microcalculatoarelor în administrarea productivității poate produce beneficii semnificative managerilor de proiecte și estimatorilor din domeniul construcțiilor", au declarat K.D. Lemon și profesorul J. Christian, ambii din New Brunswick (Canada). De

asemenea, cei doi au publicat o lucrare intitulată *A Micro-computer Application for Managing Construction Productivity* în care au prezentat dezvoltarea și utilizarea unei aplicații pentru prezicerea, administrarea și monitorizarea productivității pe șantiere, folosind opiniile experte ale personalului angajat în activitățile de construcții. Au prezentat și opiniile lor legate de introducerea unei metode inovative pentru calculul indicelui global de productivitate pe șantiere și pentru determinarea zonelor critice. Sistemul propus de ei a fost implementat în 1989 folosind pachetul de programe LOTUS 123, rezultând un sistem decizional relativ simplu și totuși puternic și eficient. Metodele folosite în această aplicație au implementat atât structuri de tip *Command Language* cât și macro-uri de tastatură. Sistemul rezultat a fost botezat **PRODUCTIVITY EXPERT**, fiind un sistem decizional interactiv bazat pe tabele de calcul cuplate la o bază de cunoștințe. În dorința de a realiza un sistem care reproduce operațiile unui sistem expert, PRODUCTIVITY EXPERT conținea o secțiune de adunare a informațiilor, facilități explicatorii și o bază de date expandabilă. Sistemul era livrat, ca pachet comercial, pe o singură dischetă de 5,25" necesitând aproximativ 80 kb RAM, având și avantajul că pentru utilizare nu necesita cunoștințe extinse sau experiență în folosirea microcalculatoarelor.

T.C. Chang și C.W. Ibbs (*Department of Civil Engineering, University of California, Berkley*) autorii lucrării *A Fuzzy Expert System for Priority Ranking in Network Resource Allocation* au prezentat, cu ocazia conferinței CIVIL-COMP din 1989, un sistem expert demonstrativ din domeniul construcțiilor, denumit **PRIORITY RANKING**, destinat rezolvării priorității deciziilor în planificarea resurselor. Sistemul folosea preponderent variabile lingvistice pentru stabilirea gradelor de incertitudine ale factorilor externi (cum ar fi cele referitoare la probabilitatea unor ploi torențiale, temperatura probabilă, etc.) și ale parametrilor interni (felul critic al activității, mobilitatea totală etc.). În absența informațiilor necesare, sistemul folosea valori implicite, cum ar fi presupunerea că totul este *normal*. În procesul alocării resurselor un aspect foarte important era reprezentat de către determinarea priorității de ordonare pentru toate activitățile desemnate. După cum s-a demonstrat și în lucrarea amintită, teoria probabilității și raționarea incertă (fuzzy) pot fi folosite cu succes pentru stabilirea priorităților și a ordonării. Conceptul de bază pornește de la următoarele considerente:

- C_i reprezintă mulțimea factorilor externi,
- w_i este ponderea ca valoare lingvistică a factorului extern C_i considerat ca și criteriu, iar
- s_{ij} este susceptibilitatea activității j la criteriul C_i .

În acest sistem, ponderea w_i este determinată prin teoria probabilității [58], iar susceptibilitatea s_{ij} rezultă aplicând conceptul raționării incerte [48]. Dacă w_i și s_{ij} sunt deja cunoscute, atunci prioritatea activității j (notată cu P_j) poate fi determinată cu următoarea formulă

$$P_i = \frac{\sum_{i=1}^n (1 - s_{ij} \cdot w_i)}{n}$$

unde termenul $(1 - s_{ij} \cdot w_i)$ poate fi interpretat ca și măsura priorității pentru activitatea j împotriva criteriului C_i . O valoare mare a lui P_j indică faptul că activitatea j este puțin susceptibilă față de criteriul extern, deci are o prioritate ridicată. Regulile descrise în sistemul PRIORITY RANKING aveau forma generală

Dacă X este F_i Atunci Y este G_i

unde X este numele criteriului (*ploaie, temperatură, condiții de schimbare, ordine*

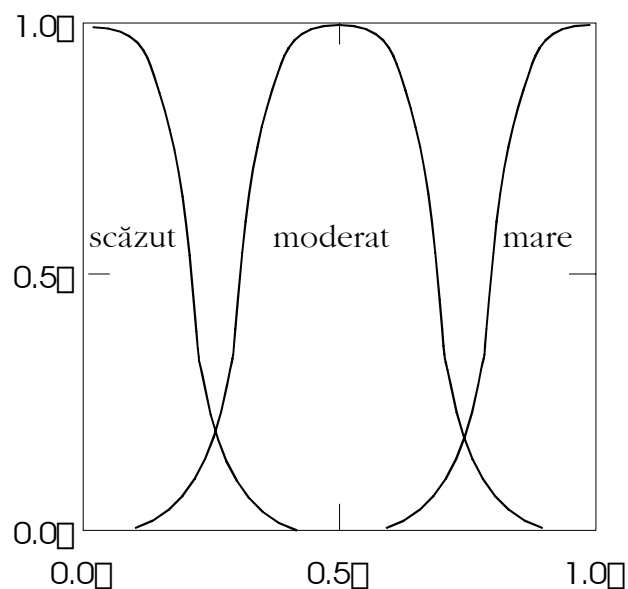


Fig.8 Funcții de apartenență ale seturilor fuzzy

schimbată etc.), Y este o afirmație în consecvență (*recomandarea folosirii ploii ca și criteriu, recomandarea folosirii temperaturii ca și criteriu etc.*), iar F_i și G_i sunt seturi incerte (termeni lingvistici, cum ar fi: *scăzut, moderat, ridicat, important, foarte puternic etc.*). Un set de incertitudini este o mulțime generalizată căreia îi poate aparține un element, această relație fiind descrisă de către o funcție de apartenență. Figura 8 ilustrează un asemenea exemplu, cu forma funcțiilor de apartenență pentru seturile incerte: *scăzut, moderat, mare*. Nu trebuie însă uitat că aceste reguli pot fi specificate sub formă generală pentru toate proiectele, dar pot fi calibrate și pentru specificul unui proiect individual în concordanță cu diversele

caracteristici particulare ale proiectului. Privind partea antecedentă a regulii amintite (adică " X este F_i ") prin prisma următorului exemplu

dacă se cunoaște că ploaia este *foarte puternică*,

atunci, folosind logica incertă, se poate deduce o consecință de genul

este *foarte recomandat* ca ploaia să se considere un criteriu.

Totuși pentru asemenea evenimente, ca ploaie și temperatură, probabilitatea ocurenței este inerentă (cu severitatea corespunzătoare). De exemplu, are mai mult sens să afirmăm că: o ploaie *foarte puternică* va fi *probabilă* săptămâna viitoare, decât să spunem că: ploaia va fi *foarte puternică* săptămâna viitoare. Pentru exemplul considerat se pot stabili valorile ocurenței din experiența observațiilor meteorologice. Astfel, în general, aceste tipuri de propoziții vor avea forma

X este F_i este p_i

unde p_i este o valoare lingvistică a incertitudinii (probabil, improbabil, puțin probabil, foarte probabil etc.), iar X poate fi criteriul considerat pentru alocarea resurselor, sau

orice altceva, a cărei ocurență este inerentă cu incertitudinea. Conform literaturii de specialitate, aceste două tipuri de expresii alcătuiesc un corp de evidență de tipul II, altfel spus o colecție de propoziții incerte

$$E = \{g_1, g_2, \dots, g_n, g'_1, g'_2, \dots, g'_n\}$$

unde g_i și g'_i (iar $i=1, 2, \dots, n$) sunt denumite *granule* și sunt definite după cum urmează

$$g_i = \text{dacă } X \text{ este } F_i \text{ atunci } Y \text{ este } G_i \quad \text{respectiv } g'_i = X \text{ este } F_i \text{ este } p_i$$

De aici se poate ajunge la următoarea expresie

$$Y \text{ este } Q \text{ este } ?\alpha$$

sau echivalent:

Care este gradul de posibilitate ca Y să fie Q ?

Cât este gradul de incertitudine ca Y să fie Q ?

În aceste expresii Q reprezintă un subset *fuzzy* iar $?\alpha$ este o valoare lingvistică de probabilitate dorită, gradul de incertitudine fiind egal cu 1 minus gradul de posibilitate al lui " Y este Q " (Q' fiind complementul lui Q). Subsetul Q poate fi privit și ca un *nivel al dorințelor*. Această abordare ar permite stabilirea unei metode de măsurare a ponderilor și stabilirea unui nivel de referință. Chang a elaborat și a publicat [48] un algoritm eficient pentru rezolvarea gradului de posibilitate, pentru orice nivel dorit. Acest algoritm a fost încorporat în sistemul expert PRIORITY RANKING. Odată cu calcularea gradului de posibilitate, pentru lista criteriilor externe, sunt obținute și ponderile acestor criterii. Sistemul prezentat a fost realizat cu ajutorul limbajului PROLOG, clauzele având forma

rule(criteriu, F_i , G_i)

rule(identificator_criteriu, pas_de_înlănțuire, A_i , B_i)

pentru ponderile w_i , respectiv
pentru susceptibilitatea s_{ij} .

A_i și B_i sunt seturi *fuzzy* în această expresie, definite pentru a interpreta termenii lingvistici ca: *mare*, *moderat*, *scăzut*, iar *rule* (regulă) este un cuvânt cheie folosit de către sistem. Seturile *fuzzy* F_i și G_i sunt la rândul lor rezultatul unor reguli productive *fuzzy*. Fiecare set de probabilitate descrie un concept *fuzzy* particular, care poate fi caracterizat printr-o funcție de apartenență. În sistemul PRIORITY RANKING aceste funcții trebuie definite ca niște clauze, sub forma

fuzzy_set(termen_lingvistic, $m_0, m_1, \dots, m_{10}$)

Într-o asemenea clauză, *fuzzy_set* este un cuvânt cheie folosit de către sistem, iar m_i ($i=0, 1, \dots, 10$) sunt cele 11 grade de apartenență care reprezintă funcția de apartenență. Limbajul PROLOG dispune de un sistem de gestiune pentru baze de date, simplu dar performant, având și un motor de inferență care face uz de strategia *backtracking*. Pentru a permite

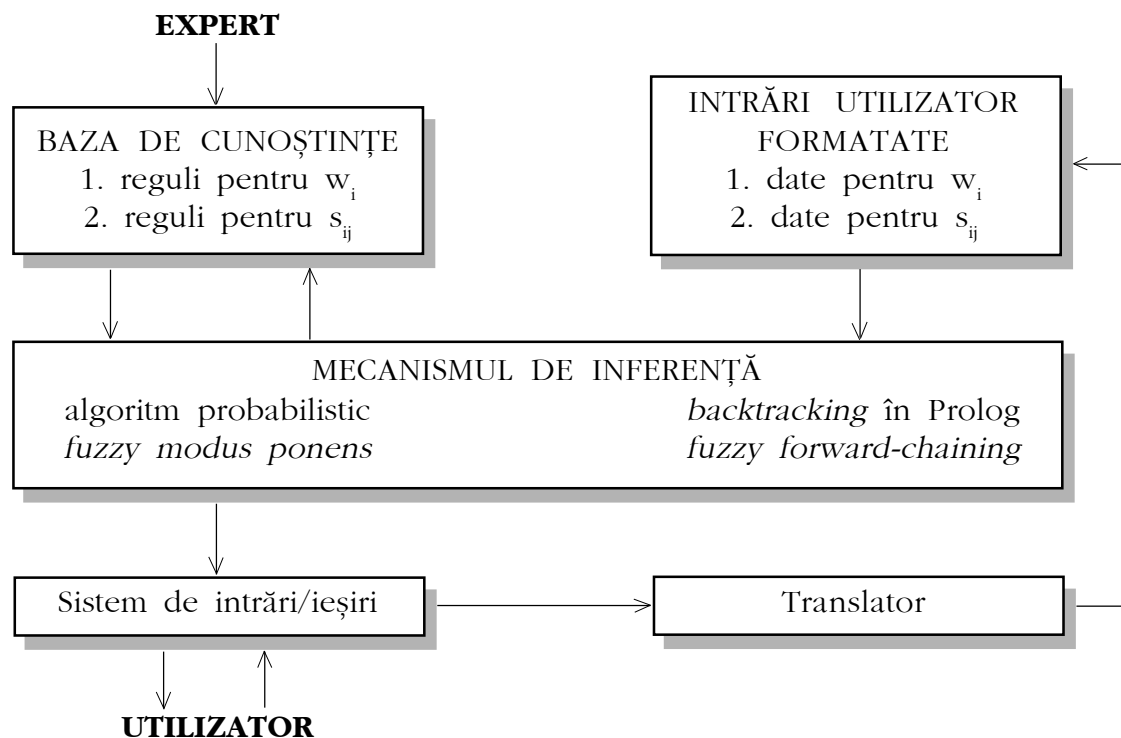


Fig.9 Structura sistemului expert PRIORITY RANKING

raționarea cu noțiuni incerte prin motorul de inferență existent, expresiile *fuzzy* trebuie scrise ca predicate PROLOG. În versiunile sistemului PRIORITY RANKING s-au folosit pe lângă PROLOG și limbajele de programare C și ASSEMBLY, în mare parte pentru scrierea unor proceduri și funcții necesare calculelor matematice din domeniul raționării incerte. Aceste părți au fost integrate în PROLOG sub formă de predicate, permițând astfel mecanismului de inferență înglobat să folosească aceste proceduri și funcții ca pe cele native din cadrul limbajului. Detaliile referitoare la programarea sistemului sunt descrise în [48], iar figura 9 ilustrează structura sistemului amintit. Folosind un calculator IBM XT/286 la 8MHz, obținerea unei valori P_j pentru o activitate j având o bază de cunoștințe cu 7 criterii și 28 de reguli productive *fuzzy*, a necesitat 6...7 secunde. Pentru testarea sistemului s-a considerat proiectul de demolare al unei clădiri vechi care conținea birouri, și construirea în locul acesteia a unei clădiri administrative noi, mai mari, dispunând de un amplasament într-o zonă cu vreme rea, parcurgând și perioada de iarnă. Proiectul abordat conținea 92 activități. În cadrul acestei aplicații a fost necesară și scrierea unui program pentru implementarea algoritmului RALS [48], necesar alocării resurselor. Din utilizarea sistemului au rezultat mai multe variante (fiecare necesitând maxim 40 de muncitori) cu durate de 355, 370, 371 și 378 de zile, în funcție de anotimpul în care ar fi demarate lucrările. Pornirea lucrărilor în primăvară conducea la durata totală cea mai redusă (355 zile), iar începerea lucrărilor în vară (370 zile) conducea la vârfurile cele mai scăzute de resurse necesare, indicând astfel și efectul criteriilor externe considerate (ploaie, temperatură).

În Marea Britanie au fost dezvoltate mai multe proceduri cu scopul de a alătura automat soluțiile de execuție corespunzătoare, proiectelor de construcții considerate finale sau aflate într-o stare de completitudine substanțială, deși problemele privind execuția și proiectarea erau tratate de sisteme de management separate. În lucrarea publicată de către

dr. M. Alshaw (Department of Civil Engineering, Salford University, U.K.) și D. Jagger (School of the Built Environment, Liverpool Polytechnic, U.K.) sub titlul *An Expert System to Assist in Generating and Scheduling Construction Activities*, s-au prezentat aspectele cercetării efectuate pentru testarea fezabilității dezvoltării unei facilități care ar stabili automat soluția de execuție pentru un proiect dat. Cei doi cercetători au reușit în 1989 să elaboreze un prototip destinat generării automate a activităților de construcții în ordinea lor logică și practică, folosindu-se de integrarea tehnologiei sistemelor expert, a bazelor de date și a produselor software pentru administrarea proiectelor. Obiectivele principale urmărite în cercetare au fost:

- elaborarea unei structuri simple de cunoștințe și corespunzător manipulabile, care ar profila cantitatea enormă de cunoștințe nestructurate din construcții,
- deciderea/alegerea celui mai bun mecanism de reprezentare, și
- examinarea fezabilității integrării instrumentelor existente de înmagazinare și de planificare, cu tehnologia sistemelor expert.

Folosind analiza experimentală, Gray a propus în 1986 o metodologie bazată pe reguli [87] pentru selectarea activităților de construcții și pentru generarea predecesorilor acestora. Regulile erau destinate pentru reglementarea procesului de descompunere al activităților, pentru respectarea condițiilor limită în ceea ce privește resursele și pentru stabilirea predecesorilor activităților. Colectivul condus de O'Connor [150] respectiv de Ibbs [101] au descris un sistem de planificare pentru analiza inițială și analiza progresivă a rețelelor. Soluția propusă era un sistem integrat, alcătuit dintr-un mediu pentru sisteme expert, o bază de date și un pachet de administrare proiecte. Cunoștințele din domeniul construcțiilor erau obținute de la contractori/antreprenori, proprietari și persoane academice, iar activitățile erau identificate la nivel de detalii fine. Levitt și Kunz au elaborat sistemul denumit **PLATFORM** încă din 1985 [123], destinat actualizării rețelelor de activități la construirea platformelor marine. Obiectivul principal al acestui sistem era actualizarea rețelei de activități prin alterarea atributelor acestora, cum ar fi durata și influența asupra activității imediat următoare. Duratale erau actualizate pe baza timpilor activităților efectuate, iar activitățile următoare erau selectate din soluții predefinite în cazul activităților cu caracter incert. Navichandra împreună cu alții a propus un sistem pentru generarea rețelelor, considerând un set de activități ca intrare și producând o planificare ca ieșire, prin stabilirea unor precedente între activități, sistemul fiind denumit **GHOST** [145]. Acest sistem pornea presupunând că toate activitățile sunt executate simultan, apoi folosea cunoștințele conținute pentru modificarea rețelei prin introducerea liniarității ori de câte ori activitățile nu puteau fi executate în paralel. Un alt sistem care putea genera activități, precedente, estimări de costuri și rețele a fost dezvoltat de către un colectiv condus de Hendrickson [98]. Ca date de intrare erau necesare specificațiile elementelor, informații despre amplasament și accesibilitatea resurselor. Cunoștințele erau înmagazinate în program sub forma unor tabele de decizie care erau convertite apoi în rețele de cadre.

Pe baza acestor lucrări s-a ajuns la concluzia că sistemele orientate obiect și reprezentările în cadre sunt cele mai potrivite pentru gestionarea eficientă a cunoștințelor. Au fost utilizate

și medii sofisticate, ca ART, KEE și KNOWLEDGE GRAFT, acestea necesitând însă echipamente relativ expensive pentru a putea fi exploatare. Prototipul realizat în 1989 de către Alshawy și Jagger a fost denumit **MIRCI** (*Management Interface for the Construction Industry*) și avea trei componente: un sistem expert, un sistem de înmagazinare a datelor și un administrator de proiecte. Activitățile de construcții erau generate împreună cu precedentele lor de către sistemul expert, utilizând bineînțeles cunoștințe de specialitate din acest domeniu. Informațiile generate erau apoi trimise administratorului de proiecte pentru a permite reprezentarea grafică a activităților planificate sub formă de rețea. Sistemul expert și administratorul de proiecte erau cuplate la o interfață comună pentru facilitarea comunicației. Această interfață conținea și o bază de date care oferea o sursă eficientă pentru transferul și gestiunea informațiilor. Astfel cele trei componente erau integrate în întregime pentru ușurarea trecerii informațiilor între ele, evitând traducerile producătoare de erori și consumatoare de timp. Figura 10 ilustrează cele trei părți ale sistemului MIRCI și fluxurile de informații dintre acestea. Utilizatorii puteau interacționa cu sistemul în orice fază. În etapa în care se făcea apel la sistemul expert încorporat, printr-o interfață prietenoasă puteau fi introduse informații noi referitoare la un proiect particular sau se permitea modificarea informațiilor existente. La nivelul bazei de date se puteau accesa sau insera informații generale, cum ar fi despre tehnologii de construcții, detalii despre resurse etc. În faza de planificare a proiectului se puteau utiliza toate facilitățile oferite de către sistem. Această libertate de manipulare a datelor oferea utilizatorilor oportunitatea de a testa toate soluțiile alternative posibile, atât la începutul planificării cât și în orice fază de rulare. Pentru dezvoltarea sistemului au fost utilizate trei pachete comerciale de programe:

- ❶ LEONARDO și LEVEL 3 (*Creative Logic Limited, Middlesex, England*) ca bază, pentru crearea componentei de sistem expert, acestea conținând un shell hibrid puternic, capabil de a furniza reguli, cadre și grafică interactivă împreună cu un limbaj de programare procedural.
- ❷ DBASE III PLUS (*Ashton-Tate, Maidenhead, England*) pentru faza de înmagazinare, având în vedere popularitatea și răspândirea datorate ușurinței utilizării și limbajului puternic de care dispunea.
- ❸ PERTMASTER ADVANCED (*Pertmaster International, Bradford, England*) ca instrument de planificare, adică administrator de proiecte. Acest pachet era destul de răspândit în Regatul Unit, furnizând facilități de proiectare și dispunând de asemenea de o interfață de utilizare prietenoasă.

SIPEC este un alt sistem, realizat de Kartam și Levitt [106], o încercare de adaptare a planificatorului neliniar independent de domenii (SIPE, realizat în 1984 [192]) în modul de a produce planificări în construcții. Autorii, pentru a înmagazina informațiile despre diferite componente de clădiri, referitoare la diferite părți și funcțiuni, s-au folosit de capacitatea de reprezentare în cadre oferită de SIPE. Instalarea diverselor componente era realizată prin acțiuni (operatori) atașate la cadrele reprezentând componentele. Aceste acțiuni erau generice prin natura lor în așa fel încât repetitivitatea unei construcții modulate era exploatată din plin (de exemplu, acțiunea "do-beam" putea fi atașată oricărei grinzi

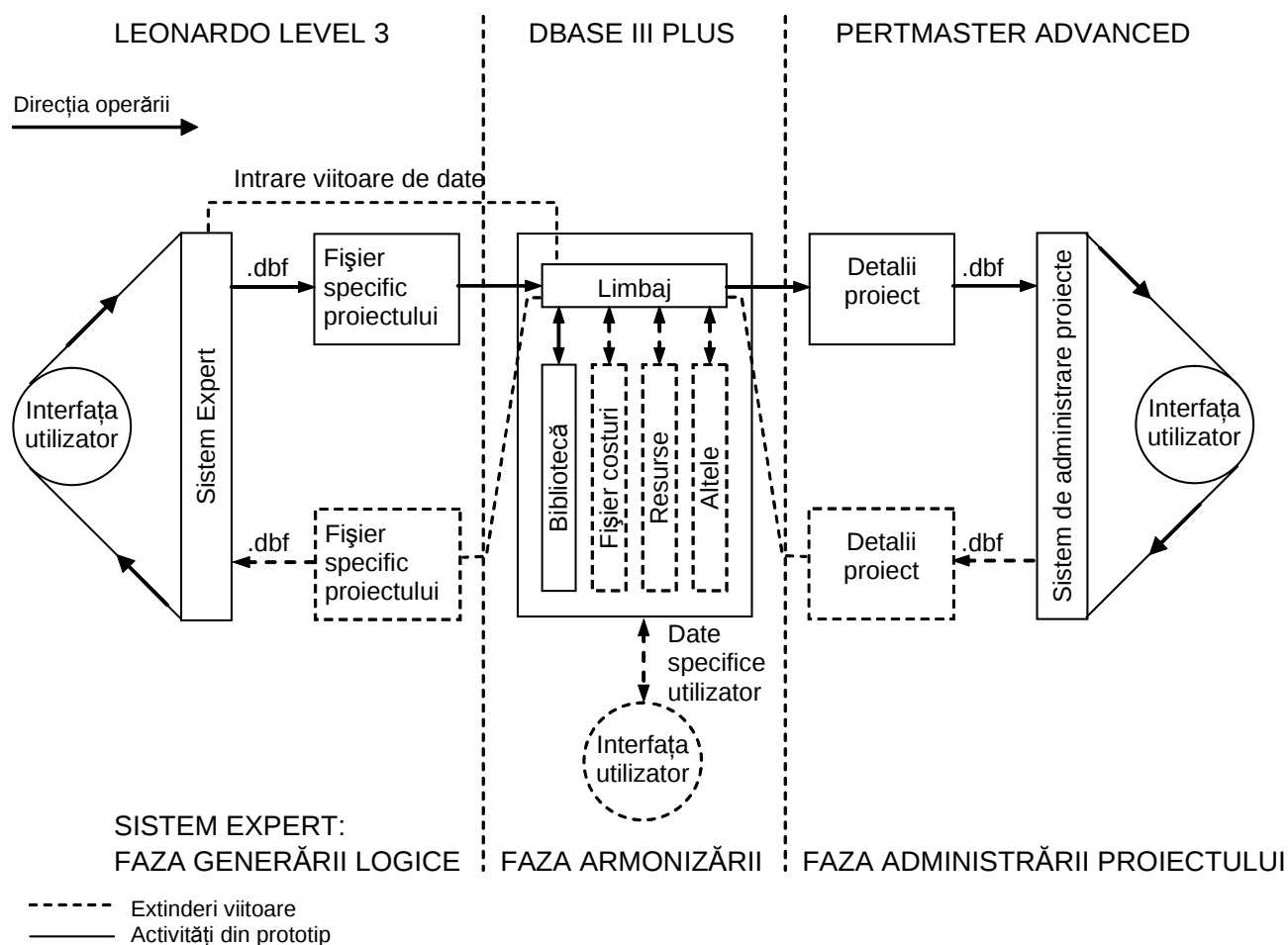


Fig.10 Structura sistemului MIRCI

care se dorea a fi introdusă). Introducerea unei componente de către un operator devenind o activitate, ordonarea activităților era realizată prin propagarea constrângerilor impuse activităților de tehnologia componentelor asociați acestora. În esență era vorba de o ordonare dictată de relațiile fizice dintre componentele obiectului de construcție (de tipurile: "supported-by", "enclosed-by" etc.). Planificarea rezultată din SIPEC arăta un paralelism substanțial al diferitelor activități. Acest fapt se datora ignorării cerințelor din partea resurselor, limitării resurselor și neconsiderării interacțiunilor profesionale pentru ordonarea activităților. Din această cauză sistemul era împiedicat în ceea ce privește efectuarea calculelor duratelor pentru activități.

În lucrarea intitulată *A Conceptual Model for Claim Management in Construction: An AI Approach*, publicată de către N. Riad, D. Arditi și J. Mohammadi (*Department of Civil Engineering, Illinois Institute of Technology, Chicago*) în 1989, s-a prezentat tratarea problematicii disputelor legate de termene și durate în construcții, argumentând avantajele aplicării sistemelor expert cu baze de cunoștințe pentru administrarea și rezolvarea plângerilor. Autorii considerau că problema esențială în rezolvarea conflictelor era generată de lipsa abordării preventive respectiv de lipsa evaluărilor sistematice și comprehensive din punct de vedere tehnic a întregii probleme. Domeniul de cunoaștere al sistemului expert prezentat în lucrarea publicată era îndreptat atât spre analiza diverselor dispute care se nasc din cauza diferitelor întârzieri (considerate scuzabile-compensabile, scuzabile-necompensabile, nescuzabile, independente, concurente, seriale) cât și spre determinarea

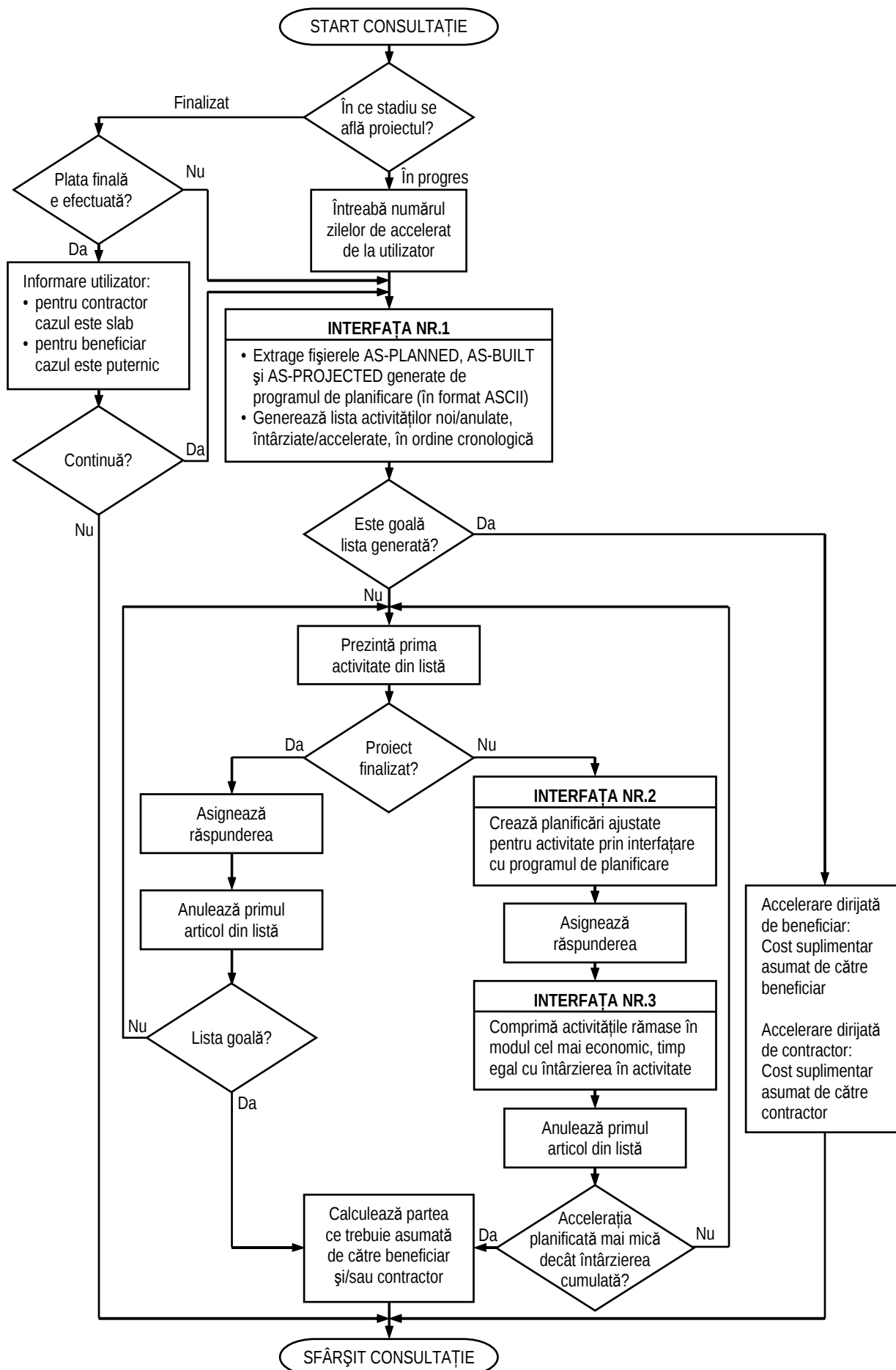


Fig.11 Secvența generală a unei sesiuni interactive propuse de către N. Riad, D. Arditi și J. Mohammadi

responsabilității fiecărei părți. S-a utilizat o procedură numită *Time Impact Analysis* care presupunea folosirea uneltelor de planificare pe bază de rețele, pentru identificarea, cuantificarea și explicarea cauzelor de variație a planificării. Avantajele unui asemenea sistem au fost considerate următoarele:

- Sistemul poate înmagazina un volum mare de date în baza de date de care dispune, permițând în același timp manevrarea și analizarea acestora într-un timp redus.
- Sistemul poate oferi soluții mai eficiente din punct de vedere al costurilor la rezoluțiile disputelor (față de metodele existente și utilizate anterior).
- Utilizatorul poate consulta sistemul după apariția unui conflict, în mod privat. Utilizatorul poate consulta sistemul și înainte de apariția unui conflict, efectuând o analiză *what if*. Acesta garantează confidențialitatea informațiilor prezentate, opinia expertului fiind disponibilă oricând, în caz de nevoie.
- Sistemul se comportă imparțial (neutru) și nu favorizează nici una din părțile implicate în dispută (cum s-ar putea întâmpla în cazul unui mediator uman).
- Sistemul vine în ajutorul personalului de pe teren, ajutându-l să surprindă faptele și implicațiile modificărilor propuse la contract, precum și drepturile acordate de lege.
- Sistemul poate fi extins continuu prin completarea bazei de cunoștințe, folosind și cunoștințele utilizatorilor săi, câștigând astfel mai multă experiență în timp (după cum se prezintă și în [128]).
- Disputele se vor rezolva într-un timp mai scurt folosind sistemul expert.
- Sistemul furnizează o metodă inexpensivă pentru transferul cunoștințelor și pentru specializarea constructorilor practicanți în domeniul juridic.

Sistemul propus a fost realizat sub mediul PERSONAL CONSULTANT PLUS (Texas Instruments). Alegerea acestui mediu s-a făcut pe baza considerentelor prezentate în [23]. Sistemul expert realizat poate interacționa cu DBASE III, permite rularea unor programe scrise în limbajele C, PASCAL și FORTRAN. De asemenea, s-a elaborat și un set de programe scrise în LISP pentru interfațarea ilustrată în Figura 11, iar prototipul bazei de cunoștințe a fost realizat sub formă hierarhică arborescentă.

3.3.6 Sisteme pentru proiectarea construcțiilor

În domeniul proiectării construcțiilor, o lucrare foarte interesantă este cea prezentată de către M.A. Rosenman și Prof. J.S. Gero (*Design Computing Unit, Department of Architectural Science, University of Sidney, Australia*), cu titlul: *SOLAREXPERT - An Expert System for Evaluating Passive Solar Energy Designs*. Proiectul realizat a fost sprijinit substanțial de către Consiliul Național de Cercetare, Dezvoltare și Demonstrare Energetică administrată de către Departamentul Industriilor Primare și Energiei din "Commonwealth". Sistemul **SOLAREXPERT** rulează pe stații de lucru Sun (SPARC) sub sistemul de operare Unix, fiind compus din următoarele module:

- ❶ Programul principal de control, scris în C folosind SunView (pe lângă asigurarea legăturilor dintre celelalte module permite și accesarea acestora cu ajutorul unor pictograme, conține meniul principal).
- ❷ *Shell*-ul pentru sisteme expert BUILD [164] realizat în Quintus Prolog 2.0 (un mediu de dezvoltare bazat pe reguli – deși restricționate ca formă aceste reguli sunt foarte apropiate de limbajul natural englez). Quintus Prolog este un limbaj care permite apeluri către funcții externe (străine) și care poate interfața sistemul Unix și implicit limbajul C. BUILD este activat prin apeluri lansate din meniul principal, fiind capabil la rândul lui să apeleze interfețele grafice sau alte programe procedurale atunci când este nevoie. Este un sistem axat pe obiecte și permite atât înlănțuiri înapoi cât și înainte, dispunând și de facilități explicative. În plus, acest mediu dispune și de câteva capacități de raționare ipotetică, permițând întrebări/presupuneri "*what if*" pentru explorarea posibilităților alternative.
- ❸ Baza de cunoștințe, sub forma unei colecții inițiale de aproximativ 135 reguli în formatul acceptat de mediul BUILD, permițând utilizatorilor să verifice pe rând proiectarea fiecărei zone/părți din clădirea considerată.
- ❹ Interfețele grafice interactive, la început numai două ca număr, putând fi apelate din meniul principal sau direct din mediul BUILD. Prima era destinată localizării, cu ajutorul unei hărți Australiene (fiind activat din meniul principal prin selectarea pictogramei care reprezintă harta Australiei), având la dispoziție multiple caracteristici marcate pe harta electronică (temperaturi, gradul de acoperire cu nori etc.). A doua interfață (activată prin selectarea iconului de casuță din meniul principal) era destinată introducerii informațiilor referitoare la clădire și amplasament. Permitea trasarea granițelor amplasamentului, definirea orientării, precizarea copacilor și ale obstrucțiilor liniare (clădiri vecine, garduri etc.), definirea zonelor din clădirile considerate (camere, dormitoare, zone de serviciu/anexe, zone umbrite/verande etc.). În fereastra care conține attributele zonei curente, se pot specifica detalii referitoare la adâncimea/lățimea zonei, materialele/culorile fiecărui perete, pardoseală, tavan etc. Există patru tipuri de copaci predefiniți, fiecare putând fi modificat grafic sau prin panoul cu attributele corespunzătoare (diametru, înălțime, foliaj, caracter etc.).
- ❺ Programe procedurale (scrise în C) atașate modulului de amplasament, permițând vizualizarea umbrelor pe amplasament (din copaci și obstrucții) între diverse intervale orare și în diverse anotimpuri. Descrierea detaliată a acestor facilități se găsește în manualul de utilizare [165].
- ❻ Programe utilitare realizând chestionări în stil *pop-up*, facilități de ajutor, menținerea unui fișier de urmărire cu conținutul dialogului realizat de către utilizator în sistem.

3.3.7 Sisteme pentru analiza structurală

Trecând la sistemele expert din domeniul analizei structurale, în lucrarea *Adaptive*

Connections of Multitype Structural Descriptors in a TLS for Structural Analysis (autori: R. Fruchter și J. Gluck, *Department of Civil Engineering, Technion-Israel Institute of Technology, Haifa*) publicată în 1989 s-a prezentat un model pentru conexiuni adaptive între descriptorii structurali în cadrul unui sistem de instruire/învățare (TLS: *Training and Learning System*). Sistemul realizat de autorii susmenționați dispunea de o arhitectură ierarhică alcătuită din module "inteligente", destinate modelării structurilor. În prima versiune implementată a sistemului, cunoștințele care legau nivelele de abstractizare erau organizate în două stive: una a cunoștințelor euristice generale și o alta a cunoștințelor specifice de caz. Ultima era reprezentată de descriptorii structurali și de baza de cunoștințe instructive corespunzătoare (TKB: *Training Knowledge Base*). În lucrarea publicată au fost tratate următoarele probleme:

- reprezentarea gnosologică a diferitelor tipuri de descriptorii structurali
- achiziționarea cunoștințelor după tipurile descriptorilor – sub forma regulilor de instruire (TR: *Training Rules*),
- învățarea/însușirea conexiunilor dintre regulile de instruire descriptivă structurală de la un nivel specific de abstractizare,
- înțelegerea și modelarea unui mecanism pentru conexiuni adaptive între multiplele tipuri de reguli de instruire descriptivă structurală,
- efectele acestor conexiuni asupra răspunsului structural și traducerea lor în reguli de instruire noi.

Scopul major al proceselor de instruire și de învățare din sistemul amintit era focusarea pe modurile critice ale descriptorilor structurali. Instruirea/învățarea asista procesul de analiză la nivelul modulului structural, dar și la nivelul regulilor de instruire. Trăsăturile principale ale sistemului erau reprezentate atât prin capacitatea de îmbunătățire a cunoașterii într-un mediu dinamic cât și prin abilitatea acestui sistem de a determina, abstractiza, aminti și folosi conexiunile dintre multiplele tipuri de reguli de instruire descriptivă structurală, la un nivel de abstractizare specific. Conexiunile urmau să se adapteze în funcție de cerințele schimbătoare ale stării analizei structurale. Dezvoltarea acestui model era destinat pentru îmbunătățirea performanțelor analizei structurale și a sistemelor de autoinstruire/învățare. Procesul de analiză trecea prin nivelele de abstractizare ale ierarhiei structurale din cadrul sistemului, stagiile de procesare ale informațiilor fiind următoarele:

- Dacă cunoștințele specifice cazului la nivelul structural curent sunt insuficiente pentru instruirea referitoare la acest model structural, procesarea va fi ghidată de cunoștințele generale euristice. Rezultatul va consta din răspunsul modulului structural și prin noile reguli de instruire descriptivă care vor fi stocate la acel nivel.
- Dacă cunoașterea specifică de caz al modulului structural curent conține reguli de instruire achiziționate în prealabil care corespund stării date, atunci aceste reguli sunt extrase și procesarea returnează starea de răspuns al modulului structural obținut prin aplicarea regulilor de instruire. În cazul în care regulile

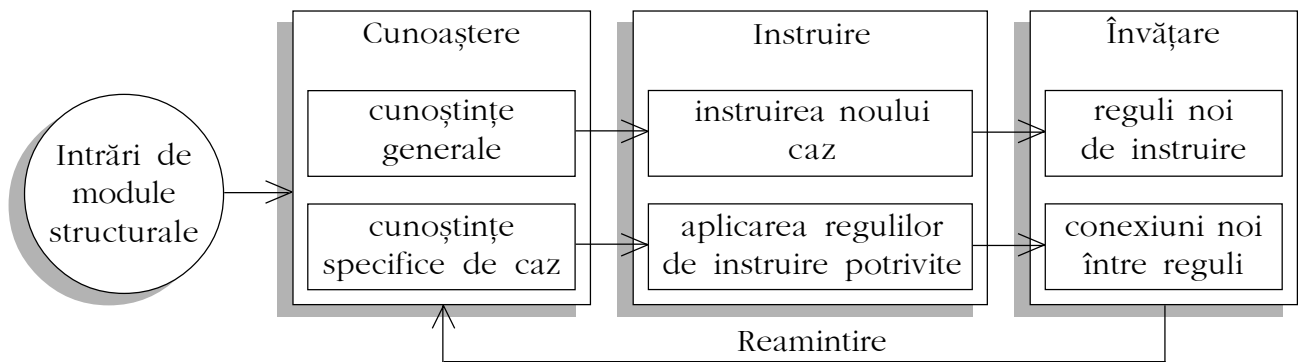


Fig.12 Fluxul de informații la un nivel curent de abstractizare

care se extrag sunt legate de același simptom de răspuns, este determinată o conexiune, abstractizată și reamintită de către mecanismul de legături.

- Regulile de instruire și conexiunile sunt activate dinamic, în funcție de cerințele informaționale. Dacă apar modificări de stări structurale, conexiunile active în prealabil vor fi schimbate adaptiv și răspunsul rezultat va fi unul actualizat.

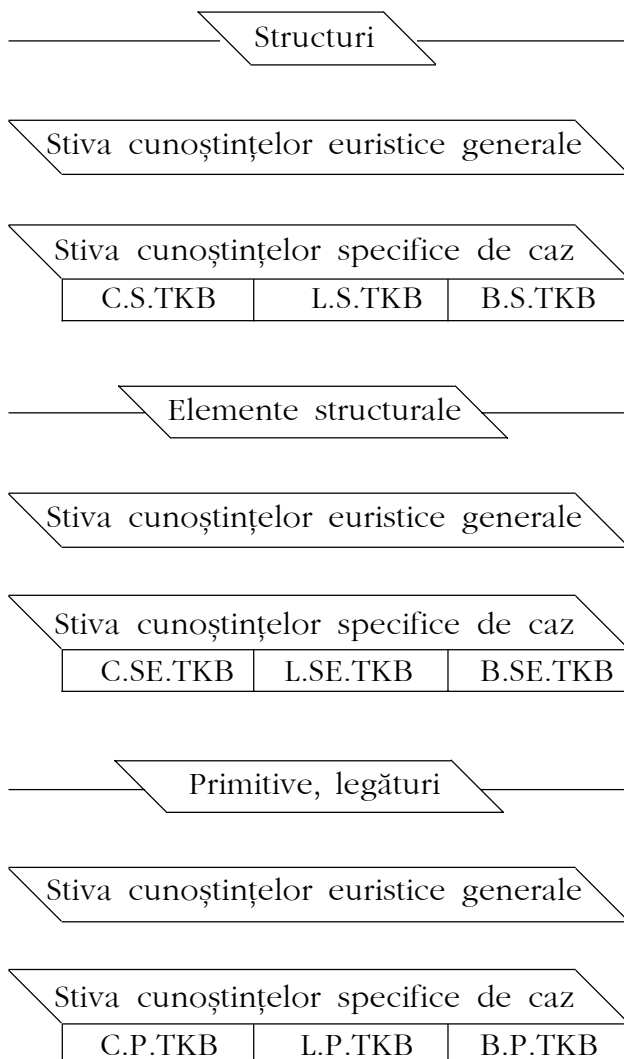


Fig.13 Ansamblul organizat al cunoștințelor în sistemul realizat de Fruchter și Gluck

Fluxul informațiilor la nivelul curent de abstractizare este cel ilustrat în figura 12. Sistemul amintit a fost implementat în CPROLOG la sfârșitul deceniului trecut.

Modelarea unui sistem pentru analiză structurală, conținând atât cunoștințe factice cât și subiective și experiență, implică necesitatea organizării acestor elemente de cunoaștere ca obiecte separate, dar interconectate. Pe baza modelului de abstractizare prezentat în [119], în sistemul mai sus amintit s-a elaborat o hierarhie structurală de module "inteligente". Aceasta se compune din trei niveluri de abstractizare: *structuri*, *elemente structurale*, *primitive și legături*. Cele două din urmă reprezintă nivelul cel mai scăzut de abstractizare, conținând informații fundamentale (fizice și de comportament) necesare pentru realizarea analizei structurale. Modulul *legătură* este elementul fundamental care oferă noțiunile pentru combinarea *primitivei* și pentru transferul informațiilor de la o componentă la alta. Aceste două module amintite (*primitive* și *legături*) sunt considerate elementele de bază în alcătuirea ca modul al nivelului următor: *elemente*

structurale. El poate fi utilizat în continuare pentru modelarea sistemelor mai complexe la nivelul superior *structuri*. Procesul de analiză a modulelor structurale se bazează pe principiile și regulile staticii și dinamicii construcțiilor. Cunoștințele de domeniu, care cuplează nivelurile superioare cu cele inferioare, sunt organizate în două stive: stiva *cunoștințelor euristice generale* și stiva *cunoștințelor specifice de caz*. Organizarea acestor cunoștințe este ilustrată în figura 13. Stiva cunoștințelor euristice generale poate fi reprezentată la diferite niveluri de abstractizare prin reguli, după cum urmează:

- La nivelul superior, denumit *Structură (S)*: reguli pentru combinarea elementelor structurale, reguli pentru obținerea răspunsului structural, reguli generale de instruire și învățare pentru focusarea asupra modurilor critice ale descriptorilor de structuri.
- La nivelul următor, de *Elemente structurale (SE)*: reguli pentru combinarea primitivelor și legăturilor, reguli pentru obținerea răspunsului elementului structural, reguli generale de instruire și învățare pentru focusarea asupra modurilor critice ale descriptorilor elementelor structurale.
- La nivelul denumit *Primitive și legături (P)*: reguli pentru analiza primitivelor și a legăturilor, fapte reprezentând parametri fizici, reguli generale de instruire și învățare pentru focusarea asupra modurilor critice ale primitivelor și legăturilor.

Stiva cunoașterii specifice de caz este constituită din *baze de cunoștințe instructive (TKB)* reprezentând cei trei descriptori de structură propuși (prezențați în figura 13): *configurația (C)*, *comportamentul (B)* și *încărcarea (L)*. Referitor la procesul interpretării analizei structurale, baza de cunoștințe are următoarele caracteristici evolutive:

- *transformarea* (instruirea) informațiilor de intrare și/sau ieșire în scopul evidențierii problemelor și a modurilor critice.
- *învățarea* prin achiziționarea și rafinarea cunoștințelor.

Aceste trăsături evidențiază faptul că nici o bază de cunoștințe raționale nu poate fi completă și nu poate rămâne neschimbată. Din această cauză, arhitectura unui sistem - dorit a fi realizat pe bază de cunoștințe - trebuie să conțină mecanisme de instruire și învățare. Aceste concepte au fost conturate în versiunea precedentă a sistemului prezentat [69]. Ideea de bază a fost instruirea șabloanelor de intrare prin aplicarea unor reguli de instruire/învățare rezultatelor de ieșire formalizate. În versiunea prezentată în 1989 au fost definite noi asigări pentru procesele de instruire și învățare, acestea asistând analiza structurală și interpretarea la fiecare nivel.

Într-o altă lucrare, intitulată *An Object Oriented Approach to Structural Analysis and Design* (autor: G. R. Miller, *Department of Civil Engineering, University of Washington, Seattle, S.U.A.*), se prezintă o aplicație practică orientată pe obiecte, în cadrul dezvoltării softului destinat ingineriei structurilor. Lucrarea este fundamentată pe o realizare anterioară finanțată de către Fundația Națională de Știință din SUA și de către Symbolics Inc.,

(Cambridge, Massachusetts). Programele de calculator existente în acest domeniu sunt bazate în majoritate pe metode și perspective care s-au schimbat foarte puțin față de vremea când normele obișnuite de interacțiune cu calculatoarele erau execuțiile în pachete (rulările grupate), deși acest aspect de multe ori este ascuns utilizatorilor prin căile programelor de pre- și postprocesare. De exemplu, una din ipotezele fundamentale tradiționale referitoare la derularea analizei structurale este aceea că analiza se efectuează pe obiecte bine definite, în esență statice. Astfel, cele mai multe pachete de programe sunt scrise în limbaje de programare și stiluri corespunzătoare acestui tip de calcul. Analiza însăși rămâne o sarcină *batch*, utilizatorul fiind servit prin module atașate de pre- și postprocesare. În contextul proiectării însă, cel implicat are de a face în mod tipic cu date inerent dinamice, în evoluare, ce modelează obiecte care dispun de multe atribute în afara celor pur numerice. Obișnuința în tratarea acestor cazuri era în general cuplarea unor programe dezvoltate individual, obținerea unei integrări într-o oarecare măsură prin comunicare limitată sau indirectă, folosind date partajate [194]. Însă această integrare este închisă granularității, distribuției (în plaja timpului și a procesoarelor), modularității și perspectivelor fundamentale a ceea ce a existat tradițional în locul a ceea ce trebuia sau putea să existe.

Nu este nici un motiv pentru care să credem că cea mai bună cale pentru realizarea mai multor sarcini împreună este aceeași cu realizarea independentă a sarcinilor. Luând în considerare aceste observații, probabil nu ar fi lipsită de temei reconsiderarea problematicii, eventual chiar repornirea în acest domeniu cu un statut curat, dezvoltând metode și instrumente integrate fundamentale, care să permită o utilizare mai bună și mai directă a tehnologiilor mai noi din domeniul hard și soft.

Nu există un consens general referitor la definirea completă a noțiunii de programare orientată obiect, deși în [191] există o schemă interesantă pentru definire formală și o tratare clară a căilor de realizare, publicată încă din 1987. Aspecte ale programării orientate obiect au început să devină din ce în ce mai comune într-o varietate a limbajelor de programare. Față de alte perspective de calcul, care includ: *programarea procedurală* (în care un program este privit ca un set de algoritmi, de cele mai multe ori secvențial), *programarea funcțională* (în care un program este privit ca un flux de date printr-o rețea de noduri funcții), *programarea logică* (sau orientată pe constrângeri, în care un program este privit ca o mulțime de relații sau constrângeri ce se manifestă între intrări și ieșiri), *programarea orientată obiect* poate fi considerată (în ceea ce privește un program) și ca un model analog al unui fenomen preluat din lumea reală. Un asemenea program poate alcătui un mediu care conține obiecte (cum ar fi: grinzi, planșee, stâlpi) cu atribute analoge componentelor reale (cum ar fi: greutatea, lungimea, caracteristici mecanice și proprietăți de materiale) și comportamente corespunzătoare (cum ar fi: prezența și efectul greutății proprii, deformarea sub încărcări, deplasări). Mai mult, obiectele pot comunica între ele și cu utilizatorul și pot fi înzestrate cu competențe necesare (chiar *intelență*) pentru realizarea unor cereri. Problema domeniului rezolvitor constă, astfel, din agenți independenți care interacționează și îndeplinesc cereri. Această perspectivă are trăsătura dorită de a permite folosirea modelării naturale a entităților fizice și abstracte pentru

rezolvarea problemei și pentru dezvoltarea programului în cadrul domeniului problemei date.

Una din caracteristicile cele mai pregnante ale programării orientate obiect este capacitatea partajării atributelor și a competențelor, de regulă prin moșteniri de clasă (ca în limbajele asemănătoare cu SIMULA [34]) sau prin delegare [125]. Altfel spus, din obiectele existente pot fi construite noi tipuri de obiecte, acestea obținând capabilitățile și cunoștințele necesare pentru a face tot ce pot și subtipurile, fără a necesita codificare adițională. De exemplu: presupunând că cineva a realizat câteva entități fundamentale grafice (linii, puncte etc.) care sunt capabile de a se autotrasa, mișca, și așa mai departe, atunci în elaborarea unui element structural pentru analiză, cum ar fi o un element de grindă simplă, este posibilă obținerea capacităților și informațiilor geometrice necesare pentru reprezentarea schematică prin moștenirea lor directă de la tipul liniei. Acesta se poate duce mai departe, se poate considera elaborarea elementelor următoare, cum ar fi grinzi cu secțiunea profilată, care ar putea moșteni atât caracteristicile geometrice cât și o proporție considerabilă a caracteristicilor structurale de la elementul tip grindă structurală simplă (elementară). În scopuri ce țin de proiectare, se pot imagina obiecte componente variate care își cunosc proprietățile secționale detaliate și care știu cum să se autoverifice pentru compleanța codificării. Prin construirea unui obiect adițional, care moștenește atât de la elementele capabile să se autoanalizeze cât și de la elementele care se pot autoproiecta, rezultatul este un membru care se poate autoanaliza și autoproiecta, respectiv i se poate cere acest lucru convenabil oricând și oriunde. Codul care intră în definirea fiecărui tip de obiect și comportamentele corespunzătoare derivate din acesta pot fi refolosite efectiv prin căile moștenirii. În plus, pentru a oferi conveniențe de programare, această capacitate poate fi exploatată și de către utilizatori și de către un mediu adaptiv dinamic prin oferirea unei extensibilități flexibile mediului obiectului. Această capacitate pentru extensibilitate poate fi foarte utilă, mai ales din punctul de vedere al utilizatorilor, într-o exploatare în cadrul birourilor de proiectare. Există un număr din ce în ce mai mare de limbaje de programare și de extensii ale limbajelor care sunt orientate obiect. Dintre limbajele orientate pur pe obiecte, mai puțin cunoscute dar considerate deja tradiționale, se pot aminti Small-Talk (având diverse versiuni la activ) [88], Trellis/Owl [149], Beta [112] etc. De asemenea există extensii ale limbajului LISP: LOOPS, Flavors [142], Common LISP Object System sau CLOS [37] etc.; extensii la limbaje procedurale cum ar fi C++[178], Objective-C [51] și Object Pascal [167]. Fiecare dintre aceste limbaje/extensii de limbaje are trăsăturile și limitările proprii. Fiind dată această atracție conceptuală pentru abordarea orientată pe obiecte, a existat o mare cantitate de efort destinat elaborării teoriilor și limbajelor de programare concurente și distribuite orientate obiect. Acestea au urmat în general două direcții: definirea tuturor calculelor în termeni de agenți de calcul independenți, asincron-comunicativi, cum ar fi modelul *Actor* [15] realizat în 1973 de către Hewitt; și extinderea prin căi diverse ale sistemelor existente orientate obiect pentru includerea unor capacități de concurență și distribuire.

Modelul **Actor** a contribuit la generarea limbajelor, cum ar fi ACT3 [16] și ABCL/R [189]. Acest model era bazat pe entități care comunicau asincron, denumiți actori, fiind definiți

printr-o adresă de livrare (mail), un comportament și o mulțime de "cunoscuți" către care puteau trimite mesaje [15]. Calculele erau realizate prin crearea actorilor cu comportamentul corespunzător (în general schimbător), aceștia răspundeau mesajelor recepționate conform comportamentului propriu creând noi actori și pasând mesaje, schimbând propriul comportament, și așa mai departe. În acest model, obiectele sunt intențional distincte și independente. Nu există un suport direct pentru moștenirea comportamentului din moment ce acesta necesită o dependență ierarhică între tipurile de obiecte. Pe de altă parte, independența completă a obiectelor corespunde ideal pentru implementări concurente și distribuite. Pentru a include trăsături concurente și distribuite, extinderea sistemelor orientate obiect ar fi trebuit să presupună păstrarea trăsăturilor dorite ale acestor sisteme și în același timp biruirea dificultăților inerent asociate cu aceste trăsături. De exemplu: trăsătura de partajare implică dependența între obiecte, care prezintă imediat constrângeri în termenii distribuirii arbitrare ale obiectelor de-a lungul diferitelor spații de adrese. Pentru a exemplifica câteva limbaje și extensii care conțin asemenea trăsături, se pot aminti Concurrent SmallTalk [197], Distributed SmallTalk [33] Hybrid [147], Parallel Objects [50] și Emerald [105]. În spiritul programării orientate obiect, fiecare din sistemele și extensiile amintite încearcă suportarea concurenței și/sau distribuției *task*-urilor într-un mod cât mai transparent posibil. Atât limbajele cât și extensiile limbajelor orientate obiect care se bazează pe modelul *Actor* sunt în esență, prin natura lor, experimentale. Totuși conceptele și metodele fundamentale pot fi utilizate pentru explorarea diferitelor căi prin care concurența și distribuția poate fi folosită productiv în cadrul aplicațiilor structurale. Deși unele metode algoritmice, dintre cele considerate consacrate, pot fi transcrise într-un mediu orientat pe obiecte, o abordare nouă orientată obiect poate oferi un formalism natural pentru efectuarea activităților tradițional distincte într-un mod concurent, ceea ce se poate dovedi util în domenii ca proiectarea, analiza și planificarea construcțiilor.

Sub titlul *A Model for Parallel Training and Learning in Dynamic Structural Response Analysis* (autori: R. Fruchter și J. Gluck, *Department of Civil Engineering, Technion-Israel Institute of Technology, Haifa*) s-a prezentat un model de instruire și învățare paralelă (PTL) a unor caracteristici de stare, prin analiza răspunsului dinamic al structurii. Trăsătura principală a modelului prezentat este capacitatea de achiziționare de cunoștințe și exercitarea controlului asupra stării caracteristicilor răspunsului dinamic la fiecare pas al analizei. Din punctul de vedere al arhitecturii, sistemul este alcătuit dintr-o ierarhie de module "inteligente" destinate modelării structurilor (asemănătoare cu cea prezentată anterior în figura 13). Prin arhitectura sistemului, precum și prin fluxul de informații se consideră următoarele caracteristici referitoare la răspunsul dinamic al structurilor:

- intrările și ieșirile sunt funcții de timp,
- caracteristicile răspunsului pot avea o istorie (referitoare la stări) și
- schimbările critice de stare ale variatelor caracteristici de răspuns pot să apară la pași de timp diferiți.

În figura 14 este ilustrată o vedere schematică de ansamblu a sistemului. Cunoașterea este reprezentată prin baza de date (BD) și baza de cunoștințe (BC). Modulele de procesare

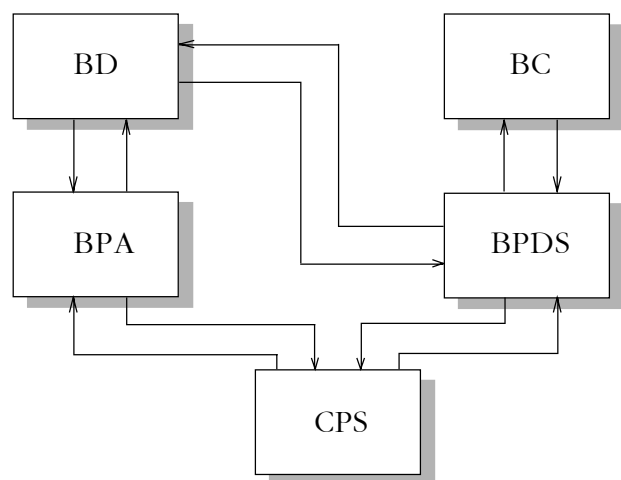


Fig.14

Aspectul general al sistemului

(distincte prin funcțiile lor) sunt următoarele: controlorul pentru procesarea structurii (CPS), blocul pentru procesarea analizei (BPA) și blocul pentru procesarea descriptorilor structurali (BPDS). Starea răspunsului dinamic, la un moment dat al analizei (la pasul t_i), este reprezentată printr-un vector caracteristic. Întreaga istorie a răspunsului de-a lungul studiului de caz constă dintr-un șir al vectorilor caracteristici, instanțiați pentru pașii de timp ceruți. CPS controlează procesarea acestui șir de vectori caracteristici, pas cu pas, BPA realizează actualizarea variabilelor dinamice de stare, iar BPDS procesează

variabilele de stare referitoare la descrierea structurii (așa numiți descriptori structurali). Procesarea informațiilor este realizată în felul următor:

- ❶ Dacă analiza este la pasul inițial de timp t_0 , CPS asignează pentru început task-ul (instruirea și învățarea inițială referitoare la structura dată) la BPDS. După ce BD este actualizată de BPDS, CPS va emite un semnal către BPA și BPDS pentru pornirea prelucrării vectorului caracteristic inițial care a fost creat.
- ❷ Dacă nu se observă de către BPDS modificări critice în starea descriptorilor structurali, atunci componentele vectorului caracteristic sunt procesate în paralel (de către BPDS și BPA) pentru următorul pas de timp și BD este actualizată cu noile valori ale variabilelor dinamice de stare.
- ❸ Dacă se constată o modificare de stare la (una sau mai multe) variabilele cu titlu de descriptorii structurali din vectorul caracteristic, acest lucru se va semnala din BPDS în CPS. Ca urmare CPS va semnala BPA pentru abandonarea procesării în cazul pasului de timp curent considerat. După ce BPDS va actualiza BD cu noua configurație și/sau comportament instruit, CPS va reasigna sarcinile de analiză dinamică pentru BPA.

Fluxul informațiilor va fi continuu pe parcursul analizei răspunsului dinamic al structurii. Cazurile excepționale vor fi numai acelea în care rezultatele instructive obținute la un pas de timp specific vor focusa pe stările critice care conduc la dezvoltarea mecanismului de cedare. În asemenea cazuri analiza se va termina înainte de atingerea sfârșitului scontat.

3.3.8 Sisteme destinate proiectării structurale

În ceea ce privește proiectarea structurală, există de asemenea mai multe realizări semnificative. Într-o lucrare intitulată *A Prolog-Based Representation of Standards for Structural Design*, B.Kumar și B.H.V.Topping (*Department of Civil Engineering, Heriott-Watt University, Edinburgh*) au publicat o schemă de reprezentare bazată pe limbajul

Prolog, destinată procesării automate a normelor din domeniul practicii ingineriei structurale. În cadrul unor încercări similare s-au utilizat, în majoritatea cazurilor, reprezentări pe bază de reguli, pentru resursele cu caracter normativ în programele destinate asistării proiectării. În lucrarea amintită însă, autorii au propus reprezentarea normelor cu aplicabilitate practică ca fapte și procesarea lor prin utilizarea unor reguli generice standard-independente. Deoarece structura de reprezentare concepută se poate asimila cu conceptele și cu componentele fundamentale din limbajul de programare Prolog [49], în mod firesc s-a optat pentru utilizarea acestuia. Pentru varianta demonstrativă s-au folosit normele britanice din domeniul structurilor metalice (BS5950). Reprezentarea standardelor ca fapte s-a dovedit un pas înainte în reprezentarea cunoștințelor și generarea sistemelor expert destinate construcțiilor, fără a necesita cunoștințe din domeniul limbajelor de programare. Normele reprezentate ca fapte pot fi manipulate foarte ușor de către un set de reguli generale standard-independente. De fapt, programarea în Prolog implică numai manipularea faptelor de către o mulțime de reguli. Ideea din spatele acestui tip de reprezentare și manipulare ale standardelor este aceeași conceptual cu cea a programării în Prolog. Mai mult, această reprezentare permite propagarea mai ușoară a constrângerilor, între diferite părți ale sistemului. Cum componentele datelor-entități sunt reprezentate explicit în baza de cunoștințe, schimbarea dintr-o dată-entitate poate fi indusă foarte ușor în celelalte. Această trăsătură poate fi utilizată cu succes în întregul proces de proiectare [47]. Fără o astfel de reprezentare explicită a componentelor era foarte dificilă realizarea unei asemenea abordări, cu asemenea facilități. Mulțimea regulilor generice, responsabile de manipularea faptelor, reprezentând standardele, s-a dovedit a fi în general suficientă pentru a fi utilizată cu orice norme sau reglementări reprezentate prin schema propusă. În [113] și [185] se pot găsi detalii referitoare la prototipul bazei de cunoștințe elaborate pe baza schemei prezentate în lucrarea amintită, sistemul purtând denumirea **STAPRO**.

J.Miles și C.J.Moore (*School of Engineering, University of Wales College of Cardiff, Wales*) au tratat problematica utilizării sistemelor expert în domeniul proiectării podurilor. În prototipul elaborat cu ajutorul limbajului LPA Prolog, respectiv cu ajutorul pachetului de instrumente FLEX (destinate dezvoltării sistemelor expert), s-au limitat doar la podurile rutiere. Primul sistem expert realizat în acest domeniu (autori: Welch și Biswas, 1986) prezenta limitări serioase, iar a doua realizare (autori: Burgoyne și Sham, 1987) a luat în considerare doar problemele podurilor prefabricate din beton precomprimat. Cercetările efectuate precum și conceptele care au stat la baza realizării sistemului conceput de către Miles și Moore au fost publicate în lucrarea *An Expert System for the Conceptual Design of Bridges* (1989).

Domeniul complex al proiectării structurale este abordat în lucrarea *Structural Design Using Intelligent Objects*, autor P.N.Payne (*ICAD Engineering Automation Ltd, Crawley, Winchester, England*). Sistemele de proiectare structurală necesită în general un număr semnificativ de *intrări* din partea inginerului (din moment ce au prea puțină *intelență*) pentru a realiza automat proiecte corespunzătoare. În lucrarea amintită se discută principiile și se prezintă elaborarea unui set instrumental de obiecte structurale inteligente

care permit inginerului structurist să-și automatizeze o mare parte din muncă. De asemenea, obiectele permit utilizatorului să controleze comportamentul lor în aplicațiile care necesită expertitate adițională. Sistemul realizat sub numele **ICAD** încorporează:

- un limbaj de proiectare declarativ, orientat pe obiecte și dirijat de cereri, cu moșteniri de tipul "kind-of" și "part-of",
- o interfață grafică (ce se poate exploata cu *mouse*),
- o bază de date relațională cu facilități de chestionare,
- o facilitate de raportare formatată și de specificații textuale.

Limbajul de proiectare permite utilizatorilor definirea sub-obiectelor proprii sub aspect fizic (exemplu: bară, stâlp) și abstract (exemplu: mulțime de calculat). Fiecare dintre aceste obiecte va conține proprii:

- parametri de proiectare și expresiile ce îi definesc,
- părți (fizice sau abstracte),
- lista de intrări și valorile implicite sau o listă a altor obiecte ale căror definiții ar trebui moștenite.

Reprezentările grafice sunt alcătuite din primitivele grafice ale sistemului ICAD. Pentru poziționarea și orientarea relativă ale acestor părți, limbajul încorporat dispune de o varietate de metode convenabile care permit utilizatorului să se concentreze pe aspectele ingineresti (cum ar fi mai degrabă claritatea sau potrivirea decât coordonatele exacte ale unor puncte). Proiectul în sine este reprezentat ca o structură arborescentă, rădăcina

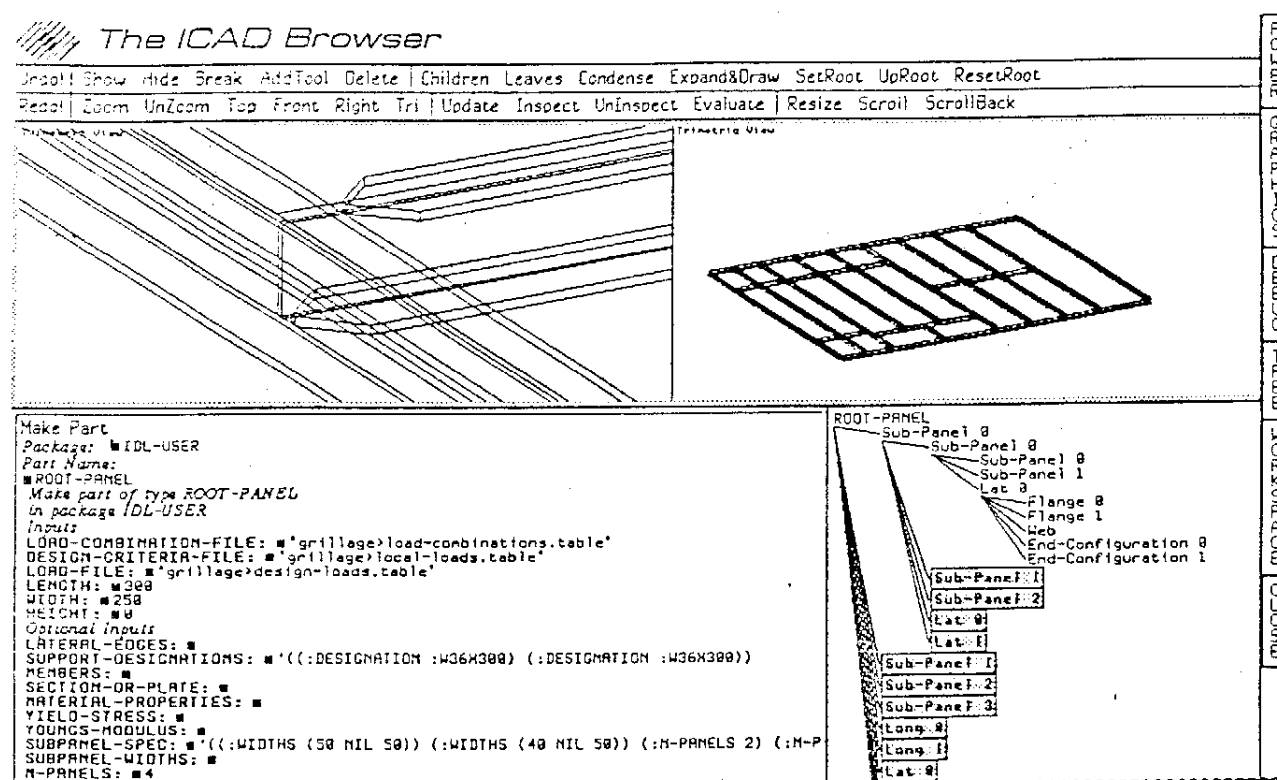


Fig.15 Proiectarea unei rețele de grinzi la un pod metalic prin sistemul ICAD

reprezentând definirea proiectului pe ansamblu. Rădăcina este alcătuită dintr-un număr de părți care sunt definite prin specificațiile părților. Aceste părți au la rândul lor ca și subcomponente alte părți, și așa mai departe, în funcție de complexitatea proiectului. Părțile unui obiect sunt denumite frecvent copii, ilustrând astfel poziția și relațiile lor. În mod similar, obiectul este părintele părților corespunzătoare. Partea nu este în mod necesar un obiect fizic, dar este o entitate conceptuală care aparține părintelui său. De exemplu, diagramele sau geometria alcătuirii pot fi părți utile pentru referiri în cadrul definirilor care permit entității să-și determine parametri proprii de proiectare. În figura 15 este reprodus un ecran din cadrul unei aplicații ICAD având ca subiect realizarea unei rețele de grinzi din cadrul proiectului unui pod metalic. Sistemul permite folosirea mai multor ferestre grafice care pot fi subdivizate în panouri, putând astfel genera sau compara proiecte alternative. Unul din panouri reține istoria comenzilor permițând astfel revenirea secvențială asupra lor (*undo*). Obiectele fizice create pot fi vizualizate sub formă de sită-grafică tridimensională, din orice perspectivă sau proiecție. Afișările pot fi scalate, rotite, translatate etc. Părțile pot fi afișate, evidențiate sau chestionate (cerute să-și identifice numele și parametrii de proiectare) selectiv. Entitățile grafice definite ca pseudo-părți (sau părți abstracte), cum ar fi diagramele de efort, pot fi suprapuse pe proiect sau pot fi afișate în ferestre proprii separate. În ceea ce privește *intrările* și *ieșirile* sistemului, utilizatorii pot trimite specificații de proiectare și printr-o rețea (dacă este cazul), primind de la distanță rezultatele proiectului lor sau ieșiri specifice cerute din acesta. Facilitățile de raportare din ICAD pot fi folosite și pentru specificarea conținutului și al aspectului fișierelor de date folosite ca intrări de către alte programe. Regulile pot folosi utilități soft convenționale pentru rularea altor programe sau pentru comunicarea cu alte calculatoare legate la rețea. Interefețele CAD pot traduce proiectul în fișiere de date cu formate corespunzătoare sistemului destinat.

Într-o altă lucrare, publicată sub titlul *Knowledge-Based Expert System for Damage Assessment Based on Fuzzy Reasoning* de către N.Shiraishi, H.Furuta (Departamentul de construcții, Universitatea din Kyoto), M.Umano (Centrul de calcul, Universitatea Osaka) și K.Kawakami (NTT Corporation) din Japonia, s-a prezentat elaborarea unui sistem expert pentru aprecierea degradării la structurile podurilor. În acest caz, accentul s-a pus pe evaluarea stării de degradare a podurilor realizate din beton armat, din cauza frecvenței de raportare a acestor evenimente [71]. Similar cu alte sisteme expert din acea perioadă, sistemul realizat de către cercetătorii japonezi era constituit din bază de reguli, spațiu de manevră și interpretor [169]. Totuși, față de alte sisteme din acea perioadă, acest sistem se distinge prin trăsătura remarcabilă de a include un sistem practic de manipulare a mulțimilor incerte care putea trata seturi *fuzzy* în procesul de mănuire a datelor, în reprezentarea regulilor și în cadrul procedurilor de inferență. Prin utilizarea acestui sistem exista posibilitatea manipulării mai multor tipuri de incertitudini și ambiguități angrenate inerent în date, reguli și procese, într-o manieră unificată și simplă. Procesul de inferență era realizat conform schemei prezentate în figura 16. Gradele de corespondență dintre date și reguli posibile, înmagazinate în baza de reguli, erau examinate succesiv.

În ansamblu, sistemul se compune din partea de intrări și ieșiri, spațiul de manevră

(memoria de lucru), interpretor (cu mecanismul de inferență) și sistemul de manipulare a mulțimilor incerte, după cum este ilustrat în figura 17. Datele de intrare sunt înmagazinate în spațiul de manevră prin subsistemul de intrări/ieșiri. Subsistemul de intrări/ieșiri a fost realizat pentru facilitarea încărcării datelor. Când cantitatea de date este mare, datele sunt reținute în diferite regiuni din cadrul spațiului adresabil, această diviziune a spațiului adresabil fiind foarte utilă la scurtarea timpului de implementare. Interpretorul lucrează la selectarea regulilor adecvate din baza de reguli prin care se realizează procesul de raționare. Întregul sistem a fost realizat cu ajutorul limbajului LISP și a fost dezvoltat pe o stație de lucru pe 32 biți. Sistemul expert elaborat a fost testat prin evaluarea stării curente a unui pod din beton armat, dat în folosință în anul 1938. Folosind rezultatele investigațiilor de specialitate cât și elementele disponibile referitoare la condițiile de proiectare și la mediul ambiental, au fost estimate mai multe cauze probabile pentru avariile și degradările existente. Pornind de la aceste cauze s-a obținut și modul deteriorării. Pe urmă,

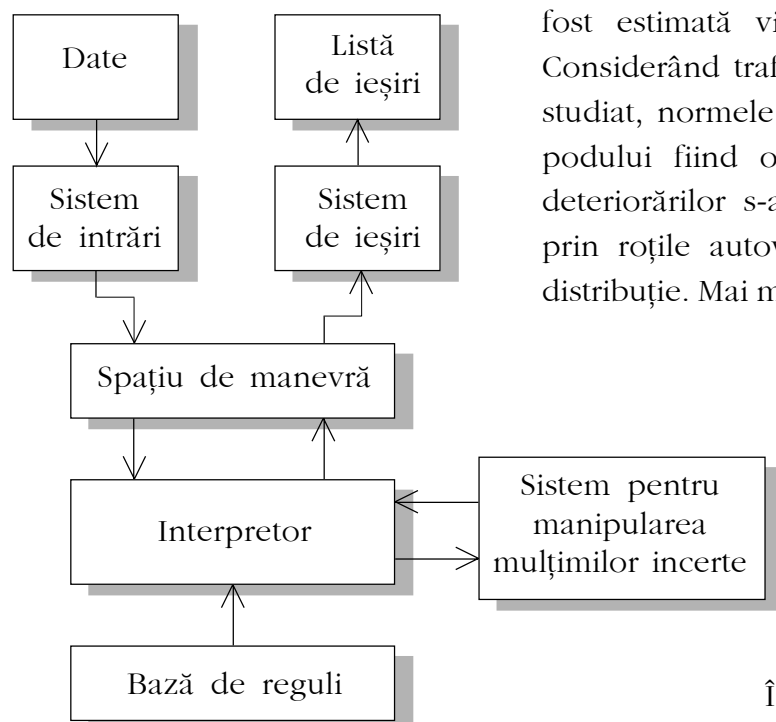


Fig.17 Arhitectura sistemului

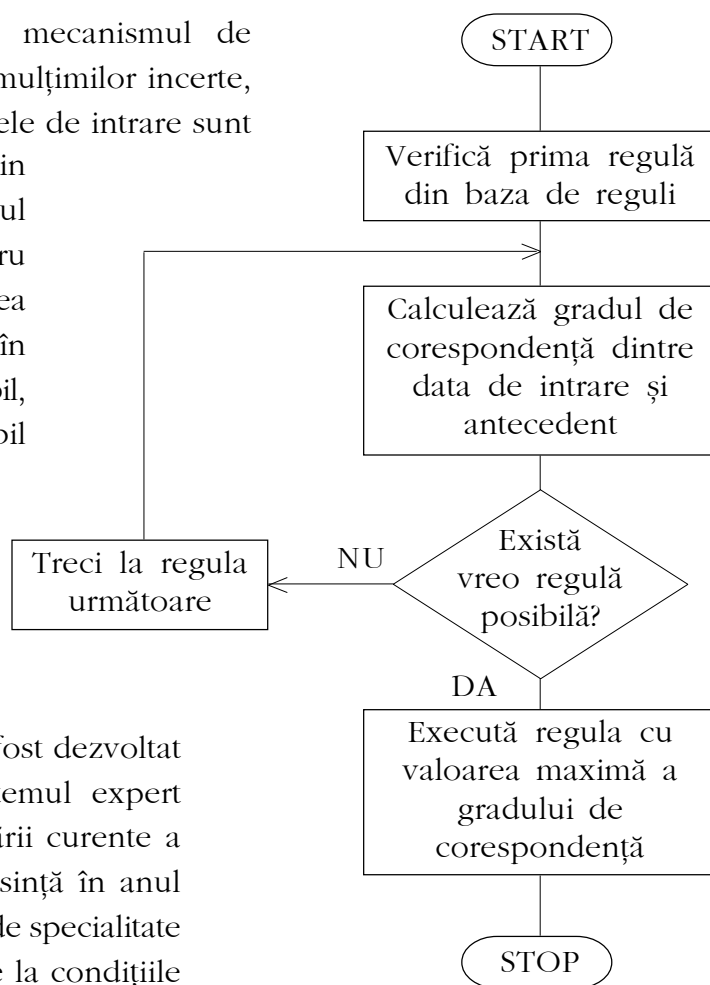


Fig.16 Procesul de inferență

folosind cauzele și condițiile de mediu ambiental, a fost estimată viteza de propagare a deteriorării. Considerând traficul intens la care era supus podul studiat, normele de proiectare utilizate la realizarea podului fiind o versiune veche din 1926, cauza deteriorărilor s-a dovedit a fi încărcarea exagerată prin roțile autovehiculelor și lipsa unor grinzi de distribuție. Mai mult, defectele drenajului de suprafață aveau o influență nefastă, afectând foarte mult estimările cu privire la durata de exploatare rămasă, deteriorările sub formă de fisuri datorate acestei cauze având viteza cea mai mare de propagare.

În domeniul proiectării structurilor din beton armat au existat numeroase preocupări pentru utilizarea sistemelor

bazate pe cunoștințe. O lucrare remarcabilă publicată în acest domeniu sub titlul *A Knowledge-Based System for the Design of Reinforced Concrete Halls* este cea realizată în Egipt de către Nabil A.B. Yehia (*Facultatea de inginerie, Universitatea Cairo, Ghiza*), Assem A.L.Sabbah (*General Organization for Housing, Building and Planning Research, Ghiza*) și Admad M.Mosallam (*Hosny Consulting Engineers, Cairo*). Sistemul realizat și prezentat era destinat utilizării pe microcalculatoare și dispunea de o bază de cunoștințe destinată proiectării halelor cu structura din beton armat. El conținea două subsisteme principale: unul *intelligent* (realizat în Prolog) pentru propuneri cu privire la configurația sistemului structural și un altul *algorithmic* (realizat în Fortran) destinat atât analizei cât și proiectării diverselor componente structurale. Lucrarea realizată reprezenta un pas important înainte în dezvoltarea sistemelor expert cu baze de cunoștințe în domeniul ingineriei structurale, deoarece era capabil să învețe din experiența situațiilor prin care era trecut anterior. Privit dintr-un alt punct de vedere, sistemul poate fi considerat un preprocesor inteligent pentru programele algoritmice ce înlocuiește cunoașterea aprofundată. La conceperea și dezvoltarea sistemului au fost considerate aspectele pozitive și realizările reușite din sistemele anterioare, dintre acestea meritând a fi amintite: **HI-RISE** [129] realizat pentru proiectarea preliminară a clădirilor înalte, **DESTINY** [176] pentru proiectarea structurală a clădirilor și **INDEX** [114] pentru proiectarea clădirilor industriale. Modulul inteligent (reprezentând preprocesorul), numit *Hall Layout Module*, folosea tehnici și instrumente din domeniul inteligenței artificiale, cum era utilizarea operațiilor pentru schimbarea stării spațiului până la atingerea unei anumite stări stabile (până la atingerea unui scop). De asemenea, era folosită o tehnică de generalizare/specializare pentru înțelegerea naturii problemelor necunoscute sistemului. Acest subsistem conținea cinci submodule, cu funcții pentru intrări, configurație, operații, învățare și ieșiri, descrise destul de sumar în lucrarea publicată. Mecanismul de control a fost prezentat în contextul celor cinci întrebări sugerate de Winston [193]. În lucrare există și o prezentare a domeniului problemei sub formă arborescentă, precum și a creșterii acestuia prin submodulele de învățare (achiziționare automată de cunoștințe). Modulul de proiectare (reprezentând procesorul logic), numit *Hall Design Module*, aplica algoritmi deja consacrați (preluați din cărțile tehnice de specialitate) atât pentru analiza cât și pentru proiectarea structurii sub formă de cadre poligonale din beton armat, incluzând și dimensionarea elementelor structurale.

Pentru proiectarea practică a plăcilor de bază la stâlpii metalici, T.J.McCarthy și Z.Nouas (*Department of Civil and Structural Engineering, UMIST, Manchester, England*) au dezvoltat sistemul expert **BPLATE**, publicându-și ideile în lucrarea intitulată *An Experimental Expert System for the Design of Column Base Plates*. Lucrarea lor și prototipul realizat sunt considerate de către mai mulți cercetători o demonstrație a trăsăturilor principale ale unui sistem expert. Datorită facilităților explicative (caracteristice sistemelor expert) prototipul poate fi utilizat cu succes și ca instrument în procesul de instruire al tinerilor ingineri. Modelul propus pentru procesul de proiectare dispune de trei trepte: proiectarea preliminară, proiectarea detaliilor și analiză, respectiv o apreciere critică a rezultatelor finale. Bazele de cunoștințe au fost realizate utilizând tehnica reprezentării în cadre (*frames*) care sunt apoi manipulate de către mașina de inferență în scopul dirijării

proiectării. Această modalitate de reprezentare s-a dovedit a fi cea mai potrivită pentru combinarea componentelor independente și pentru specificarea tipurilor de analize necesare de la caz la caz. Prin separarea bazei de cunoștințe de programul care efectuează controlul asupra proceselor, datele pot fi adăugate la sistem și modificate fără dificultate. Programul elaborat conține anumite cunoștințe a priori înmagazinate în baze de cunoștințe și calculează valorile ce reprezintă informații de proiectare (conform specificațiilor din BS5950) prin aplicarea unor algoritmi numerici. Cunoașterea este structurată sub formă arborescentă, reflectând astfel descompunerea ierarhică a problemei de proiectare. Această abordare este descrisă mai pe larg în [130] și [131]. Una din bazele de cunoștințe conține un număr de configurații referitoare la plăci de bază, iar cealaltă deține o serie de modele potrivite pentru configurații. Configurația se referă la descrierea componentelor de utilizat, pe când modelul oferă informații mai detaliate, indicând și metodele de analiză care trebuie efectuate. Programul a fost scris în AutoLISP [2], o implementare a limbajului XLISP în AutoCAD. Pachetul AutoCAD (realizat de Autodesk Inc.), unul dintre cele mai cunoscute pachete de proiectare asistată de calculator, poate fi astfel utilizat ca o interfață între utilizator și sistemul expert BPLATE. Această modalitate de elaborare a unui sistem expert în cadrul unui pachet existent de proiectare asistată permite dezvoltarea în continuare a programului pentru a realiza și o bază de date grafice pentru desene și soluții complete de proiectare.

O altă lucrare remarcabilă este cea realizată de către J.F.Peters (*Carnegie-Mellon University, Pittsburgh, Pennsylvania, U.S.A.*), descrisă în articolul intitulat *An Expert System for the Strength Evaluation of Existing Structural Members*. Sistemul expert realizat sub numele **SEES** era destinat evaluării rezistenței în cazul elementelor structurale existente. Motivele care au condus la realizarea acestui sistem au fost prezentate de către autor în articolul publicat, după cum urmează:

- expertiza clădirilor vechi în condițiile modificării destinației poate fi o problemă complexă, de mare finețe, experiență și intuiție inginerescă;
- s-au modificat normele care au stat la baza calculelor, la fel încărcările și proprietățile materialelor. Au survenit îmbătrâniri, degradări, uneori au dispărut chiar planurile de execuție și materialele vechi se identifică în general mai greu;
- uneori normele prea puțin precise lăsau la latitudinea proiectantului anumite aspecte legate de calcule, adoptarea soluțiilor constructive etc.

Cu ajutorul sistemului SEES se pot efectua verificări ale elementelor de tip stâlp sau grindă, realizate din diverse materiale (oțel, beton armat, lemn, materiale compozite) la solicitări variate (efort axial, încovoiere, forfecare, torsiune și încovoiere combinată cu efort axial). Răspunsul sistemului constă din validarea comportamentului preliminar în raport cu anumite criterii (cum ar fi: rezistența, deformațiile, ductilitatea etc.) prin efectuarea unor comparații.

Nu pot încheia această parte fără a aminti și câteva realizări din țara noastră în acest domeniu. În articolul intitulat *Raport asupra sistemului interactiv de programare DM-LISP*,

publicat în 1981 de către D.M.Sotirescu și M.Ștefănescu (*Facultatea de automatică, Institutul Politehnic din București*), s-a prezentat în detaliu un sistem implementat în cadrul catedrei de calculatoare din I.P.București, cu scopul de a utiliza acest dialect Lisp în cercetările legate de inteligență artificială și în domeniile conexe. Aceasta era una dintre primele materializări ale interesului pentru inteligența artificială în țara noastră. Din punctul de vedere al construcțiilor însă, primele rezultate remarcabile datează din 1985, când C.Botezat (C.C.E.-C.I.E.T.C.), D.Constantinescu, V.Dinu (I.P.C.T.) și M.Malița (I.C.S.I.T.-T.C.I.) au realizat sistemul expert **SEDAN** (Sistem expert pentru predimensionarea cadrelor din beton armat) printr-o colaborare a I.P.C.T. și I.P. București. Sistemul dispunea de patru faze succesive:

- ❶ introducerea datelor generale,
- ❷ predimensionarea grinzilor
- ❸ predimensionarea stâlpilor și
- ❹ furnizarea rezultatelor pentru varianta de cadru considerată.

Fazele ❷ și/sau ❸ pot fi omise în cazul în care utilizatorul dorește doar verificarea unui cadru predimensionat anterior (în această situație însă trebuie să precizeze dimensiunile acestuia). În vederea alegerii variantei finale de cadru, utilizatorul poate parcurge faza ❹ a programului de câte ori dorește, modificând una sau mai multe bare din cadru, până când obține valorile dorite ale parametrilor. Sistemul dispune de o interfață de utilizare *prietenosă* (exploatănd posibilitățile oferite de meniuri). Programul a fost scris utilizând limbajul LISP pe minicalculatoare INDEPENDENT și CORAL, necesitând în final aproximativ 1Mb memorie. Rularea sistemului se realiza sub interpretorul TCLISP (realizat de către D.Tufiș, I.C.S.T.-T.C.I.) apelând și module scrise în Fortran-77 (programul SECON etc.). Timpul de rulare depindea de opțiunile utilizatorului, de numărul de nivele ale cadrului și de încărcarea sistemului cu alte module ce se executau în paralel.

3.3.9 Sisteme în domeniul ingineriei seismice

Dezvoltat de către K.B.Dussom în teza sa de masterat la nivel de prototip, sistemul expert **QUAKE** este destinat să asiste selectarea accelerogramelor de proiectare specifice amplasamentelor preconizate, dintr-o bază de date cu înregistrări. Sistemul a fost realizat cu shell-ul M.1, iar rutinele externe pentru calculul caracteristicilor mișcării terenului au fost codificate în C și Fortran. Baza de date conține articole pentru caracteristicile principale ale accelerogramelor înregistrate în California (S.U.A.) în ultimii 60 de ani, acestea referindu-se la identificarea completă a înregistrării (localitate, adresă, coordonate geografice, data, durata, poziția și coordonatele epicentrului etc.), momentul apariției și valorile extreme ale accelerației, vitezei, deplasării, magnitudinea, tipul terenului etc. Înregistrările efective ale accelerogramelor sunt stocate pe benzi separate de unde pot fi copiate după consultarea sistemului. Baza de cunoștințe constă din două părți: prima conține informații cu privire la strategia de rezolvare a problemelor și la controlul sistemului de inferență, iar a doua conține cunoștințe deduse din experiența și intuiția experților umani cu privire la proiectarea structurală antiseismică. Sistemul este deschis, deci poate

fi actualizat, completat etc. după dorință. Consultarea este orientată spre obținerea în final a unei liste de accelerograme selectate, ordonate după criteriile de similitudine și localizare, cu respectarea limitelor de credibilitate acceptate de utilizator. Ele pot fi accelerograme înregistrate, sau generate (artificiale) folosind diverse tehnici (prin combinare, utilizând programe ca EQGEN, SIMQKE, ARMD, sau folosind combinații ale funcțiilor Green, sau a celor de evoluție a spectrelor dependente de frecvență etc.). La testele de credibilitate s-au aplicat formulele dezvoltate de Trifunac și Brady care au în vedere vârful accelerației terenului, intensitatea măsurată ca integrală a pătratului accelerațiilor în lungul axei timpului și gradientul înregistrării măsurat prin raportul dintre intensitate și durată. Prin dialogul cu utilizatorul sistemul cere informații cu privire la amplasament (natura terenului) și construcție (tip, frecvența fundamentală etc.). Ordonarea variantelor propuse de sistem se realizează pe baza factorilor de credibilitate și a avariilor potențiale asociate accelerogramelor respective. Descrierea detaliată a sistemului împreună cu rezultatele obținute în urma implementării au fost publicate prin lucrarea: *QUAKE: An Expert System for the Selection of Design Earthquake Accelerogram*, de către K.B.Dussom, T.Hadj-Hamon și R.M.Bakeer (Tulane University, New Orleans, S.U.A.).

Sistemul expert **WQUAKE** destinat evaluării comportamentului clădirilor solicitate sau deteriorate în timpul cutremurelor este prezentat în lucrarea *A Knowledge-Based Assistant for Earthquake Resistant Design* de către H.Adeli și S.L.Hung (Ohio State University, S.U.A.).

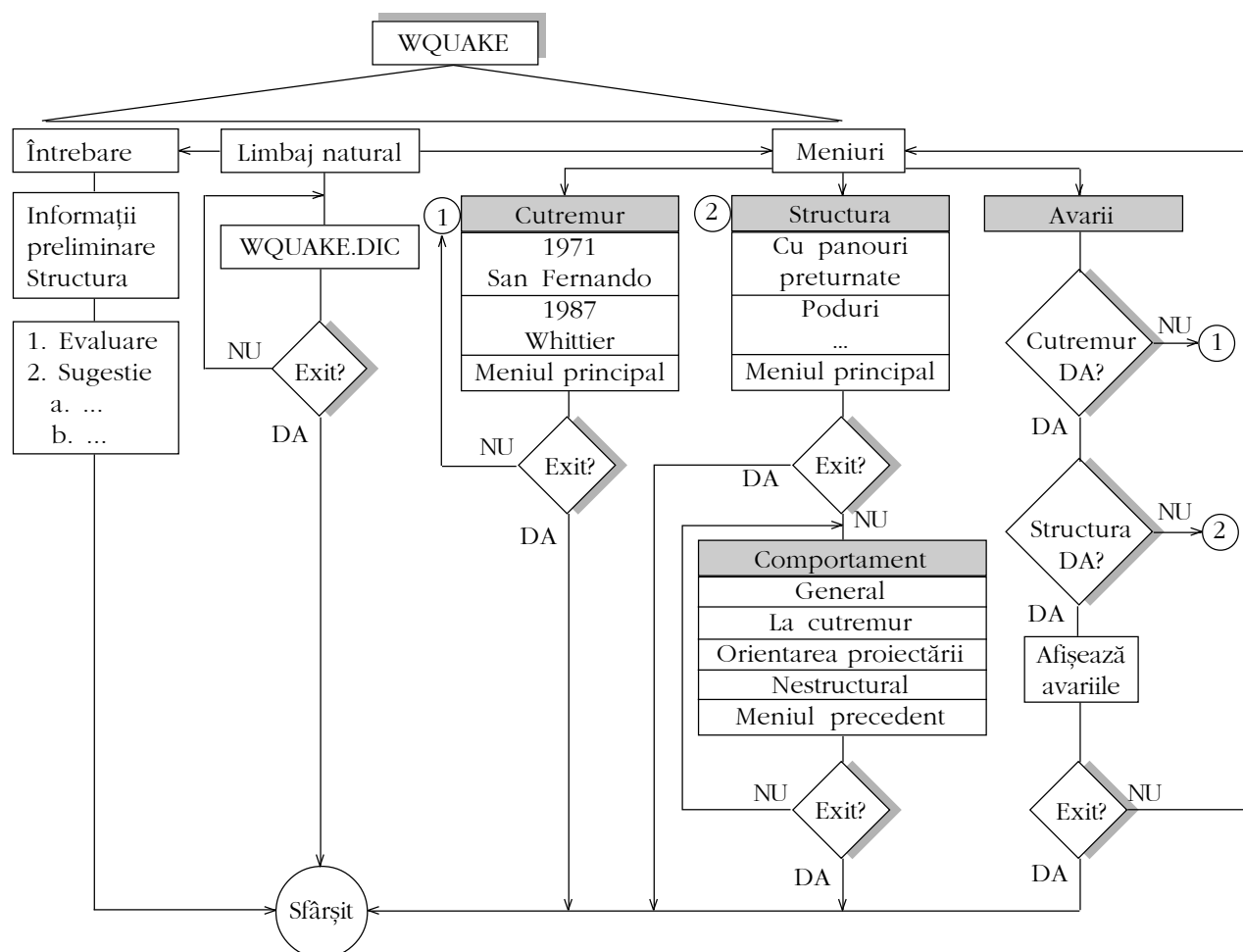


Fig.18 Detalierea interfeței utilizator din sistemul WQUAKE

Sistemul a fost dezvoltat cu ajutorul generatorului GURU și a limbajului TURBO C. Baza de cunoștințe a acestui sistem conține învățături extrase din rapoartele ingineresti de constatare - efectuate în legătură cu acțiunile distructive ale cutremurelor din San Fernando (1971) și Whittier (1987), California,- comportamentul structural și avariile fiind prezentate prin baze de date relaționale și reguli de producție. Era în intenția autorilor de a continua lărgirea bazei de cunoștințe în legătură și cu alte tipuri, cazuri (exemple) de cutremure, precum și integrarea sistemului WQUAKE într-un sistem expert mai larg (EXQUAKE), acesta constituind un repertor al cunoștințelor referitoare la avarii și comportamentul structural manifestate în timpul cutremurelor.

Mediul generator GURU constituie o platformă de dezvoltare a sistemelor expert, incluzând o serie de facilități (ca de exemplu: managementul bazelor de date, aplicații tip foi de calcul, procesarea limbajului natural etc.). Astfel interfața utilizator-sistem WQUAKE a fost organizată pe trei ramuri: interfața de interogare (cu privire la structură, evaluarea avariilor, sugestii); procesarea limbajului natural (prin vocabularul WQUAKE.DIC cu cca 300 de cuvinte engleze implementate cu comanda *WORDMAN* și exploatate prin comanda *CHAT* din GURU); interfața tip meniu, ramificată pe informații despre cutremur, date despre structură (descriere, comportament în timpul cutremurelor, sumar etc.), avarii constatate (la cutremurele evaluate amintite mai sus: din 1971 la San Fernando și din 1987 la Whittier). Facilitățile grafice ale mediului GURU neadecvate aplicațiilor din ingineria structurală, au fost suplinite prin posibilitățile oferite de limbajul TURBO C al firmei Borland (în prezent Inprise, Inc.). În figura 18 este ilustrată schema interfeței utilizator din sistemul WQUAKE. Modelarea cunoștințelor în GURU se poate face doar prin reguli de producție de tipul

if <condiții> **then** <concluzii> **do** <acțiuni>

dar în unele strategii de reprezentare logică structurală se preferă utilizarea cadrelor (frames). Această limitare a fost depășită, folosind facilitățile existente în GURU pentru crearea bazelor de date relaționale (cu relații explicite/implicite între înregistrări). În baza de cunoștințe realizată pentru sistemul WQUAKE este acumulată experiența evolutivă cu privire la comportarea în cazul cutremurelor, în special a construcțiilor din panouri preturnate pe șantier (tilt-up constructions) - au fost urmărite și raportate peste 120 de asemenea clădiri - pornind de la cutremurul din San Fernando (1971), cu modificările ulterioare aduse pe linie antiseismică în normele generale de construcții, până la efectele acestora, constatate la cutremurul din Whittier (1987). În figura 19 se prezintă organizarea cunoștințelor referitoare la avariile produse de cutremure în construcțiile realizate din panouri preturnate.

Într-un raport de cercetare, intitulat *Expert Systems in Earthquake-Resistant design of Buildings*, publicat în *N.C.E.E.R. Bulletin* (vol.2, nr.3, iulie 1988, pag.1-2) se arată că trei universități prestigioase din S.U.A. (Cornell, Lehigh și Carnegie-Mellon) au convenit să dezvolte fiecare câte un modul de sistem expert destinat proiectării antiseismice a clădirilor, urmând ca în final acestea să fie integrate într-un sistem unic, util arhitecților și inginerilor constructori pentru asistarea proiectării, respectiv a evaluării riscului de avarii la clădirile amplasate în zone seismice.

Astfel, la Cornell University a fost dezvoltat sistemul **STRAKE** care înmagazinează cunoștințe din codurile existente și din procedurile de evaluare, completate cu cunoașterea obținută din încercările de laborator, analizele numerice, opiniile experților umani și nu în ultimul rând din rapoartele de constatare întocmite de specialiști după vizitarea zonelor în care au avut loc cutremure. Aici sunt înglobate și cunoștințe din Japanese Rapid Evaluation Procedure, EERI, SEAOC, de la Chinese Academy of Building Research din Beijing, din proiectul SOCRATES al Universității Cornell și Ithaca Software (HOOPS) etc. Acest modul este menit să asiste inginerii structuriști pentru verificarea conceptelor lor de proiectare în raport cu regulile de proiectare antiseismică incluse în baza de cunoștințe, pentru a-i ajuta la realizarea proiectelor unor clădiri mai sigure.

În cadrul universității Lehigh s-a dezvoltat sistemul **ARCHQUAKE** menit să introducă cunoștințe de proiectare structurală antiseismică încă din faza incipientă a proiectării arhitecturale. Sistemul se concentrează pe aspectele arhitecturale ale proiectării clădirilor (ca de exemplu configurația orizontală/verticală, vecinătăți etc.). Vizualizarea grafică a caracteristicilor nedorite vor ajuta arhitecții să analizeze consecințele variantelor de proiectare și să adopte soluții mai bune. Baza de cunoștințe foarte vastă a sistemului cuprinde informații extrase dintr-o serie de proiecte de cercetare din cadrul N.C.E.E.R., și din alte proiecte (SEAOC, ICBO, BOCA, Centrul ATLSS din Lehigh University etc.), având ca obiect studii analitice, metode de evaluare, cercetări experimentale etc. menite să asigure descrierea arhitecturală completă a clădirii și să faciliteze transferul informațiilor relevante între procesele de proiectare (de exemplu între arhitect și inginerul proiectant etc.).

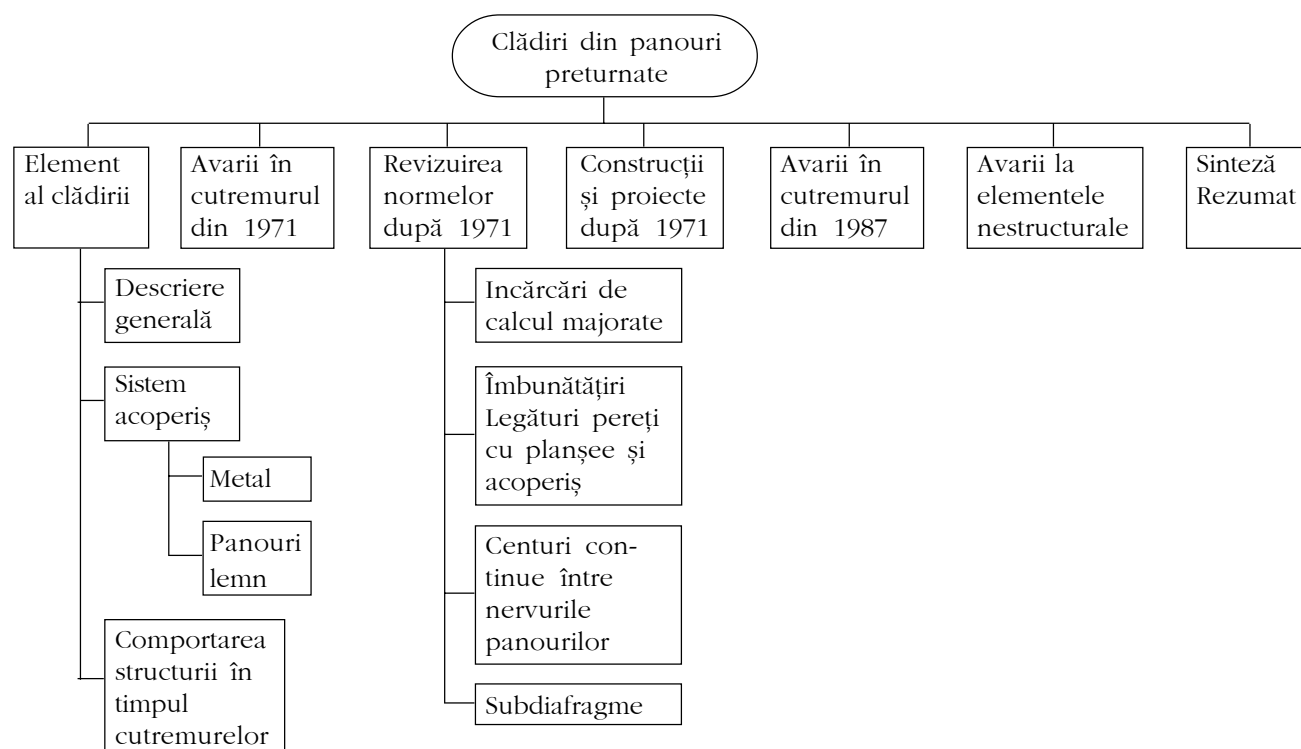


Fig.19 Organizarea cunoștințelor privind comportarea la cutremure a clădirilor din panouri preturnate.

La Carnegie-Mellon University atenția a fost îndreptată spre dezvoltarea sistemului expert **CMU-EVAL** destinat asistării la evaluarea siguranței seismice a clădirilor existente și să propună recomandări de reabilitare pentru clădirile respective în vederea îmbunătățirii calităților antiseismice, dacă acest lucru este necesar. Inițial, cea mai mare parte din baza de cunoștințe a fost preluată din documentația AT-14, respectiv din proiectele N.C.E.E.R. referitoare la evaluarea performanței antiseismice a clădirilor existente și într-o măsură mai mică din studii analitice sau din cercetări experimentale.

Modulele STRAKE și CMU-EVAL de sisteme expert au fost dezvoltate utilizând mediile de generare OPS-83, respectiv EXSYS și KEE, iar pentru dezvoltarea sistemului ARCHQUAKE s-a folosit limbajul PROLOG și pachetul *3D Modeling Graphics Software*.

3.3.10 Sisteme pentru ingineria geotehnică

În acest domeniu, J.Oliphant (*Heriot-Watt University, Edinburgh*) și D.I.Blockley (*University of Bristol*) au prezentat un sistem expert destinat alegerii și proiectării structurilor de retenție (ziduri de sprijin, palplanșe, grupuri de piloți etc.) în articolul *Knowledge Based System: Advisor on the Selection of Earth Retaining Structures*. Rezultatele cercetărilor publicate de către cei doi autori sunt derivate din posibilitatea definirii a trei direcții principale de dezvoltare a bazelor de cunoștințe în domeniul structurilor de retenție. Prima direcție se referă la investigarea pe teren a datelor geotehnice și de mediu pentru interpretarea acestora, a doua se referă la alegerea tipului convenabil pentru structura de sprijin, iar a treia se referă la proiectarea propriu zisă a structurii selectate. Sistemul expert prezentat în lucrarea menționată abordează ultimele două direcții amintite.

Pentru elaborarea cunoștințelor organizate ierarhic și logic, care să modeleze datele și opiniile de specialitate, referitor la problematica alegerii tipului structurii de sprijinire și aprecierea siguranței, a fost utilizată teoria pământurilor (*Grounded Theory*) a lui Glaser și Strauss (1968). Baza de cunoștințe a fost redactată în formatul FRIL/PROLOG și implementată pe un calculator Vax 11/750 (care rula de fapt C-Prolog) sub sistemul de operare Unix. Mașina de inferență era bazată pe funcțiile oferite de către shell-ul FRISP (*Fuzzy Relational Inference with Suport Logic*) care permitea utilizatorului să interogheze baza de cunoștințe fie în modul clasic Prolog, fie utilizând meniuri într-o multitudine de variante. Sistemul de inferență permitea și tratarea cunoștințelor incerte prin introducerea conceptelor de suport necesar ($0 < N < 1$) și a suportului posibil ($0 < P < 1$). Primul măsura încrederea cu care setul de evaluări ale atributelor aparțin sigur conceptului exprimat prin propoziție, iar al doilea includea elementele nesigure, incerte (nenecesare) dar posibile. Pentru coerență trebuie să fie îndeplinite

$$N + (1 + P) \leq 1 \text{ sau } P \geq N.$$

Astfel, dacă perechea $(N, P) = (0, 0)$ avem cazul de *sigur fals*, neîntemeiat, pe când $(N, P) = (1, 1)$ înseamnă *sigur adevărat* deci întemeiat. Cazul $(N, P) = (0, 1)$ corespunde situației de *nu știu*, deci mașina de inferență adoptată permitea tratarea și a acestui caz. Pentru frazele

compune blocul de inferență evalua perechea (N,P) rezultantă după regula modelului multiplicativ sau al modelului de minim (diferit de teoria probabilităților), respectiv:

Dacă $X : [N1, P1]$ și

$Y : [N2, P2]$

Atunci $(X \text{ AND } Y) : [N1 \times N2, P1 \times P2]$

la modulul multiplicativ, sau

$[\min(N1, N2), \min(P1, P2)]$ la modelul de minim;

$(X \text{ OR } Y) : [N1 + N2 - N1 \times N2, P1 + P2 - P1 \times P2]$ la modulul multiplicativ, sau

$[N1 + N2 - \min(N1, N2), P1 + P2 - \min(P1, P2)]$ la modelul de minim.

Sistemul permite și documentarea soluțiilor propuse (prin opțiunea *WHY*) respectiv analiza comparativă și de sensibilitate a variantelor recomandate (prin opțiunile la cererea *DO A WHAT IF QUERY*).

Un alt sistem expert din același domeniu este prezentat de I.E.G.Davey-Wilson (*Oxford Polytechnic*) în lucrarea *Development of a Prolog Based Expert System for Groundwater Control*. Sistemul interactiv **GWX** (*Groundwater Expert*) asistă utilizatorul în evaluarea celor mai eficiente procedee și metode pentru controlul apelor freatice la lucrările de săpături (excavații), pe baza parametrilor locali caracteristici unui amplasament.

Baza de cunoștințe conține informații despre 27 de metode diferite de control al apelor freatice. Informațiile sunt fragmentate în fapte caracteristice parametrilor principali ai fiecărei metode, acestea la rândul lor sunt împărțite pe segmente, fiecăruia fiindu-i atașat un rang (pentru comparabilitate) care clasifică cât de potrivită este metoda respectivă din punctul de vedere al aceluși parametru. Aceste segmente conțin și comentarii referitoare la context. Mare parte din baza de cunoștințe a fost completată după raportul CIRIA referitor la controlul apelor freatice pentru lucrările temporare, în care sunt detaliate și evaluate cantitativ și calitativ metodele respective. După analiza acestui material au fost identificați 12 parametri principali:

tipul proiectului; tipul terenului; adâncimea de săpare; mărimea săpăturilor; calendarul de instalare; durata de execuție; distanța față de cea mai apropiată construcție; zgomotul; vibrațiile; tasarea; oscilația apei; costul.

La acestea au mai fost adăugate: adâncimea până la apa freatică, și adâncimea până la roca de bază. Acești parametri se referă la toate cele 27 de metode, între acestea fiind incluse și: izolarea cu palplanșe, cu rășini, înghețare cu azot lichid, electro-osmoză etc. Faptele și regulile au fost reprezentate în cadrul bazei de cunoștințe prin limbajul LPA PROLOG sub sistemul de operare DOS fiind fragmentate în 5 secțiuni. Baza de cunoștințe a prototipului funcțional prezentat conținea inițial 135 de predicate și 533 de fapte împărțite în 1400 de segmente. Sistemul expert GWX furniza în final rapoarte după cele mai bune și cele mai rele metode corespunzătoare cazului studiat, detalii referitoare la fiecare metodă (în ordinea clasificării), permite resetarea parametrilor (a ponderilor implicite inițiale), modificarea datelor de intrare etc.

3.3.11 Sisteme pentru tehnologia lucrărilor și robotică în construcții

Ca exemplu de sistem expert din acest domeniu, merită să amintesc lucrarea *The Development of a Real Time Expert Sistem for Excavation* publicată de D.A.Bradley și D.W.Steward (*University of Lancaster*) în care se descrie teoria realizării unui robot excavator a cărui funcționare autonomă ar necesita coordonarea printr-un sistem expert, asigurând o reacție dinamică în timp real din partea mecanismelor robotului.

Deși există exemple de automatizare și de robotizare în principal în domeniul producerii materialelor de construcții, idea robotizării șantierelor de construcții va fi materializată probabil în viitor, rămânerea în urmă față de robotizarea din industrie datorându-se în special organizării precare a șantierelor și complexității condițiilor locale de lucru. Introducerea roboților pe șantiere se poate realiza treptat prin dezvoltarea unor utilaje specializate pentru anumite tipuri de lucrări.

Exemplul abordat, al unui robot excavator, implică crearea bazei de cunoștințe organizate pe 3 nivele, vizând:

- *cunoștințele strategice* (de exemplu: cum să fie amplasat inițial excavatorul pentru a putea asigura calea de acces a camioanelor și în final ca robotul să poată părăsi golul excavat),
- *cunoștințe tactice* (de exemplu: sub ce unghi va ataca cupa un anumit tip de teren),
- *cunoștințe manuale* (care ar modela acțiunile unui operator mecanic).

La acestea s-ar mai adăuga o bază de *cunoștințe de diagnostizare*, care să identifice eventualele instalații subterane, rețele etc. și să asigure o oprire de urgență.

Problema controlului în timp real se poate soluționa prin modularea inferenței printr-un "manager" cu rol de consultant care ar trata problemele strategice și tactice, pe când modulul de control (cu intrări de la senzori și ieșiri - cu un *feed-back* rapid - la subansamblele care comandă mișcările mecanice ale robotului) ar procesa cunoștințele de la nivelul tactic și manual. Aici e vorba în principal de reluarea ciclurilor de excavare tipice pe baza noilor coordonate furnizate de "manager". Se apreciază că prin automatizare, ciclul mediu de excavare de cca 10 secunde în cazul unui operator uman, va putea fi redus la cca 6 secunde pentru excavatorul robot.

3.3.12 Educația în construcții și sistemele expert

După cum a reieșit din anumite cercetări, sistemele expert pot fi aplicate cu succes și în procesele educative din domeniul construcțiilor. Cărțile, notițele etc. pot oferi surse importante și valoroase de informare, dar rareori conțin cunoștințe euristice, limitând astfel abilitatea studenților de a înțelege probleme și aspecte practice și reale. Un sistem expert bine proiectat și corespunzător realizat, poate reduce semnificativ această dificultate în

ceea ce privește însușirea convențională a cunoștințelor. De-a lungul folosirii oricărui tip de sistem expert, studentul câștigă o anumită experiență și familiarizare cu o aplicație importantă și evolutivă. Asemenea sisteme pot facilita relativ ușor ca studentul să înțeleagă logica și fundalul tehnic al proceselor din construcții. Acestea sunt și concluziile la care au ajuns autorii lucrării *The Educational Value of Expert Systems*: S.Alkass și F.C.Harris (*School of Construction Engineering and Technology, Wolverhampton Polytechnic, England*), studiind două sisteme expert: **CLAIM EXPERT** [19] (un sistem destinat asistării deciziilor referitoare la formularea unor acțiuni legale) și **ESEMPS** [21] [22] (un sistem expert pentru mișcarea terasamentelor și alegerea echipamentelor). Sistemul CLAIM EXPERT este capabil să identifice diferitele aspecte potențiale de îndreptățire, din faptele date, și oferă linii de ghidare pentru pregătirea documentației necesare, împreună cu explicații asupra raționamentelor parcurse. Sistemul se bazează pe reguli și este constituit dintr-o serie de întrebări cuplate prin structuri decizionale (*IF-THEN*) sub formă arborescentă. Sistemul ESEMPS a fost realizat în scopul atingerii următoarelor obiective:

- ❶ Obținerea unei înțelegeri cât mai precise asupra metodelor de analiză și a metodelor de luarea deciziilor utilizate de către personalul implicat din companiile cu profil de construcții de drumuri și închiriere de utilaje și echipamente.
- ❷ Aranjarea cunoștințelor colectate într-o formă utilizabilă pentru sfătuirea corectă a neinițiaților (sau a celor cu experiență redusă) asupra metodelor de construire și despre echipamentele corespunzătoare diferitelor condiții de lucru.
- ❸ Cuplarea la sistem a unor programe externe destinate calculelor economice și folosirii resurselor.

Sistemul este bazat pe reguli, mecanismul de inferență permițând deducțiile prin înlănțuire înainte și înapoi, incertitudinile fiind tratate prin valori numerice (de la +5 la -5). Figura 20 ilustrează conturul relațiilor din interiorul sistemului ESEMPS. Atenția autorilor a fost îndreptată mai ales către expunerea studenților la conceptul sistemelor expert, la început încurajând construirea modelelor mai simple, iar mai apoi ale aspectelor mai complexe legate de disponibilitățile de autoeducare. Deși original cele două sisteme expert nu erau destinate scopurilor didactice, ele și-au dovedit totuși utilitatea și în acest domeniu. Mai mult, s-a evidențiat faptul că facilitățile accesibile din ultimele *shell*-uri asigură conceptul aproape ideal pentru elaborarea unor sisteme expert destinate procesului educațional.

În general, calculatoarele au fost și sunt folosite în foarte multe moduri pentru asistarea procesului de învățare, rulând și exploatând pachete de programe variate, elaborate pentru satisfacerea nevoilor unor industrii complexe, cum ar fi și construcțiile. Lansdown în 1982 prezenta următoarea enumerare [118]:

- aplicații în care calculele joacă un rol major (de exemplu din domeniile analizei structurale și ale proiectării);
- alte programe, care manipulează texte (cum ar fi procesoarele de texte, editoarele);

posibile soluții la asemenea probleme. Chiar pachete de programe bine încheiate, prin combinarea unor facilități explicatorii, pot satisface cerințele unei înțelegeri mai bune. Sistemele expert manevrează mai degrabă cunoștințe decât date, oferind astfel utilizatorului oportunitatea de a face alterări (de exemplu să corecteze o greșeală sau să adauge reguli noi în baza de cunoștințe), chiar variante. Facilitatea explicativă oferită de sistemele performante este esențială pentru utilizator (după cum este important de știut de ce a sugerat sistemul o soluție particulară sau ce raționament se află în spatele acestuia), fiind clar un instrument educațional foarte valoros. Sistemele expert permit astfel transferarea sugestivă a cunoașterii expertului către cei mai puțin inițiați, deci implicit și către cei care doresc să se perfecționeze. De asemenea, cunoștințele sunt accesibile tot timpul, de la pornirea sistemului de calcul, ceea ce reprezintă un avantaj considerabil față de sarcina laborioasă a strângerii informațiilor difuze despre un subiect.

CAP.4 SISTEME PROPRII DEDICATE CONSTRUCȚIILOR

4.1 PROTOTIPURI ȘI APLICAȚII

4.1.1 Sistemul expert BPC

Lucrând ca inginer în cadrul oficiului de calcul al T.A.G.C.Ind Oradea, în anul 1989 am absolvit cursul de formare: *Analiză și proiectare sisteme și produse informatice de gestiune - FELIX-MINI (cod: F1A)*, organizat în cadrul *Centrului Teritorial de Calcul Electronic din Cluj-Napoca*. Ca temă, pentru lucrarea de absolvire, mi-am propus realizarea unui produs informatic conversațional care să permită consultarea bibliotecii de programe C.P.A.C. (*Cercetare și Proiectare Asistate de Calculator*) în domeniul calculului construcțiilor. Această bibliotecă de programe conținea aproape toate aplicațiile existente în țara noastră (destinate calculului construcțiilor) și, pentru o orientare mai ușoară, caracteristicile lor erau publicate într-un catalog editat de către I.C.C.P.D.C.- I.P.C.T. București. Scopul acestui sistem propus de mine era eliminarea timpilor pierduți cu căutarea unor pachete de programe existente, prin automatizarea procesului de căutare/alegere, oferind totodată și o informare concisă și eficientă cu privire la caracteristicile, facilitățile oferite, precum și la resursele necesitate. Adică, dacă cineva știa ce calcule dorește să efectueze și ce echipamente are la dispoziție, i se ofereau denumirile pachetelor de programe corespunzătoare și modul de acces la ele. În mod similar, se puteau obține și soluții privind echipamentele necesare, știind ce programe se doreau a fi folosite. Sistemul, realizat pentru minicalculatoarele din gama CORAL și INDEPENDENT, a fost denumit BPC (*Biblioteca de Programe în Construcții*), pachetul fiind compus din 2 aplicații independente (FIX și IND) și dintr-un program segmentat (BPC), scrise toate în FORTRAN-77. Programul FIX avea scopul de a genera fișiere secvențiale cu articole de lungime fixă (cerință de bază pentru exploatarea acestora în acces direct), iar programul IND servea pentru generarea fișierelor secvențial indexate cu acces prin valori de chei. Programul BPC, după lansare, încărcă mai întâi subrutina COD pentru identificarea perifericelor utilizate (VDT 52S sau DAF2020 în cazul terminalelor) după care urmau temporizate: un ecran cu titlu, un alt ecran de prezentare și în fine un ecran cu meniul principal (toate afișările erau realizate cu subrutina ECR). În continuare se încărcău subrutinele corespunzătoare ramurilor selectate din meniuri până la oferirea soluțiilor găsite din baza de date. Programul principal din BPC permitea și folosirea limbajului natural pentru scurtarea obținerii unor soluții (prin evitarea meniurilor imbricate). În cazul frazelor interrogative libere ale utilizatorului, la început era activată subrutina INT care analiza termenii introduși (completat de subprogramul CIT), identificând succesiv cuvintele cu cele din vocabularul fix (fișierul indexat VOC.DAT) și în caz de succes, pornind de la informațiile din articolul identificat, se completau niște vectori. Sfârșitul analizei era marcat prin memorarea într-o variabilă a indicelui care conținea terminatorul de frază (punct sau semn de întrebare). Informațiile oferite de subrutina INT erau apoi prelucrate de o altă subrutină (INC) de interpretare care, pe baza valorilor, furniza subprogramului REZ datele necesare construirii răspunsului la întrebarea utilizatorului.

Cu ocazia susținerii examenului de absolvire, după prezentarea lucrării întocmite și a sistemului realizat, domnul ing. Constantin Vasile (coordonatorul de programe) a amintit de sisteme expert, opinând că lucrarea prezentată de către mine s-ar putea înscrie într-o asemenea direcție de dezvoltare. Acesta a fost momentul în care am aflat de existența unor asemenea sisteme, urmând ca după puțin timp să mă înscriu la doctorat abordând domeniile dezvoltării și aplicării acestor sisteme în construcții.

Primele preocupări practice le-am avut în direcția dezvoltării acestui prim prototip ca sistem expert. Am creat baza de cunoștințe al sistemului utilizând 6 fișiere cu acces direct și am dezvoltat modulul de inferență împreună cu un analizor semantic pentru a trata cererile formulate prin fraze libere de către utilizator. Raspunsurile corespunzătoare cererilor interpretate erau furnizate de către un generator, interfața utilizator și interpretorul de comenzi fiind și ele regândite. Acestea din urmă ofereau, printr-un sistem de meniuri, opțiuni sub formă de răspunsuri directe sintetice (fără a face apel la modulul de inferență) precum și explicații semantice și sintactice. Oarecum paradoxal, pentru exprimarea unor cereri sub formă de fraze "libere", utilizatorul trebuia să respecte totuși o anumită sintaxă, aceasta materializându-se prin câteva reguli simple și semne de

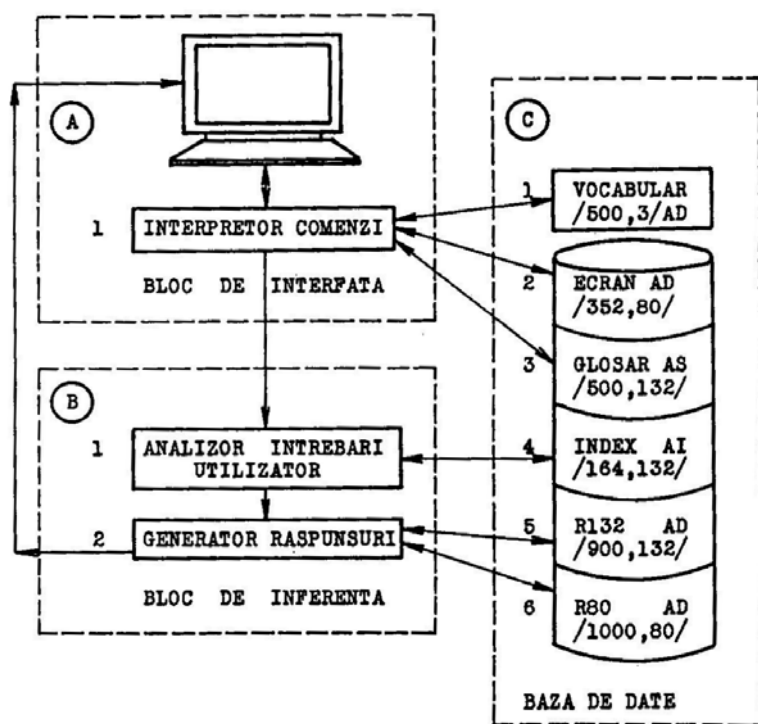


Fig.21 Arhitectura sistemului expert BPC

punctuație prestabilite (se analizau doar cuvintele scrise cu majuscule, virgula suplinea operatorul logic AND iar slash-ul operatorul logic OR, iar ca și terminator de frază se accepta doar punctul sau semnul de întrebare). Vocabularul dezvoltat pentru analizorul semantic era destul de restrâns dar suficient de eficient, servind și generatorul de răspunsuri. Sesiunea de consultare se putea înregistra opțional într-un fișier cu ajutorul unui trasor, în ideea revenirii sau derulării secvențiale ulterioare a pașilor parcurși. Arhitectura acestui sistem expert (prototipul BPC) este ilustrată în figura 21.

În anii care au urmat am încercat să mă documentez cât mai temeinic în domeniul de cercetare ales. Cum era și normal, am fost nevoit să sacrific destul de multe lucruri pentru a obține rezultatele dorite. Am fost nevoit să achiziționez cărți din străinătate, să apelez la bunăvoința mai multor persoane pentru a procura și copia diverse articole, reușind chiar să cumpăr și un mediu de dezvoltare comercial (Micro-PS) cu care am început primele experimentări. Tot prin bunăvoința mai multor persoane am avut acces la prototipuri de sisteme expert realizate în domeniul medicinei, electrotehnicii, mecanicii etc., precum și

la diverse medii de dezvoltare comerciale și demonstrative. Mi-au fost de foarte mare ajutor și copiile articolelor și extrasele conferințelor referitoare la sisteme expert, primite de la colegi (mulțumiri în special domnilor profesori P.Alexa și I.Olariu). Dintre primele încercări aș aminti și transcrierea sistemului BPC cu ajutorul generatorului de sisteme expert TURBO EXPERT (primit prin bunăvoința domnului profesor I.Ciongradi). În ANEXA 1 sunt prezentate câteva aspecte referitoare la acest sistem.

Realizarea unui prototip dezvoltat din sistemul BPC a făcut și obiectul unor contracte, încheiate între Institutul Politehnic (actualmente Universitatea Tehnică) din Cluj-Napoca și Ministerul Învățământului și Științei, în anii 1991 - 1992. Astfel, într-o primă etapă (în 1991) am realizat modelul matematic al sistemului [78], iar în anul următor partea interfeței utilizator-sistem [79].

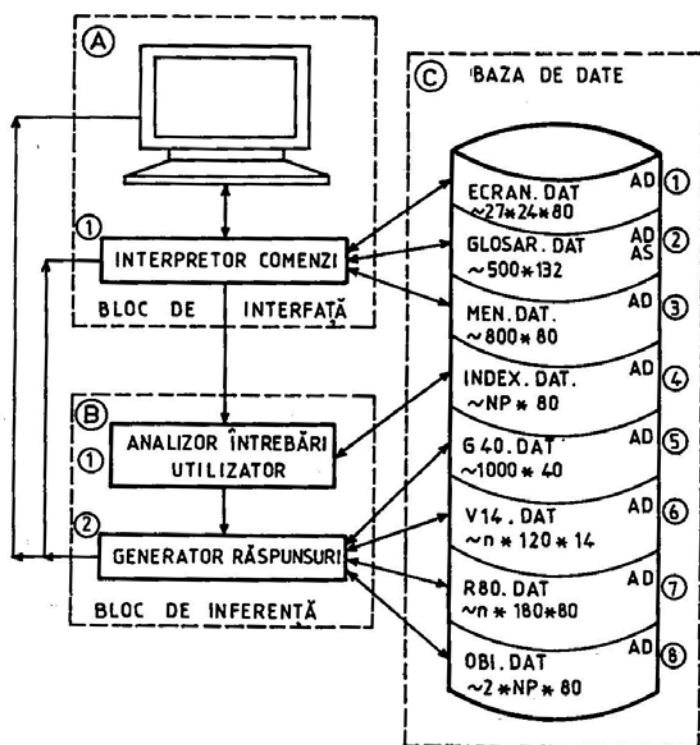


Fig.22 Arhitectura sistemului expert BPC, reconceptuat pentru PC-uri

a fost majorat de la 6 la 8, odată cu redimensionarea (optimizarea) acestora (figura 22). Numărul modulelor de programe din cadrul noului sistem BPC a fost extins la 24 (de la 13 în varianta anterioară).

Dezvoltarea bazei de cunoștințe implica regândirea modului de organizare și de exploatare al sistemului, în primul rând în sensul unei mai bune orientări a utilizatorului (neinformatician) la precizarea cererilor sale. Rezolvarea acestui aspect a necesitat modificarea logicii sistemului de inferență. Astfel de la ideea inițială de a se genera răspunsuri pe baza unor cereri sub formă de fraze libere utilizator (care implicau totuși respectarea unor reguli elementare de sintaxă și necesitau tastări multiple, iar în caz de

Prototipul dezvoltat cu ocazia acestor două contracte era bazat pe cel realizat în Fortran, descris anterior. Acest lucru nu a însemnat doar transpunerea prototipului anterior de pe minicalculatoare pe PC-uri, ci reanaliza întregului sistem ținând cont și de schimbările inevitabile petrecute în domeniul abordat precum și de performanțele și posibilitățile apărute prin dezvoltarea tehnici de calcul. După o analiză atentă, am reconceptuat atât structura fișierelor din baza de cunoștințe cât și instrumentele necesare funcționării și gestionării sistemului. Arhitectura noului sistem BPC era similară cu cea anterioară, schimbările ieșind în evidență mai pregnant la configurarea bazei de cunoștințe. Astfel numărul fișierelor

eșec reformularea întrebărilor) am optat pentru generarea răspunsurilor pe baza unor cereri conduse prin meniuri arborescente (înlănțuite). Deci blocul de inferență a fost reproiectat, renunțând la interpretorul semantic, sesiunile desfășurate cu interogări exprimate prin fraze libere dovedindu-se ineficiente (din cauza erorilor frecvente de tastare și a timpului necesar unor formulări corecte) mai ales la utilizatorii neexperimentați. Pentru a sugera complexitatea interfeței concepute: există 57 de meniuri destinate doar procesului de interogare/consultare (înlănțuirea acestora fiind prezentată în ANEXA 1).

Funcțiile și structura noului bloc de inferență sunt prezentate schematic în figura 23. Nucleul acestui modul este alcătuit din generatorul de răspuns (subrutina GR) care la rândul lui apelează 28 de rutine specializate pentru tratarea regulilor stocate în articolul din GEN.DAT al programului analizat. La generarea răspunsurilor pentru selectarea opțiunilor prevăzute în meniuri, răspunsul adecvat se obține extrăgând din fișierul G40.DAT textele fixe de dialog (specificațiile de antet) și din articolul din GEN.DAT al programului interogat, codul de producție al răspunsului. Prin interpretarea acestor coduri se obțin adresele de acces la informațiile din fișierele bazei de date, de unde acestea sunt citite și apoi afișate, generându-se astfel răspunsul așteptat. În cazul interogării "libere" permise în cadrul sistemului, generarea răspunsurilor devine mai complicată. În acest caz, identificarea programelor care satisfac anumite cerințe implică verificarea succesivă a includerii codului cerințelor (din coloanele corespondente) în articolele cod ale

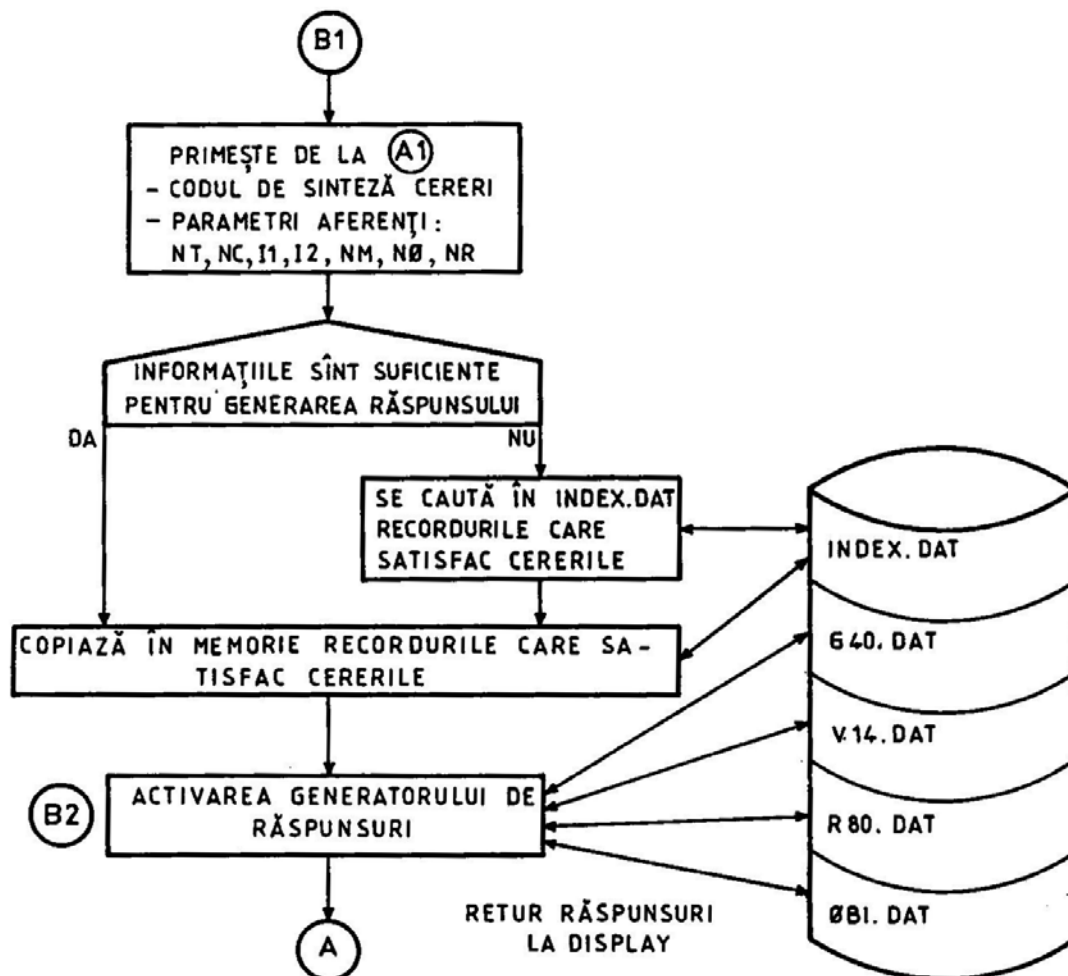


Fig.23 Funcțiile și structura blocului de inferență în sistemul BPC

programelor din fișierul GEN.DAT. Verificarea aceasta este realizată de mai multe subrutine, printr-o tehnică de mascare, utilizând funcția de produs logic *IAND* din Fortran/2 IBM. Acest proces de căutare este consumator de timp și la data realizării sistemului constituia neajunsul major al sistemului BPC.

Generarea sistemului expert BPC implica completarea (umplerea) bazei de cunoștințe cu informațiile care urmau a fi prelucrate la formularea răspunsurilor pe baza cererilor specificate de utilizator și corelarea acestor informații (referitoare la un program în curs de catalogare) cu înregistrarea concomitentă a acestuia în fișierul GEN.DAT (INDEX.DAT) acesta din urmă constituind de fapt cheia (codul sintetic al regulilor de producție) pe baza căreia puteau fi elaborate răspunsuri adecvate întrebărilor puse. Pentru realizarea acestor obiective am elaborat pachetul de programe GSBPC (Generare Sistem BPC) având două componente principale: CIF și GSB. Acest pachet suplimentar totalizează cca 650 de linii de cod (scrise în Fortran/2 IBM) fiind destinat în exclusivitate administratorului de sistem.

Din păcate și acest sistem expert a fost dezvoltat și implementat doar până la faza de prototip funcțional, baza de cunoștințe fiind încărcată doar cu cca. 10-15% din caracteristicile programelor de calcul din catalogul IPCT. Pentru completarea bazei de cunoștințe a fost conceput un formular (chestionar) tip grilă, cuprinzând cunoștințele necesare referitoare la programele de calcul existente la data respectivă, acesta fiind prezentat tot în ANEXA 1. Pentru organizarea informațiilor referitoare la programele de calcul existente, au fost reținute 32 clase de tipuri de programe (dintre care au fost nominalizate efectiv 30 clase), fiecărei clase corespunzându-i câte o grupă de subclase cu 5-10 variante posibile. În prototipul dezvoltat numărul claselor era limitat la 32, iar al subclaselor la 255, deci în principiu sistemul permitea catalogarea până la $32 \times 255 = 8160$ tipuri distincte de programe. Lipsa de fonduri, documentarea dificilă datorită schimbărilor sociale și economice de la începutul anilor '90 (modificările de adrese, transformările din cadrul societăților comerciale etc.) precum și volumul foarte mare de muncă necesită m-au obligat din păcate la sistarea dezvoltării și perfecționării acestui produs.

4.1.2 Instrumente pentru proiectare asistată de calculator

Din anul 1995, având deja o pregătire teoretică consistentă și colaborând cu un colectiv foarte entuziast și tânăr din cadrul unei firme de proiectare clujene (Consort Construcții s.r.l.), am încercat realizarea unei suite de aplicații în ideea automatizării cât mai extinse a procesului de proiectare. Era o problemă foarte incitantă nu numai din punctul de vedere al activității de proiectare propriu-zise desfășurate în cadrul firmei, dar mai ales din cauza diversității platformelor de lucru și a echipamentelor existente. Astfel, modelarea arhitecturală (ca și redactarea memoriilor aferente și a documentațiilor) se realiza pe un calculator Apple-Macintosh PowerPC 6100/60, pe când calculele structurale și planșele de rezistență destinate execuției erau realizate pe PC-uri compatibile IBM (486DX2/66). Deci prima problemă a fost realizarea unei rețele prin interconectarea echipamentelor (diferite calculatoare, imprimante, plotter), care să permită transferul și schimbul

corespunzător de informații. Având în vedere valoarea investiției inițiale în echipamente și programe, patronul firmei a lansat ideea unei mese rotunde pentru a discuta și analiza toate aspectele prezente și previzibile ale activităților din cadrul firmei. Așa s-a născut și ideea de a proiecta proiectarea.

După cum opina și Octavian Lipovan într-un articol publicat în *CADreport*-ul nr.2 din 1997, foarte mulți consideră că procesul de proiectare se aseamănă cu un *puzzle*. Dacă ne gândim că de-a lungul activității de proiectare de cele mai multe ori manevrăm componente sau entități date, chiar dacă acestea pot avea caracter schimbător dinamic (deoarece suferă modificări și transformări de-a lungul procesului), atunci afirmația precedentă poate fi considerată în mare măsură corectă. Astfel forma finală a proiectului ar fi constituită din *înghețarea* acestor componente într-o stare și într-un moment considerate optime din punctul de vedere al cerințelor cunoscute. După opinia lui O.Lipovan, mecanismul generic în cazul proiectării ar fi următorul:

- Pornirea procesului este asigurată de mulțimea unor date de la care se poate ajunge la un întreg.
- Acest întreg se poate contura în mai multe variante, dar din aceste variante merită a fi alese doar câteva, care corespund cerințelor formulate (în urma unor comparații adecvate).
- Următorul pas este studierea părților. Fiecare întreg ales va parcurge un drum invers, fiind descompus până la nivelul unor părți elementare.
- Fiecare entitate astfel obținută va suferi un proces de optimizare, atât individual cât și față de celelalte componente ale aceluiași întreg.
- Recompunând părțile astfel optimizate, întregurile rezultate vor fi comparate ca variante. Acest ciclu se repetă până când va rămâne doar o variantă (corespunzătoare).
- Ulterior, acest întreg ales poate fi supus unor noi cicluri de analiză și optimizare, până când rezultatul va satisface toate condițiile inițiale și cele formulate în cursul procesului.

Toți acești pași presupun legături interdisciplinare, temporizări adecvate și coordonare, deoarece fiecare proiect se compune din mai multe părți și faze, iar acestea presupun diverse domenii de specialitate. În cadrul fiecărei faze de proiect există însă un domeniu de specialitate ce joacă rol central. De regulă, *dirijorul* proiectului apare din acest domeniu, printre sarcinile acestuia fiind asigurarea fluxului corespunzător de informații, coordonarea specialităților, temporizarea fazelor, analizelor și a calificativelor ce se referă la întregul proiect. Aceste observații au un caracter general, dar sunt valide și în cazul proiectării construcțiilor, indiferent de numărul persoanelor care concură la realizarea acestei activități.

Care este însă rolul tehnicii de calcul în procesul amintit? După cum se știe, calculatoarele personale reprezintă azi instrumente fundamentale foarte reușite pentru trei activități umane esențiale: jocul, munca și comunicarea. Foarte mulți consideră însă că adevăratul

scop al utilizării calculatoarelor ar fi viteza pe care ar asigura-o aceste instrumente în derularea diverselor activități. Mai puțini își dau însă seama de faptul că dacă calculatorul oferă posibilitatea rezolvării mai multor probleme într-un interval de timp dat (față de metodele manuale), atunci volumul respectiv de lucru trebuie efectuat corespunzător. Deci, pe de o parte utilizarea echipamentelor, nu tocmai ieftine, trebuie să fie rentabilă, pe de altă parte trebuie să ținem pasul cu aceste echipamente. În proiectarea construcțiilor mai sunt însă și alte considerente care trebuie luate în seamă. Există cerințe (cum ar fi calculul structurilor spațiale extinse, plimbările virtuale prin modelele tridimensionale, simularea jocurilor de lumini și umbre în studiile arhitecturale etc.) ale căror îndeplinire nu se poate imagina fără aportul sistemelor moderne de calcul. Dar punctul de vedere probabil cel mai important rămâne calitatea proiectului și a datelor: ușurința realizării copiilor, edițiile în exemplare multiple și în diferite variante, modificările eventuale putând fi urmărite ușor.

O problemă foarte importantă era și alegerea colaboratorilor. Dacă pentru partea de arhitectură erau deja două persoane alese, chiar înainte de materializarea societății amintite, pentru partea structurală și tehnologică am încercat găsirea unor colaboratori în stare să folosească eficient resursele materiale existente. Astfel, pentru tehnologie și organizare s-a apelat la serviciile unui cadru didactic experimentat de la U.T.C.N. care ținea și cursuri de utilizarea calculatoarelor, iar pentru partea structurală, precum și pentru instalații și echipamente de construcții, am cooptat tineri absolvenți și doctoranzi din cadrul instituției amintite, având în vedere și experiența mea didactică. Având în vedere pregătirea informatică a acestor ingineri, s-au ales în primul rând pachete de programe cu care erau deja familiarizați. Deja după primele luni de activitate, înmulțirea comenzilor și a proiectelor realizate, împreună cu abilitatea și predispoziția către nou a colectivului format, făcea evidentă necesitatea extinderii uneltelor (atât sub aspectul echipamentelor cât și sub aspectul softului). Deoarece nu foloseam un pachet integrat de programe pentru proiectare, iar proiectele realizate erau foarte variate (de la căsuțe de vacanță până la hotel cu patru stele, centre comerciale și de afaceri, spital etc.), am încercat să extindem capacitățile pachetelor de programe utilizate, prin dezvoltarea unor instrumente proprii, care să ne ușureze cât mai mult activitatea. Nimeni nu se entuziasma când era vorba de realizarea desenelor de execuție sau a detaliilor necesare, așa că primul pas a fost crearea unor biblioteci de obiecte și desene parametrizate sub ArchiCAD și Autocad. În paralel, folosind facilitățile oferite de tabelele de calcul moderne, au fost concepute și dezvoltate aplicații necesare automatizării calculelor pentru instalațiile de alimentare cu apă, încălzire, iluminare etc. După acestea au urmat instrumente complementare pentru automatizarea dimensionării diverselor elemente structurale, precum și pentru facilitarea realizării desenelor. Întrucât planurile de armare și de cofrare erau realizate sub Autocad, majoritatea acestor instrumente au fost scrise în AutoLISP. Scopul urmărit era proiectarea cât mai facilă, într-un timp cât mai redus, cu cât mai puține erori. Astfel ni se ivea ocazia de a considera mai multe variante în cadrul aceluiași proiect, împreună cu alegerea mai ușoară a versiunii dorite. De exemplu, la același obiectiv puteau fi luate în seamă diverse modalități de încălzire, comparând costurile de realizare și de întreținere chiar pe durate extinse sau diverse stratificații pentru planșee, pereți etc., comparând avantajele și dezavantajele derivate din caracteristicile structurale, higrotermice, și așa mai departe.

Ciclul de proiectare se desfășura în felul următor: după clarificarea temei, modelul tridimensional al obiectivului era realizat de către compartimentul de arhitectură. Din ArchiCAD se transfera modelul sub formă de fișiere DWG în Autocad, la compartimentul de structuri, pentru a realiza predimensionarea și modelarea structurală ținând cont și de observațiile celor de la compartimentul de instalații. În timp ce compartimentul de structuri se ocupa cu analiza structurală, cei de la compartimentul de instalații realizau proiectul de specialitate (alimentarea cu apă, canalizarea, încălzirea etc.). În urma analizei structurale se realizau planșele de rezistență necesare execuției, după care modelul actualizat era retransferat arhitecților care se ocupau de clarificarea detaliilor, generarea desenelor arhitecturale și de realizarea materialelor pentru prezentări: imagini fotorealiste, secvențe animate etc. (în măsura în care existau cerințe în acest sens). Pe baza modelului actualizat în urma analizei structurale și după clarificarea detaliilor, compartimentul de tehnologie-organizare întocmea documentațiile necesare (antemăsurătoare, proiect de organizare etc.). Deși inițial realizarea instrumentelor ajutătoare avea doar scopul autometizării proiectării, pentru a reduce efortul și durata necesare realizării unor proiecte de calitate (activități la care sub îndrumarea mea și cu concursul meu participa aproape tot colectivul firmei, dintre care aș remarca în mod special inginerii: Lucian Ștefan, Attila Téglás, Cosmin Chiorean, Vlad Anastasiu și Bogdan Mert), nu este exclusă dezvoltarea acestora cu aspect comercial (având în vedere interesul stârnit prin prezentarea lor la diverse expoziții și conferințe). Din această cauză, dintre aceste instrumente ași aminti doar câteva la ale căror dezvoltare am contribuit și eu în mod direct. Acestea (în variante mai perfecționate) împreună cu alte module similare, dezvoltate ulterior de către inginerul Attila Téglás, sunt grupate în prezent în pachetul ASCAD (figura 24), destinat proiectării structurale sub Autocad, neexistând însă deocamdată o versiune comercială. Iată deci modulele la ale căror concepere și dezvoltare am lucrat:

- modul interactiv pentru generarea desenului de cofrare la planșee din beton armat. Programul ține cont de grinzile, centurile aferente și permite generarea automată a secțiunilor caracteristice.
- modul interactiv pentru dimensionarea planșeelor din beton armat și pentru generarea planșei de armare. Din punctul de vedere al intrărilor se poate opta între citirea datelor (caracteristicile fizico-mecanice ale elementelor, înfășurătoare de momente încovoietoare, înfășurătoare de forțe tăietoare) din fișierele rezultate în urma calculelor structurale efectuate cu pachetul ROBOT, sau un calcul static simplificat, sau introducerea manuală a datelor (chiar cu aproximări). Calculul static efectuat în cadrul acestui modul se realizează după două direcții, considerând câte o fâșie continuă cu lățimea de 1 metru, chiar și cu mai multe scheme de încărcare. Pe baza înfășurătoarelor de eforturi se determină cantitatea necesară de armătură, se alege în mod automat (dar se permite și intervenția utilizatorului) armăturile corespunzătoare și se generează planșa de armare conținând și secțiunile și extrasul de armătură. În versiunea actuală utilizatorul trebuie să intervină cu mouse-ul pentru a determina poziția diverselor elemente din planșa generată automat (ghidând astfel din punct de vedere estetic generarea desenului).

- modul interactiv pentru dimensionarea grinzilor continue din beton armat cu secțiune dreptunghiulară și generarea planșelor de armare. Numărul deschiderilor depinde doar de memoria disponibilă (nu este limitată prin program). Se poate utiliza foarte ușor și pentru dimensionarea grinzilor secundare sau a rețelilor de grinzi și pentru obținerea automată a desenului de armare. Ca și modulul precedent, se poate opta între citirea automată a datelor de intrare din fișierele furnizate de către pachetul ROBOT sau calculul static imediat (interactiv). Determinarea cantităților necesare de armătură și alegerea barelor corespunzătoare se face ținând cont doar de înfășurătoarele momentului încovoietor și ale forțelor tăietoare, dar în dezvoltările următoare, colegul Attila Teglás dorește să introducă și efectul torsiunii. În cazul forței tăietoare de lângă reazeme, la calculul călăreților se poate opta pentru bare separate sau pentru ridicarea unor bare de armătură (armătura înclinată se consideră în cel mult două mănunchiuri separate). La alegerea automată a barelor de oțel-beton, programul lua în considerare condițiile de capăt la fiecare deschidere pentru verificarea lungimilor corespunzătoare de ancorare. Generarea și reprezentarea secțiunilor la mijlocul deschiderilor, lângă reazeme și în vecinătatea schimbărilor armăturilor, precum și generarea extrasului de armătură erau efectuate automat. Din punctul de vedere al utilizatorului, dimensionarea grinzilor nu este altceva decât un joc grafic (chiar schimbând calitățile materiale sau dimensiunile secțiunilor de ori câte ori se dorește), în care nu trebuie decât să marcheze (făcând clic cu mouse-ul) pozițiile diverselor entități incluse în planșa de armare.
- modul pentru dimensionarea și desenarea automată a stâlpilor din beton armat cu secțiune dreptunghiulară. Calculul secțiunilor era realizat în două etape. Mai întâi se efectua o dimensionare după momentul încovoietor maxim, considerând efortul axial și forța tăietoare corespunzătoare (din gruparea specială), apoi se efectua o verificare a secțiunii după efortul axial maxim, considerând momentul încovoietor și forța tăietoare corespunzătoare (din gruparea fundamentală). Lungimea de flambaj și zveltețea erau considerate după STAS, coeficientul μ fiind determinat printr-un calcul iterativ convergent. Dimensionarea secțiunii se realiza după două direcții, iar armarea rezultată putea fi dispusă simetric sau nu (după opțiunea utilizatorului). Ulterior, tot colegul Teglás a perfecționat acest modul, în prezent fiind capabil să dimensioneze și stâlpi cu secțiune dreptunghiulară variabilă. Rezultatul, ca și la modulele anterioare, se materializa prin planșa completă de armare, cuprinzând vederile, secțiunile și extrasul de armătură.

Printr-o colaborare ivită oarecum întâmplător cu firma LINTOP (în urma unor discuții la expoziția de construcții AMBIENT'97 organizată la Cluj-Napoca), reprezentanți ai concernului suedez LINDAB, ne-am hotărât să concepem un sistem interactiv pentru evaluarea materialelor și a costurilor necesare realizării unei hale LINDAB tip standard. Ideea era de a realiza un program care să poată fi utilizat de către orice doritor, chiar și fără pregătire informatică elementară, pe bază de tarif orar. Astfel, un utilizator care știe dimensiunile orientative ale halei dorite, în câteva secunde putea obține vederea

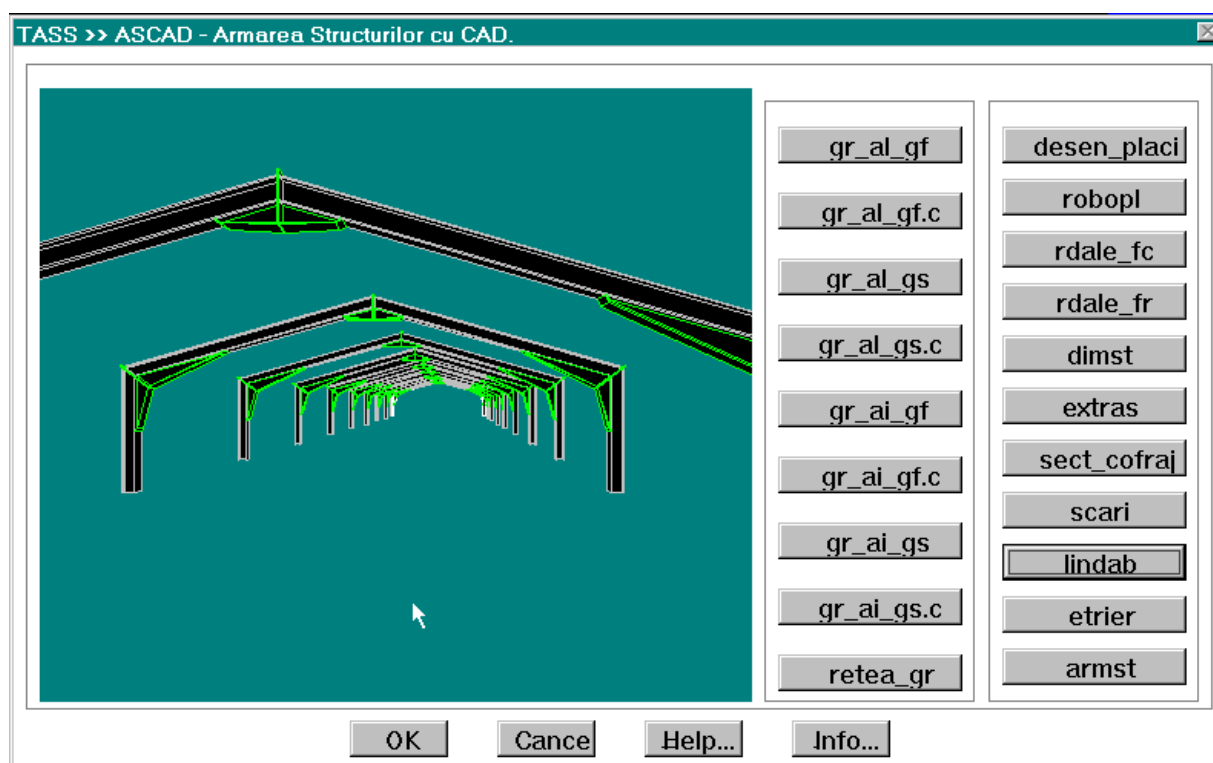


Fig.24 Ecranul de prezentare al pachetului ASCAD cu modulele amintite integrate (inclusiv prototipul pentru hale standard LINDAB)

tridimensională a halei compuse pe baza opțiunilor exprimate, beneficiind în aceeași timp și de un extras de piese și materiale (și implicit de costurile referitoare la structură și închideri). Pe lângă aspectul de sistem expert (îndeplinind rolul specialistului menit să efectueze evaluarea și proiectarea unei hale LINDAB standard), programul era menit să ușureze și procesul de evaluare realizat de către specialiștii firmei LINTOP, timpul necesar întocmirii unei oferte, și să elimine pe cât posibil erorile de evaluare. În ANEXA 2 sunt prezentate sursele dezvoltate pentru această aplicație. Din păcate, sistemul a fost dezvoltat doar parțial și nu a putut fi deocamdată finalizat, în primul rând din motive financiare. Am însă în intenția de a dezvolta acest sistem expert, dacă vor apare solicitări în acest sens, folosind tehnici și instrumente informatice mai moderne.

4.2 DEZVOLTAREA UNUI SISTEM BAZAT PE CUNOȘTINȚE, DEDICAT CONSTRUCȚIILOR

4.2.1 Conceperea și realizarea sistemului AIC Info, structurarea cunoștințelor

Participând la diverse manifestări științifice, vizitând diferite expoziții, am ajuns la concluzia că popularizarea serviciilor și mărfurilor prin broșuri tipărite și pliante are mai multe inconveniente. Cel ce dorește informații pertinente și actuale, se poate trezi cu sacose pline de tipărituri care pe lângă faptul că sunt greu de transportat, sunt și incomod de manevrat (trebuie sortate, ordonate, cautarea printre ele este destul de anevoioasă etc.). Pe de altă parte, mai mulți colegi din domeniul construcțiilor (în special comercianți și producători de materiale de construcții, dar nu numai) se întrebau cum să-și popularizeze produsele ca să fie eficienți. Reclamele inserate pe paginile cotidianelor sau ale diverselor

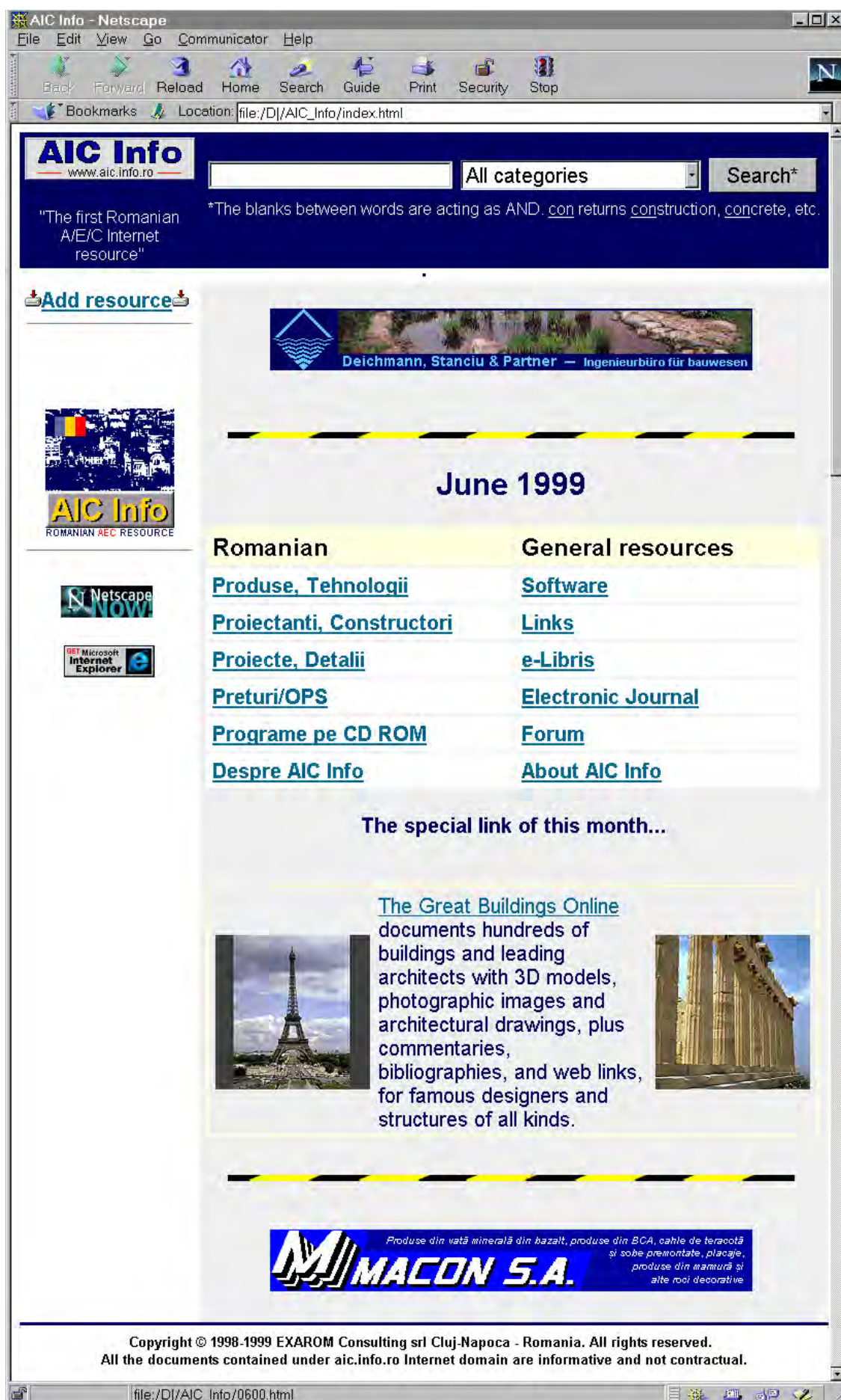


Fig.25 Pagina de prezentare a sistemului AIC INFO (vers. 2.0.10 - prima versiune funcțională cu bază de date și căutare)

publicații nu pot conține decât informații condensate, iar tipărirea și răspândirea unor foi volante are un efect foarte incert, ca și anunțurile prin mas-media. Din aceste discuții se contura cu insistență aceeași întrebare: Cum putem să ne adresăm constructorilor și celor interesați de construcții, cu efort minim și eficiență maximă? Prietenul și colegul meu, Dan Turda mi-a propus o soluție posibilă pentru ambele probleme amintite: transpunerea documentelor de specialitate (broșuri, pliante, fișe tehnice, normative) în format electronic și distribuirea acestora la diverse manifestări. Ideea nu era nouă, am mai participat la asemenea proiecte, însă în contextul tendințelor de informatizare din țară și de pe plan mondial, infrastructura și instrumentele disponibile ne ofereau perspective noi. Entuziasmați de caracterul de pionierat din punct de vedere instrumental al unui asemenea proiect, precum și de posibilele efecte informaționale și educaționale, am considerat că dacă ne-am angaja într-un asemenea proiect, dedicând suficiente resurse, în doi-trei ani am putea realiza un sistem funcțional destul de util. Experiența acumulată de către Dan Turda (în principal prin contribuția avută la dezvoltarea unui sistem comercial internațional: *X-ROADS Crossroads for Business* - prin colaborarea dintre firmele EXAROM Consulting din România și EXA-Management din Franța), combinată cu pregătirea mea și cu interesul meu pentru dezvoltarea unor sisteme "inteligente", reprezentau principalul atu pentru noi. Rezultatele de până azi par să ne dea dreptate. Deși la început trebuia să facem față unei oarecare concurențe, existând firme cu potențial material net superior față de noi, care încercau realizarea unor colecții de resurse similare în domeniul construcțiilor, în prezent suntem singurii din țară care am reușit dezvoltarea unui asemenea site dedicat.

Colaborarea mea propriu zisă cu Dan Turda a început după înființarea unui domeniu pe internet și înregistrarea acestuia sub numele de AIC INFO (AIC = Arhitectură și Inginerie în Construcții) cu adresa: **<http://www.aic.info.ro>**. Pe internet funcționau deja câteva sisteme de informare destinate construcțiilor pe plan internațional, grupate oarecum în jurul noțiunii *AEC (Architecture, Engineering & Construction)*, azi numărul acestor domenii fiind impresionant (în ANEXA 4 sunt amintite câteva dintre aceste domenii). La noi în țară existau doar niște tentative timide, mai degrabă sub forma unor colecții de adrese ale firmelor de construcții participante la diverse manifestări promoționale, expoziții. La început, sistemul AIC Info a fost conceput ca o colecție de fișe tehnice despre materiale și servicii din domeniul construcțiilor. Accentul primordial era pus pe diseminarea cât mai largă a informațiilor disponibile și accesibile din domeniul construcțiilor în țară, nu numai în scopul unei informări corecte și eficiente, dar și în scop educațional și de popularizare. Deci situl se adresează atât specialiștilor și profesioniștilor din domeniul construcțiilor cât și publicului larg doritor de cunoștințe specifice.

Deoarece echipamentele și sistemele informatice în uz erau foarte diverse (azi acest lucru este și mai adevărat), de la cunoscutele PC-uri cu sisteme de operare ca DOS sau Windows, până la stațiile grafice Sun cu sistemul de operare Solaris, urmărind o compatibilitate cât mai largă, am decis redactarea conținutului accesibil din domeniu sub forma unor pagini și documente scrise în HTML (Hyper Text Markup Language). Pentru încărcarea și consultarea unor asemenea documente era suficientă existența unui browser, iar asemenea

programe de navigare pe internet erau foarte accesibile și destul de răspândite pentru toate sistemele de operare aflate în uz. Observând explozia produsă de introducerea sistemelor GSM în domeniul telecomunicațiilor, ne așteptam la o cvasi-generalizare a accesului și la internet, cunoscând avantajele și facilitățile oferite de web. Totuși, interesul manifestat (în ceea ce ne interesa în principal: de către firmele din domeniul construcțiilor) pentru folosirea rețelei mondiale ca suport era foarte slabă, atât din cauza ignoranței cât mai ales din cauza condițiilor tehnice și financiare relativ precare. Pentru a pătrunde cât mai larg cu sistemul conceput de noi în primul rând în pătura constructorilor, am încercat distribuirea informațiilor din colecția noastră și prin imprimarea de CD-ROM-uri care erau expediate prin poștă. Având în vedere cheltuielile necesare realizării unei asemenea producții, sistemul a trebuit transformat într-o oarecare măsură, primind și un caracter comercial. Practic, CD-ROM-ul realizat putea fi folosit și ca suport pentru marketing, nu numai ca suport informațional (având în vedere aria de răspândire, ușurința utilizării și volumul disponibil pentru stocarea informațiilor). În paralel, am dezvoltat și partea de client-server de pe internet, lărgind și categoriile resurselor oferite. Pentru acoperirea cheltuielilor de dezvoltare și de întreținere am fost nevoiți să concepem și un sistem de tarificare. Astfel, cei care doreau publicarea unor materiale comerciale pe situl nostru, trebuiau să achite un abonament anual în funcție de volumul ocupat și gestionat de către noi. Cei care doreau dezvoltarea unor instrumente sau aplicații dedicate, beneficiau de tarife negociabile în funcție de condițiile și cerințele specifice date. Publicarea unor materiale educaționale, precum și consultarea sistemului, erau și sunt gratuite pentru oricine. Nu pot să nu amintesc de ajutorul primit de la firma LOGIC Telecom S.A. la pornirea sistemului (în special prin contribuția domnului Traian Opincariu) și de sprijinul susținut din partea colectivului de la Centrul de Informatică Aplicată din Cluj-Napoca (menționând în mod special contribuțiile deosebite ale domnilor Ioan Homorodean și George Comșulea).

Natura informațiilor din domeniul construcțiilor este foarte variată. Există descrieri textuale, tabelare, grafice etc. Unele au caracter informativ altele reprezintă reglementări sau recomandări, și așa mai departe. Deci, pe lângă tipul informațiilor, contează foarte mult și contextul sau modalitatea de prezentare. Colectarea și modelarea acestor informații se realizează destul de greu, dacă se dorește obținerea unui sistem informativ cu o arie de acoperire relativ mare. La început, informațiile erau culese, modelate și prelucrate destul de anevoios. Introducerea unor fișe (resurse) noi în sistem presupunea efectuarea manuală a unor schimbări și transcrierea paginilor deja existente pentru realizarea și menținerea legăturilor. Dacă la demararea sistemului (în data de 13 februarie 1998) conținutul inițial era de circa 1Mb, datorită dezvoltării continue atât în diversitate cât și în volum în luna martie din anul 1999, spațiul ocupat a atins deja 200Mb. Mărindu-se numărul fișelor din colecție, considerând și pretențiile noi din ce în ce mai numeroase ale firmelor, împreună cu ideile noastre referitoare la dezvoltarea sistemului, modalitatea inițială de gestionare nu mai era viabilă, așa că am decis transformarea întregului sistem într-o adevărată bază de cunoștințe. În cadrul acestei inițiative trebuia să ținem cont și de faptul că AIC Info era deja un sistem integrat, cunoscut și utilizat, oferind simultan accesul la informație, structurarea ei și acțiunile necesare pentru a asigura vizibilitatea sitului în condițiile în

care numărul de "pagini" internet era foarte mare. Având o varietate mare de informații (descrieri tehnologice, tabele cu caracteristici, imagini etc.) am încercat realizarea unei topologii care să permită și gestionarea mai ușoară a sistemului. Sistemul a devenit astfel un adevărat repertoriu de resurse internet. Formatul HTML permite reprezentarea informațiilor prin modelarea cunoștințelor în limbaj natural nestructurat (text), tabele, imagini, dar și o oarecare modelare a metacunoștințelor prin inserarea de coduri (*script-uri*), realizarea unor legături statice (apelarea unei pagini diferite sau a unei adrese din pagina curentă) sau dinamice (generarea unei pagini noi) etc. În același timp, sursa paginilor poate fi creată chiar și cu editoarele de text cele mai simple, respectând sintaxa limbajului HTML.

După o reanaliză profundă și atentă a sistemului, am reușit să realizăm o clasificare a resurselor disponibile ținând cont de ciclul de viață al construcției de la concepție până la demolare, respectiv de momentul și complexitatea produsului sau tehnologiei considerate. Această grupare a resurselor este ilustrată în figura 26. Pe scurt, motivația acestei structuri este următoarea:

În mod normal orice obiectiv nou trece mai întâi printr-o fază de concepție, avizare etc., ceea ce ține de fapt de grupul denumit *Servicii*. În următoarea categorie am grupat resursele referitoare la utilaje, unelte, scule și echipamente necesare execuției (urmând ca în viitor să fie incluse aici și cele necesare proiectării etc). Partea materială a construcției este deja subordonată și nivelului de complexitate al entităților, astfel după *Materiale și materii prime* urmează elementele mai complexe grupate în *Elemente de construcție, subansamble și stadii fizice*. Rețelele, echipamentele din construcții și dotările sunt la rândul lor împărțite în două categorii: interioare și exterioare, această diferențiere fiind necesară și din punctul de vedere al importanței, coroborate cu complexitatea. Cea mai complexă categorie este reprezentată de grupa *Construcții integrale* (obiective "la cheie"). Conform ciclului de viață al construcției urmează grupa *Întreținere și reparații*, iar în final *Demolare, recuperare și reciclare*.

În cadrul fiecărei categorii am definit subcategorii (pe trei niveluri de ierarhizare). În urma acestor structurări am reușit să codificăm toate fișierele existente în așa fel încât accesarea lor să se poată realiza fie prin parcurgerea categoriilor și subcategoriilor (sub formă arborescentă), fie prin căutare directă folosind cuvinte cheie. Prin dezvoltarea unor instrumente, cu ajutorul colegilor de la Centrul de Informatică Aplicată, aspectul sistemului și gestionarea resurselor a devenit unică, nemaexistând diferențe din punctul de vedere al fișelor între conținutul domeniului internet și al CD-ROM-ului distribuit.

În timp, sistemul a suferit mai multe transformări (în special estetice, dar și de concepție și conținut), încercând să-i asigurăm un aspect cât mai clar și cât mai simplu. Deși sistemul realizat este fiabil, funcționează corect (și după opinia utilizatorilor) având timpi de răspuns foarte buni, totuși nu este încă perfect. De exemplu, clasificarea actuală nu suportă referințe încrucișate, acestea fiind înlocuite pentru moment printr-o facilitare de căutare booleană (operatorul AND fiind singurul considerat), care permite obținerea rezultatelor



Fig.26 Grupele de clasificare ale resurselor din categoria *Produse/Tehnologii*

după cuvinte cheie și restrângerea numărului acestora în funcție de problema vizată. Pentru a corecta oarecum acest aspect, am păstrat/refăcut anumite legături (din paginile inițiale) între diversele categorii, sub forma unor butoane grafice.

Sistemul a cunoscut o dezvoltare considerabilă mai ales în ultimul an, fiind apreciat atât în țară cât și peste hotare. În afară de critici și laude am primit și oferte de colaborare cu firme și sisteme deja impuse pe plan mondial. Evitând însă diluarea conținutului domeniului nostru, am preferat doar inserarea unor legături către serviciile oferite. Deasemenea, pentru a limita pe cât posibil costurile de dezvoltare și de întreținere, am preferat utilizarea unor resurse cât mai accesibile (ținând cont bineînțeles și de legislația în vigoare). În prezent în spatele sistemului stă o soluție client-server în care serverul este pachetul Apache instalat pe sistemul de operare Linux, iar clientul poate fi orice browser grafic (Netscape Communicator/Navigator, MS Internet Explorer, Opera etc.). Având în vedere codurile JavaScript pe care le utilizăm, se recomandă ca browserul să fie cel puțin versiunea 4.

Serverul bazei de cunoștințe este de fapt un server de baze de date (POSTGRES, tot sub Linux), iar metacunoștințele sunt formulate utilizând în principal JavaScript și *cgi* (common gateway interface - o facilitare prin care serverul Apache lansează în execuție programe locale), exploatând însă și facilitățile oferite de HTML. Interogarea bazei de date se face prin SQL, via PERL. Căutarea este efectuată de POSTGRES, interogarea SQL fiind transmisă via front-end *psql*. Rezultatele căutării se stochează temporar în memorie (în caz de depășire se înregistrează în memoria virtuală de pe disc) și sunt tratate printr-un alt program care le formatează într-un aranjament grafic prezentabil. Paginile afișate sunt generate practic din trei părți, prima și a treia parte definind aspectul estetic, iar partea din mijloc definind conținutul. Astfel, schimbarea înfățișării interfeței sistemului (*skin-change*) se poate rezolva foarte simplu, independent de conținut, iar schimbarea conținutului paginii nu afectează aspectul acesteia. În figura 27 se poate vedea noul aspect al paginii de prezentare, generată după modalitatea amintită.



Fig.27 Pagina de intrare în sistemul AIC INFO (vers. 2.1.1, noiembrie 1999), generată dinamic pe baza resurselor din baza de cunoștințe

Baza de cunoștințe este alcătuită de fapt din descrieri ale resurselor găsite în categoriile suportate. Ea este încărcată și actualizată prin parcurgerea siturilor web găsite în internet și evaluarea relativă a suitabilityății înregistrării unei descrieri în baza de date. Sistemul ar putea beneficia de autoîncărcare, lărgindu-și singur volumul de cunoștințe, printr-un motor corespunzător de căutare (*spider*). Această căutare s-ar putea realiza relativ simplu, folosind cuvinte cheie și chiar aproximări. Din păcate, o asemenea facilitate nu ne surâde (deși nu este greu de realizat) pentru că ar crește la valori foarte mari traficul derulat, atrăgând consecințe financiare insuportabile.

La baza modelării cunoștințelor stă conceptul obiectual. Existând varietatea menționată privind atât tipul cât și contextul informațiilor din construcții, modelul bazat pe obiecte este completat și cu cadre (atât la propriu cât și la figurat, deoarece pentru a păstra o anumită compatibilitate cu browserele anterioare încă mai utilizăm generarea și prezentarea informațiilor și în *frame*-uri prin HTML).

Din punctul de vedere al modelării cunoștințelor și al funcționării, sistemul poate fi privit și ca un ansamblu virtual de cuburi ordonate, cu fațete determinate ca și tip de conținut.

Pentru a ilustra mai sugestiv funcționarea și eficiența acestui model, recurg la următoarea exemplificare:

- Să presupunem că dorim obținerea unor informații de natură grafică din sistem. În acest caz, după primirea cererii se instanțiază toate clasele existente în funcție de argumentul grafic, pentru obținerea rapidă a informației. Folosind analogia cu sistemul de cuburi, lansarea unei paradigme pentru obținerea unei informații grafice determină rotirea tuturor cuburilor în așa fel încât fețele vizibile să fie cele cu conținut grafic. Din această mulțime de selecție rezultată, urmează să se realizeze alegerea elementelor corespunzătoare, după care urmează compunerea și aranjarea paginilor care vor fi oferite utilizatorului.

Structura sistemului este destul de complexă, putând fi percepută nu numai pe categorii de obiecte ci și pe straturi (*layer-e*), după cum am ilustrat în figura 28. Astfel, peste stratul fundamental care conține cunoștințele propriu zise (cuburile amintite), există un strat de reguli și procese care interacționează cu stratul precedent, de cele mai multe ori într-un mod transparent pentru utilizator. În cadrul acestui strat se găsesc pe lângă filtre, *plug-in-uri*, motoare de interpretare etc., și aplicații (*script-uri* și *applet-uri*) dezvoltate de către noi în scop instrumental, unele chiar la cererea unor clienți. Peste acest al doilea strat, se găsește un al treilea, conținând informațiile necesare manevrării unora dintre componentele stratului anterior (fiind mai sărac în conținut și formă, ar fi cam deplasat să spun că ar conține cunoștințe). În acest strat se găsesc, pe lângă explicațiile și ajutoarele online, și recomandările privind instalarea și configurarea, sau activarea unor componente soft. Toate acestea țin de folosirea cât mai facilă a sistemului, fiind subordonate contextual oarecum interfeței utilizator.

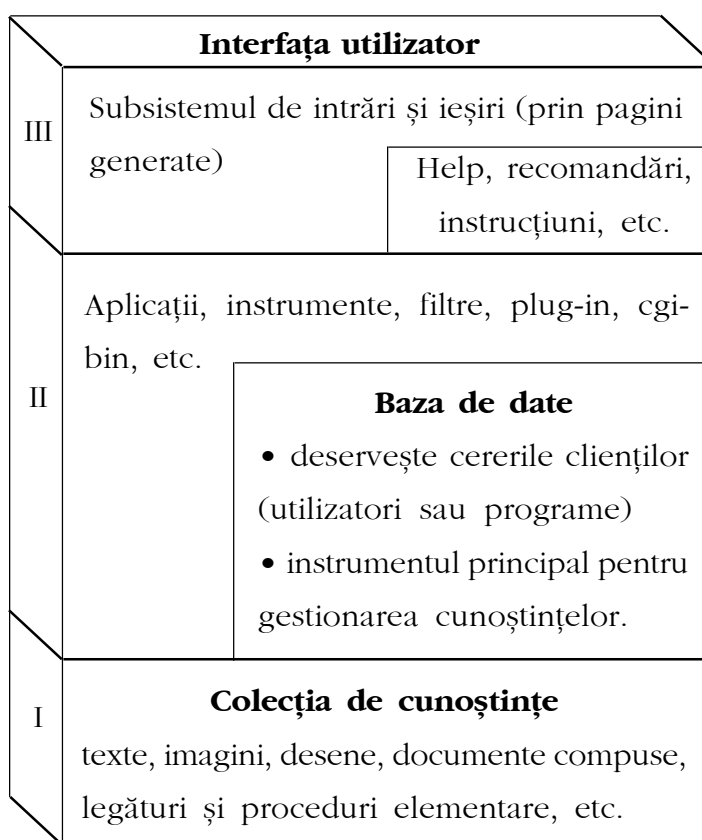


Fig.28 Stratificarea sistemului AIC Info

4.2.2. Componentele dezvoltate și funcțiunile realizate

În prezent interfața sistemului permite accesarea tuturor părților destinate consultării, fie prin legături directe din cadrul paginilor generate, fie prin legături inserate care se generează în funcție de context. Acest subsistem permite și comunicarea între utilizatori

și clienți, prin intermediul resurselor oferite de către serverul AIC Info. Pentru întreținerea bazei de cunoștințe există o altă interfață cu diferite instrumente (pentru introducerea, modificarea, filtrarea și verificarea, elementelor), restricționată din punctul de vedere al accesului. Prin această parte clienții doritori își pot gestiona și singuri anumite elemente încărcate. Componentele oferite de către sistemul AIC Info sunt ilustrate în figura 29. Cele care sunt marcate cu umbră reprezintă *resurse generale*, în sensul în care nu se adresează doar utilizatorilor români, marea majoritate a informațiilor de aici fiind în limba engleză (în *Forum* există suport și pentru limbile: germană, franceză, italiană și maghiară). Componentele ilustrate fără umbră conțin informații în cea mai mare parte în limba română; există însă și câteva excepții (respectând contextul prezentării dorite de anumite firme). Baza de cunoștințe se sprijină deci în primul rând pe o bază de date (exploatăată prin soluția client-server implementată), oferind resursele informaționale prin următoarele componente:

- **Biblioteca de Produse și Servicii pentru Construcții** (BPSC) care constituia inițial inima sistemului și era realizată din fișele tehnice amintite. În prezent ea conține informații referitoare la produse, tehnologii precum și despre furnizorii acestora, strânse în primul rând în cele nouă categorii principale prezentate mai devreme (*Servicii; Unelte și utilaje; Materii prime și materiale; Elemente, subansamble și stadii fizice; Rețele, echipamente și dotări interioare; Rețele, echipamente și dotări exterioare; Construcții integrale; Întreținere*

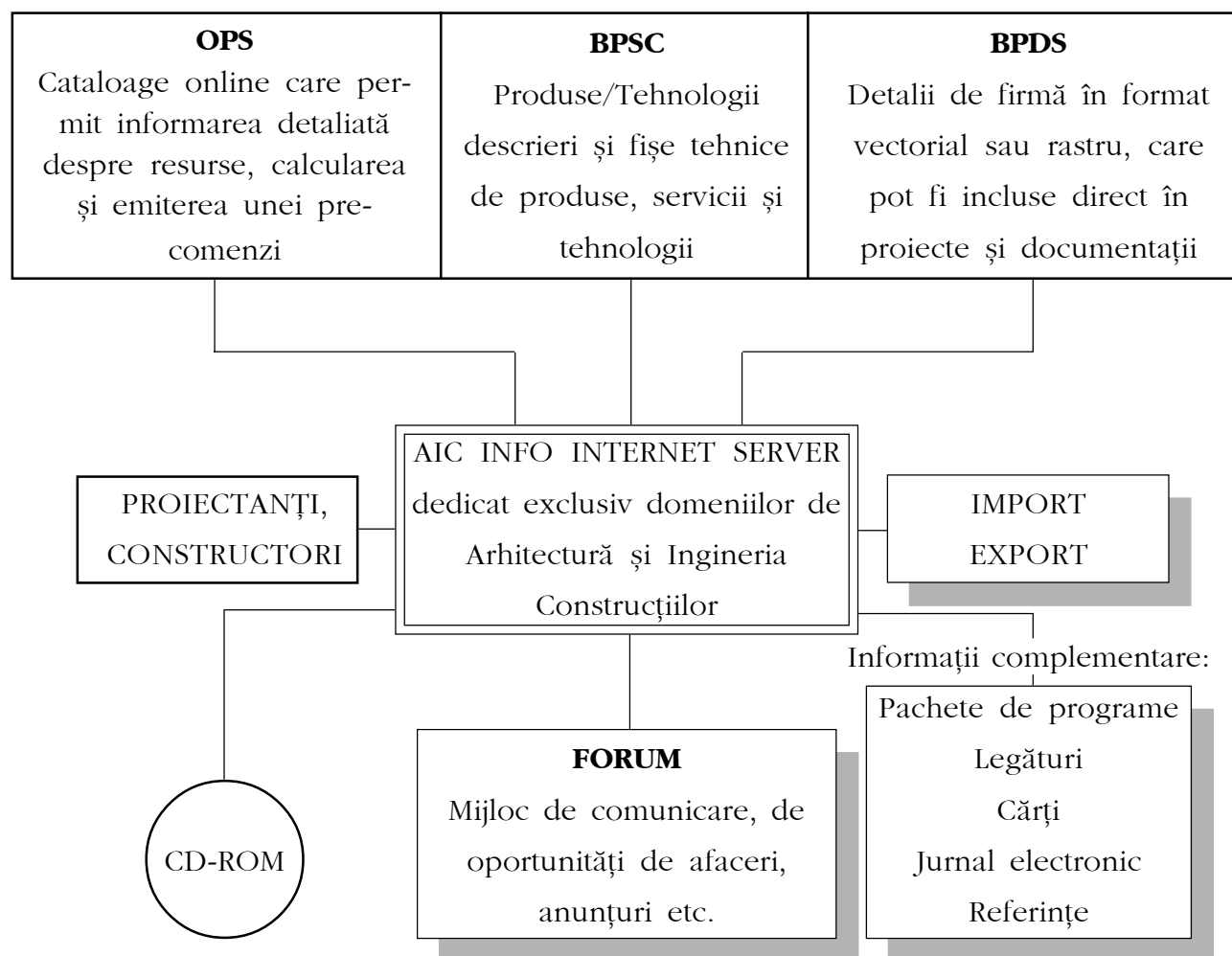


Fig.29 Componentele fundamentale ale sistemului AIC Info

și reparații; Demolare, recuperare, reciclare) și grupate sub capitolul *Produse/Tehnologii*. Există și un capitol special, referitor la protecția contra incendiilor. Aceasta este cea mai bogată parte din baza de cunoștințe. Accesarea informațiilor dorite se poate face atât prin "paginile dedicate" ale furnizorilor abonați la baza de cunoștințe, cât și prin căutare după cuvinte cheie sau prin legăturile inserate în diverse pagini. Am optat pentru această soluție fiind nevoiți să ținem cont de contextul (tehnic, comercial etc.) în care doreau cei în cauză să le fie prezentate resursele. Aceste pagini pot fi întreținute și de către furnizorul resurselor, nu numai de către noi. Prin anumite legături se pot accesa și alte situri cu sisteme internaționale de cataloage sau prezentări care merită menționate (de exemplu catalogul on-line Bosch sau situl concernului Deceuninck etc.).

Inițial sistemul a avut și o componentă conținând lista firmelor active din țară în domeniul construcțiilor. Concepută ca un director, era destinată să ofere referințe (nume, adrese, obiecte de activitate, persoane de contact etc.) în ordine alfabetică sau în funcție de anumite criterii alese. Ulterior însă am renunțat la această colecție, devenită oarecum redundantă și ineficientă din perspectiva noastră și ne-am rezumat la menținerea unei colecții de legături către firmele cu "pagini proprii" pe alte situri web, care doreau să fie incluse în baza de cunoștințe dezvoltată de către noi. Aceste legături pot fi accesate prin capitolul *Proiectanți, constructori*. Ce-i drept, trebuie să recunosc că menținerea la zi a unui director de firme (concurând și cu serviciile oferite de către alte firme, ca *Pagini aurii* sau alții) ne-ar fi depășit posibilitățile și ar fi avut repercursiuni asupra dezvoltării sistemului. Formula actuală oferă însă eficiență mai mare atât din punctul de vedere al întreținerii cât și în ceea ce privește obținerea informațiilor (siturile aflate în evidența noastră sunt verificate periodic, iar împrăștierea lor nu ne împovărează; tot periodic lansăm căutări de noi referințe pentru îmbogățirea conținutului). Ea ne ajută și la obținerea unor evidențe statistice referitoare la impactul modului de prezentare abordat de către diverse firme prin monitorizarea traficului.

• **Biblioteca de Proiecte și Detalii Standard** (BPDS) este menită să completeze BPSC. Deși conținutul ei s-ar putea încadra, într-o mare măsură, în capitolul *Produse/Tehnologii* (subcapitolul *Servicii*), am preferat dezvoltarea unei componente separate din cauza posibilităților de dezvoltare în continuare. Accesul la resursele cuprinse în această parte se poate face direct (prin *Proiecte, detalii*) sau din BPSC, prin anumite pagini care conțin referințe către proiecte realizate sau detalii oferite. Ideea realizării unei asemenea biblioteci era în primul rând performanța proiectării prin eliminarea timpului consumat la redesenarea unor detalii uniformizate și prin reducerea șanselor unor erori de reprezentare. Cele mai multe detalii se pot introduce direct în documentațiile de specialitate, în unele cazuri necesitând mici ajustări. Imaginile din proiectele prezentate pot fi vizualizate ca și detaliile, utilizând *plug-in-ul Whip* realizat de către *Autodesk Incorporated* (licența de distribuție și folosire fiind gratuite în prezent). Acest instrument permite efectuarea unor operații esențiale de vizualizare on-line (*zoom, pan*) cu un timp de răspuns satisfăcător cât și alte operații esențiale (*print, save as* etc.). Am optat pentru acest instrument deoarece (în ciuda gratuității) s-a dovedit eficient atât în cazul imaginilor vectoriale (DXF, DWG) cât și în cazul imaginilor rastru (JPEG, BMP, TIFF), exportate toate și încărcate sub formatul

comprimat propriu: DWF (*Drawing Web Format*). Acest format prezintă mai multe avantaje: pe lângă dimensiunea accesibilă a fișierelor grafice, permite și includerea informațiilor legate de drepturi de autor, realizare, dimensiune în desenul comprimat și vizualizarea acestora în momentul consultării.

Cu toate că acest subsistem nu este dezvoltat încă suficient, sunt convins că în viitor va reprezenta o resursă foarte utilă. Ca dovadă, că există interes pentru o asemenea componentă, pot aminti anunțurile și încercările pentru realizarea unor asemenea biblioteci în diferite domenii ale mecanicii (publicate și în revistele *CADreport* și *HelloCADFans*), precum și catalogul electronic realizat de către noi pentru reprezentanțele din România și Rusia ale concernului Deceuninck, pe care îl voi prezenta mai târziu. Tot prin acest subsistem am încercat să ne aducem aportul la popularizarea și sponsorizarea unui proiect umanitar destinat copiilor orfani din România (mini-campusul de lângă Gilău propus de către *Asociația Caritativă Cinderella*).

- **Online Pre-ordering System** (OPS) este tot o componentă oarecum întrețesută cu BPSC. Din câte știu, la ora actuală este încă tot singurul sistem orientat către comerțul electronic din domeniul construcțiilor. El se sprijină pe bazele de date care reflectă stocurile și prețurile la zi ale unor clienți abonați, precum și pe liste și tabele oferite de către câțiva furnizori de resurse. Informațiile oferite pot fi consultate fără constrângeri deși există și posibilitatea creării unor grupe și categorii de utilizatori cu diferite drepturi și restricții (după dorința firmelor abonate). Actualizarea și completarea acestor informații poate fi realizată și de către furnizorii lor (pe baza unei autentificări), atât sub formă pachetizată cât și on-line. Din cauza infrastructurii existente, până în prezent s-a optat la actualizarea pachetizată indirectă, apelând și la asistența oferită de noi. Aspectul cel mai important este însă acela că datele oferite sunt exacte, reflectând nu numai prețurile actuale (chiar dacă acestea sunt exprimate în valută), dar și cantitățile existente în stoc. Funcționarea acestui subsistem se poate descrie sumar, dar sugestiv, în felul următor:

Cumpărătorul interesat poate consulta lista de produse din categoria selectată de la firma dorită, alegând anumite produse în cantitatea dorită/disponibilă. Pe baza selecțiilor realizate își completează un coș virtual de cumpărături. Pe întreg parcursul procesului are acces la valorile totale și parțiale ale mărfurilor selectate, cât și la informații suplimentare legate de articolele prezentate în ofertă (există legături de la codurile de produs la elemente informative din BPSC). La închiderea coșului se realizează o pre-comandă care se transmite furnizorului și se face actualizarea virtuală a stocului on-line.

Actualizarea reală a stocului se produce doar în momentul efectuării reale a tranzacției. Operațiunile efective ce țin de tranzacția reală (facturare, achitare, livrare etc.) se realizează în afara sistemului nostru (prin contact direct, poștă etc.), deoarece infrastructura informatică și serviciile financiare nu sunt încă la un nivel corespunzător pentru a permite derularea completă a unor tranzacții via web în condiții de securitate și de performanță necesare.

Acest subsistem s-a bucurat de popularitate deja din faza experimentală, implementarea lui realizându-se chiar pe bază de contract cu Saniterm S.R.L. Oradea, la cererea expresă a acestei firme. Din ecourile apărute în urma unor manifestări (expoziția de construcții *Ambient'99* din Cluj-Napoca, conferința internațională de construcții *ÉPKO'99* ținută la Odorheiu Secuiesc) rezultă că există și alte firme interesate de asemenea facilități și chiar de acest subsistem prezentat. Deși în prezent OPS este mai degrabă un instrument informativ, cred că reprezintă un pas destul de important în transformarea mentalității actorilor din procesele comerciale din domeniul construcțiilor, demonstrând practic posibilitatea derulării în viitor a unor tranzacții într-o manieră nouă, mai performantă. În figura 30 este prezentată o captură de ecran, ilustrând și evocarea unei referințe sub formă grafică din BPSC în urma selectării codului de produs HX73_15 (înainte de completarea coșului virtual de cumpărături).

Index

Prezentarea generală a sistemului

Structura și construcția fittingurilor

Caracteristicile produsului

Coroziunea și efectele chimice

Dimensionarea sistemului și instrucțiuni de montare

▶ Calculatie de pret -- online!



SANITERM srl
str. Menumorut nr. 2
Bl. D105, et. 8, ap. 29
3700 Oradea
Tel./Fax: 059-143966
Tel. 093-244030,
093-264286

e-mail:
saniterm@medanet.ro

Afisare calculatie
Robinete

Cant	Denumire Produs	UM	Pret	Cod Produs
0	Robinet de inchidere incalzire 15 mm	buc	10.76	HX35_15
0	Robinet de inchidere incalzire 16 mm	buc	10.76	HX35_16
0	Robinet de inchidere incalzire 20 mm	buc	15.69	HX35_20
0	Robinet de inchidere incalzire 22 mm	buc	15.69	HX35_22
0	Robinet de inchidere apa 15 mm	buc	11.8	HX37_15
0	Robinet de inchidere apa 16 mm	buc	11.8	HX37_16
0	"Robinet de inchidere apa 3/4" - 15 mm	buc	10.49	HX38_15
0	"Robinet de inchidere apa 3/4" - 16 mm	buc	10.49	HX38_16
0	Supapa de golire din alama 16 mm	buc	4.17	HX32_16
0	"Ventil radiator cu cap termostatic 15x1/2"	buc	44.31	HX71_15
0	"Ventil radiator cu cap termostatic 16x1/2"	buc	44.31	HX71_16
0	"Ventil radiator manual 15x1/2"	buc	20.88	HX73_15
0	"Ventil radiator manual 16x1/2"	buc	20.88	HX73_16

Adauga la calculatie



Fig.30

Captură de ecran din OPS (cu afișarea referinței HX73_15)

- **Forum** este o componentă deosebită, concepută în primul rând pentru schimbul de informații. Acesta reprezintă un mijloc excelent de comunicare și ajutor reciproc via Internet oferind acces gratuit. Beneficiind de suport multilingual, permite publicarea informațiilor și anunțurilor în limbile: română, maghiară, engleză, germană, franceză și italiană. Acest subsistem constituia la început interfața externă a sistemului AIC Info, stând la baza dezvoltării resurselor generale (cu caracter internațional) din care face parte în prezent. Este organizat arborescent, la fiecare anunț putând atașa răspunsuri și referințe, iar prezentarea înregistrărilor se face în ordine invers cronologică, în ordinea actualității (prima dată se afișează înregistrările cele mai recente, informațiile cele mai vechi aflându-se la capătul paginii generate). Deoarece pagina se generează în maniera unitară descrisă la prezentarea sistemului (în urma interogării și restricționării/selecției aplicate asupra bazei de cunoștințe), legăturile și referințele afișate sunt întotdeauna corespunzătoare, în concordanță cu conținutul realizat de către participanți. În viitorul apropiat doresc să introduc și o legătură dinamică către partea de *Import/Export* (și vice versa), având în vedere natura celor mai recente informații înregistrate prin această componentă.

- Sub categoria **Informații complementare** se găsesc mai multe resurse importante (tot cu caracter internațional):

- ❶ Sub capitolul *Software* (Pachete de programe) se generează în prezent cinci liste (câte una pentru fiecare din categoriile ilustrate în captura de ecran din figura 31), cuprinzând prezentarea referințelor legate de pachetele de programe avute în evidență în baza de cunoștințe. Prima listă cuprinde referințele pachetelor destinate proiectării, proiectării asistate și modelării arhitecturale, desenării, realizării imaginilor fotorealiste și randate etc. În cea de a doua listă sunt prezentate referințele pachetelor destinate proiectării rețelelor, producției și ingineriei asistate de calculator etc. În lista a treia se prezintă colecția referitoare la programele de analiză structurală, iar în lista a patra sunt grupate referințele la programele destinate planificării și organizării în construcții. Ultima listă cuprinde o categorie mai aparte referindu-se la dezvoltări de programe

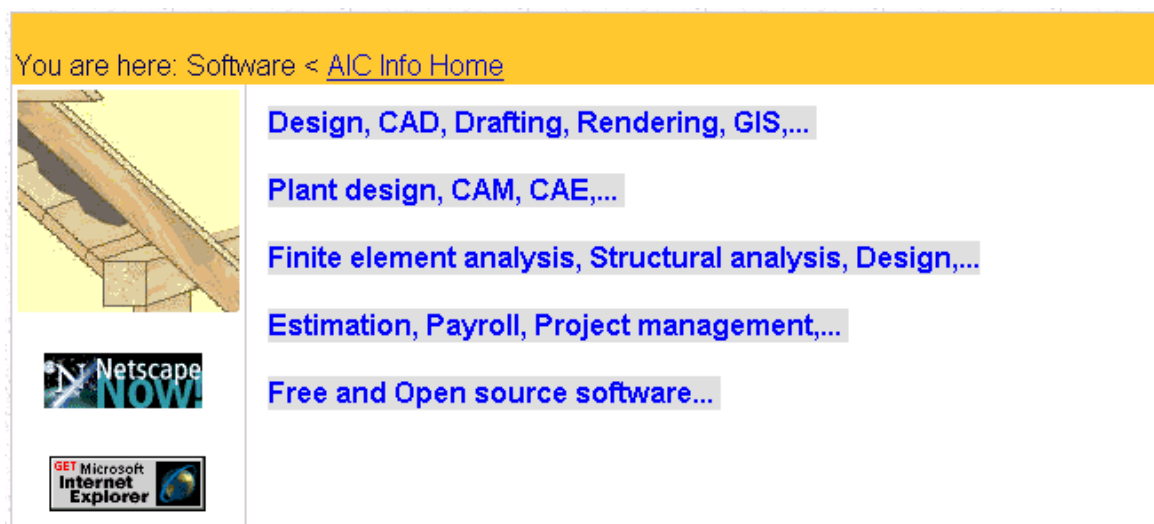


Fig.31 Cele cinci subcategorii din pagina *Software*

cu licențe de utilizare gratuite (*Freeware, General Public License, Open Source Software* etc.). Listele generate nu sunt atotcuprinzătoare. Am încercat evidențierea celor mai accesibile resurse, având în vedere facilitățile oferite, echipamentele de calcul necesare și prețul cât mai atractiv al licențelor.

- ❷ În pagina care este generată la accesarea capitolului *Links* (Legături) se prezintă o colecție alcătuită din descrierea sumară și adresele (sub formă de legături active) ale unor situri web legate de construcții, recomandate de către noi. Selectarea resurselor din acest domeniu este un proces relativ subiectiv, prin care încercăm să ținem cont atât de conținutul domeniilor cât și de funcționalitatea și estetica lor. Pentru a mări atractivitatea sitului nostru, în fiecare lună prezentăm pe pagina de intrare o legătură către un sit mai deosebit.
- ❸ La subcategoria Books (Cărți) se prezintă informațiile din sistem referitoare la câteva edituri de renume (John Wiley & Sons, Kluwer, Blackwel, E & FN Spon), precum și o listă cu câteva publicații interesante din domeniul arhitecturii, calculului construcțiilor etc. Prin această componentă se pot accesa editurile în cauză sau rezumatele cărților selectate, utilizatorii care dispun de cartele de credit putând chiar comanda publicațiile dorite (în măsura în care tranzacțiile electronice sunt posibile în zona utilizatorului). Din păcate până în prezent nu am găsit resurse românești pe Internet corespunzătoare acestui capitol.
- ❹ *E-Journal* (Jurnal electronic) este un capitol desprins din Forum și dezvoltat tot în scopul măririi atractivității sitului. El conține articole originale (în diverse limbi) și traduse, critici, prezentări etc. (sub formă de documente electronice). În cadrul documentelor publicate există *hyperlink*-uri către referințe (note, bibliografie etc.). Publicarea materialelor prin acest subsistem este gratuită (ca și lecturarea lor), însă în ceea ce privește forma și conținutul trebuie să corespundă unor criterii de evaluare. Deși copierea materialelor publicate nu poate fi împiedicată prin posibilitățile existente, reproducerea acestora nu este permisă fără acordul deținătorilor de proprietate intelectuală.
- ❺ *Referințele* sunt prezentate bilingv: în cadrul resurselor române cu titlul *Despre AIC Info/Contact*, iar în cadrul resurselor generale cu titlul *About/Contact*. Ele cuprind pagini statice ca și conținut, prin care se explică și se prezintă sistemul pe scurt.

- Distribuirea sistemului AIC Info pe **CD-ROM** era o activitate necesară la început, în primul rând din cauza condițiilor inexistente peste tot de acces la Internet. Din cauza acestui aspect, conținutul imprimat pe discuri optice era mai bogat la capitolul de *Software* decât imaginea de pe web, CD-ul conținând pe lângă ultimele distribuții ale browserelor grafice MS-IE Explorer și Netscape Communicator și diverse *plug-in*-uri, aplicații, *demo*-uri etc. la care obținusem licență de distribuție. În prezent asemenea discuri se pot obține contra cost, prin lansarea unei comenzi în cadrul capitolului corespunzător, specificând cel puțin datele marcate ca necesare în formularul afișat. Rolul acestora nu mai este suplینirea sistemului de pe Internet, ci mai degrabă completarea acestuia prin eliminarea problemelor legate de descărcarea unor pachete mai mari de programe.

Prin colecția destul de bogată de care dispune (atât ca formă cât și conținut), sistemul AIC Info are și un rol educațional important. El oferă nu numai cunoștințe din domenii destul de variate ale construcțiilor, dar și instrumente practice care rezolvă anumite probleme. Prin maniera de modelare și de prezentare a informațiilor cuprinse în baza de cunoștințe, se poate afirma că sistemul este deschis, permițând nu numai înglobarea altor cunoștințe, subsisteme (chiar cu modelări și abordări diferite), dar și transformarea la nevoie a subsistemelor implementate. Aspectul accesului gratuit și neîngrădit este iarăși foarte important, apreciat în mod deosebit de către studenți. Având informațiile structurate și o interfață simplă, utilizarea resurselor oferite este foarte eficientă, fără a presupune o pregătire specială. În ceea ce privește susținerea financiară, formula de abonament ce se aplică clienților care doresc să ofere diferite informații prin acest sistem, împreună cu o activitate de marketing și de promovare continuă, par să asigure dezvoltarea în continuare a sistemului cel puțin sub aspectul conținutului.

4.2.3 Aplicații decizionale implementate prin AIC Info

După cum am menționat, structura sistemului AIC Info se poate privi și din punctul de vedere al structurii de stratificare. În acest context, peste primul strat (care conține cunoștințele propriu-zise) există un al doilea strat cu reguli, proceduri etc. Materializarea acestor componente este și ea cel puțin atât de variată ca și formele de reprezentare ale cunoștințelor peste care se extind. Cele mai multe dintre acestea sunt instrumente dezvoltate cu un scop precis. Din mulțimea acestora aș dori să prezint două programe care se detașează oarecum de celelalte. Ambele au fost realizate cu titlu experimental, reprezentând de fapt două subsisteme decizionale diferite. Prezentarea lor permite să încerc și realizarea unor comparații, atât la nivelul evolutiv al sistemului pe ansamblu cât și la nivelul conceptelor și tehnologiilor abordate.

Primul a fost realizat anul trecut, urmărind integrarea unor informații tehnice specifice, economico-financiare și normative, folosind în acest scop o parte din documentația unui producător renumit de panouri din gips-carton, referitoare la produse și tehnologii conexe. Având în vedere nivelul de dezvoltare (din perioada realizării) al instrumentelor disponibile, am optat pentru o soluție mixtă: soluție locală și client-server. Această opțiune era motivată în primul rând de cele două modalități de acces (Internet și CD-ROM) promovate. Ținând cont însă de schimbările destul de rapide suferite de către normative, cele două variante trebuiau să funcționeze împreună, conducând astfel la soluția mixtă. Pe de altă parte, am dorit să deschid apetitul utilizatorului pentru completarea consultării, utilizând informații actualizate care pot fi găsite prin Internet. Deși acest prim sistem s-a bucurat de un real succes, datorită unor probleme legate de stabilirea exactă a prețurilor de vânzare și din cauza unor aspecte legate de concurență, Rigips România S.R.L. (beneficiarul subsistemului decizional) ne-a solicitat retragerea pe moment a acestei aplicații din cadrul serviciilor gratuite oferite prin sistemul AIC Info.

A doua aplicație s-a născut din dorința integrării informațiilor tehnice în cadrul unui mini-sistem expert pentru selecția de vopsele, lacuri și produse conexe. Această aplicație este

mai recentă (în prezent aflându-se în faza de prototip în curs de implementare), deci atât cunoștințele mele cât și instrumentele disponibile realizării au beneficiat de o anumită evoluare, permițând o dezvoltare mai performantă. Și utilizarea resurselor oferite de Internet a devenit mai extinsă față de perioada realizării primei aplicații. În aceste condiții, ținând cont și de previzibilitatea posibilelor schimbări în conținutul cunoștințelor modelate, am optat pentru două soluții complet separate: una client-server și una locală. Ceea ce am încercat de fapt cu această ocazie a fost studiul posibilității de generare automată a unui program local pornind de la un program server, mai precis realizarea unei translații din Perl în JavaScript și înlocuirea tabelelor din baza de date server prin intermediul unor proceduri locale. Deși este evident că fiabilitatea și calitatea unui sistem expert depinde în foarte mare măsură de actualitatea bazei de cunoștințe pe care este construit, posibilitatea distribuirii de programe locale prezintă o atractivitate considerabilă pentru perioada actuală.

Din punctul de vedere al utilizatorului, natura asistenței este diferită la cele două cazuri prezentate. La prima aplicație (referitoare la panourile de gips-carton) am plecat de la situația proiectantului care dorește să specifice o soluție având ca principali parametri funcționalitatea, costul și interacțiunea cu structura. Ca răspuns se generează mai multe

Calculador de costuri

Cantitate: 0 mp 1 ATS = 0 lei

RIGIPS ROMANIA S.R.L.
 Tel: +40 1 3305614, +40 1 33057-23(24), Fax: +40 1 33056-08(11)
 Banca Societate Generale cont 30293

FISA DE CALCUL
 (consumuri normale)

Pereti despartitori CW 50/75 - simpli, cu structura metalica
 Cod:
 Cantitate: 30 mp;
 Rata de schimb: 960 lei/ATS

Cost total fara TVA:
 39003 lei/mp
 40.62 ATS/mp

Elementul de montaj	Unitate de masura / Unitate de livrare	Cantitate	Consum specific [UM / mp]	Consum normal	Cantitate necesara [unitati de livrare]	Cantitate livrabila [unitati de livrare]	Pret unitar (ATS)	Valoarea (lei)
Placa RB 12.5 mm	m/buc	2.75	2	60	21.81	22	26.84	5668
Profil UW 50	m/buc	4	.8	24	6	6	6.844	3942
Profil CW 50	m/buc	2.6	1.7	51	19.61	20	8.342	1601

Instructiuni de utilizare

Acest sistem a fost proiectat pentru Netscape Navigator 4.0x si MS Internet Explorer 4.0x

1. In cadrul superior, completati campurile Cantitate si Rata de schimb de referinta (ATS = schilling austriac)
2. Selectati printr-un clic din lista rulanta Foia de calcul

Fig. 32

Capturi de ecran din subsistemul decizional realizat pentru RIGIPS România S.R.L.

documente virtuale, afișate sub forma de cadre aranjate. Din același ecran, utilizatorul are acces la informații tehnice detaliate, precum și la date tehnice și financiare pentru soluții de realizare (rețete) prestabilite. Pentru fiecare soluție aleasă se calculează costul, prin introducerea cantității și a unei parități valutare. Decizia pe care o va lua proiectantul nu este una sugerată de sistem. Programul are rolul de a asigura un acces comod și eficient la informațiile sistematizate și structurate, altfel spus de a asigura un *maximum* de informație optim accesibilă necesară sprijinirii procesului decizional.

În ceea ce privește a doua aplicație (destinată alegerii vopselelor, lacurilor și substanțelor conexe), am considerat situația utilizatorului produsului finit, fie acesta un executant sau o persoană care dorește să execute lucrări la domiciliu. Din studiul documentației disponibile am dedus un set de parametri de selecție pe care i-am integrat în baza de date, în așa fel încât numărul variantelor din care trebuie selectat produsul sau rețeta (combinația) corespunzătoare să fie *minim*. În acest caz, majoritatea regulilor aferente sistemului expert sunt implicite, ele fiind incluse în parametrii de selecție. Utilizatorul fixează parametrii problemei sale, după care obține o listă filtrată de produse sau rețete disponibile. Prin legăturile atașate, acesta poate obține informații suplimentare legate în special de tehnologia de aplicare. Soluția este modulară, o procedură de calcul de preț fiind ușor de atașat.

4.2.4 Realizări conexe (prin revalorificarea parțială a modelării implementate)

Datorită dezvoltării și promovării sistemului AIC Info, am intrat în contact cu foarte mulți actori din domeniul construcțiilor, prezenți în țara noastră. Firma KLAUSEN S.R.L. din Cluj-Napoca (dealer pentru sisteme de prindere și ancorare marca FISCHER și sisteme de siguranță și închidere marca ELZET) a fost printre primele care ne-au solicitat realizarea unor cataloage electronice proprii în vederea distribuirii cât mai eficiente a informațiilor legate de produsele comercializate. Așa am început, ca pe lângă CD-ROM-ul AIC Info, să realizăm și alte sisteme informative pe diferite suporturi. Sistemul realizat pentru firma KLAUSEN a fost înregistrat inițial pe dischete. Mai apoi prin lărgirea volumului de informații și includerea unor programe de calcul am realizat o nouă imagine multiplicată pe discuri optice. Această variantă conținea și elemente din baza de cunoștințe destinată construcțiilor, fiind foarte asemănătoare cu CD-ROM-ul AIC Info.

După realizarea acestui "minisistem" au apărut și alte firme interesate (chiar cu cerințe mai de anvergură). Una din cele mai importante este reprezentanța pentru România a concernului Deceuninck Industries N.V. cu sediul central în Belgia. La început au optat pentru realizarea unor materiale de prezentare în cadrul sistemului AIC Info, dar ulterior conducerea belgiană a concernului a decis să nu se publice prin Internet informații traduse în alte limbi legate de activitatea firmei, pentru a păstra o imagine unică (existând deja un sit dedicat al firmei). În schimb, catalogul tipărit care conține toată gama de produse și toate detaliile necesare proiectării și execuției, trebuia să fie distribuit în cercurile de specialitate din diverse țări. Apreciind rezultatele noastre, obținute prin realizarea bazei de cunoștințe anterior prezentate, au acceptat oferta noastră de a transpune catalogul

lor de produse în format electronic, tradus în limba română, în vederea unei distribuții și utilizări cât mai performante. După prezentarea prototipului realizat s-a optat pentru extinderea acestuia și cu o parte de prezentare generală cu conținut multimedia. În urma succesului acestui catalog, am primit însărcinarea de a realiza același produs și pentru reprezentanța Deceuninck din Rusia, iar în prezent particip și la realizarea CD-ului destinat pieței britanice.

La realizarea acestui sistem am apelat la maniera de modelare utilizată și în cadrul sistemului AIC Info, având în vedere că și în acest caz era vorba de informații din domeniul construcțiilor cu conținut și format destul de variat. Sistemul rezultat este tot un sistem deschis, orientat pe obiecte, deși se adresează unui cerc mai restrâns de persoane și este destinat calculatoarelor cu sistem de operare Windows pe 32 de biți (95, 98, NT sau 2000). Această destinație legată de platforma tehnică nu este foarte strânsă, catalogul realizat putând fi folosit și sub alte sisteme de operare, deci implicit și pe alte platforme de calcul. Exploatarea performantă a resurselor oferite presupune însă existența unui browser grafic (minimum versiunea 3.02) și a unor plug-in-uri ca Whip (versiunea 3.1 sau mai recent, necesar pentru prezentarea detaliilor grafice de specialitate incluse în catalog), QuickTime (versiunea 2.1 cel puțin, pentru a savura prezentarea multimedia inclusă) etc. Din câte știu, până la această oră nu există versiuni ale Whip-ului pentru sisteme de operare diferite de Windows. Din acest motiv, precum și din cauza rezultatelor testărilor efectuate de către mine (care s-au rezumat doar la variantele pe 32 de biți ale sistemului de operare Windows și la Mac-OS 7.8...8.0), există această "limitare".

Home | BPSC | Software | Forum | Links | e-Libris

fischer 

Ancoraje pentru sarcini ridicate

KLAUSEN srl
str. Dornei nr. 14,
3400 Cluj-Napoca
Tel/fax: 064-432837, 064-134301
Tel. mobil: 092 319 422

FZA - Ancoraje cu bolt
FZEA - Ancoraj cu introducere prin lovire
FAN - Ancoraj cu bolt
FHA - Ancoraje de putere mare
FB - Bolt special

-- Go! --

1. **FZA - Ancoraj cu bolt**
2. **FZA-D - Ancoraj cu introducere prin material**
3. **FZA-I - Ancoraj cu filet interior**

Material: Otel galvanizat, zincat si otel inoxidabil A4.

Nou: La cerere si din otel anticorrosiv 1.4529 – cu aprobare.

DOMENIU DE UTILIZARE

Materiale de constructii: Beton obisnuit > Bc15, piatra naturala cu structura densa.

Pentru fixarea de: constructii de otel in general, balustrade, sine de ancorare, masini, console, porti, trepte, scari, fatade, etc.

MONTARE

Montarea a fost corecta in cazul in care tuburile de ancoraj ale modelelor FZA si FZA-I sunt cufundate cu 1 mm in masa betonului. Pentru o si mai mare siguranta de montare, la modelele FZA si FZA-D a fost introdus un inel de control verde, care este vizibil doar in cazul unei montari corecte. In acest fel greselile de montare sunt excluse.




Fig.33

Captură de ecran din "minisistemul" realizat pentru firma KLAUSEN, cu legături la sistemul de bază de cunoștințe AIC Info.

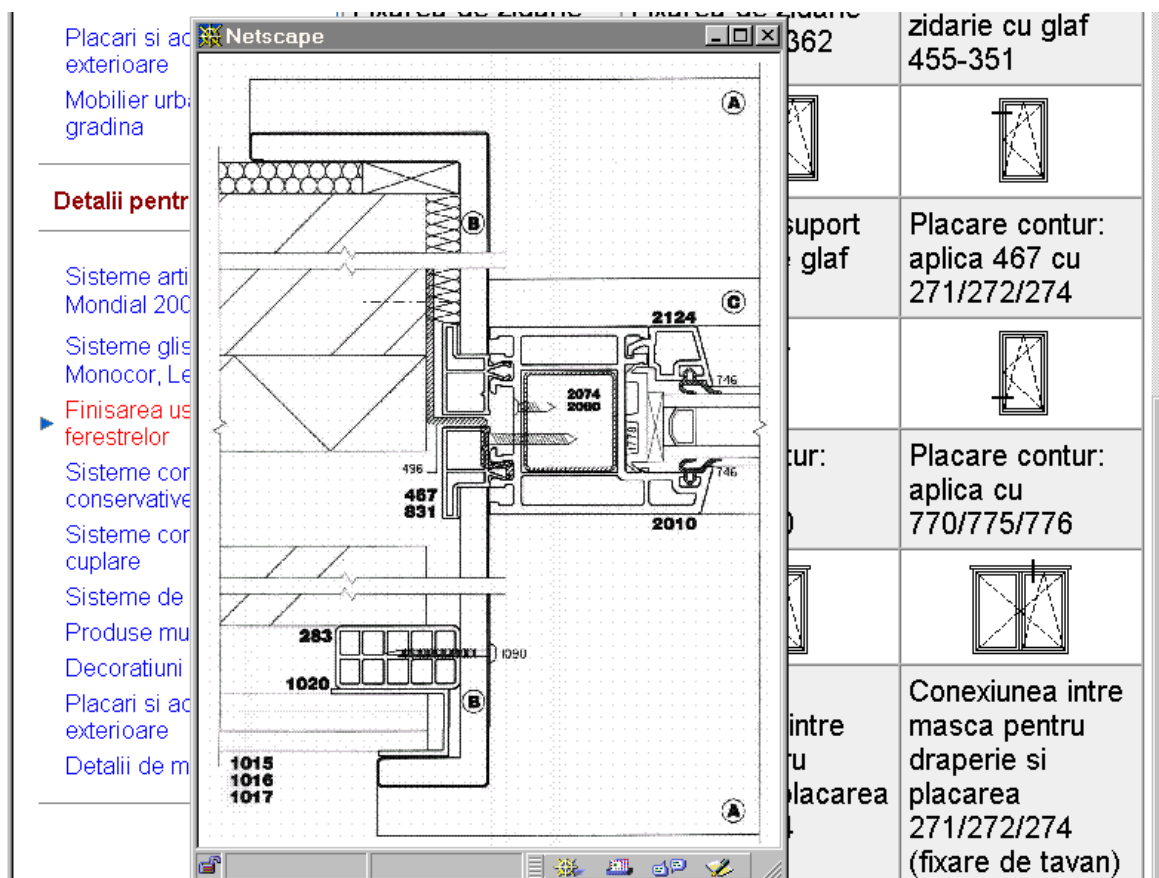


Fig.34

Prima pagină
 din catalogul
 electronic
 realizat pentru
 Deceuninck
 România

Placari si acoperiri exterioare Mobilier urban si de gradina Detalii pentru arhitecți Sisteme articulate: Mondial 2000/Standard Sisteme glisante: Monocor, Levacor ► Finisarea usilor si ferestrelor Sisteme constructive conservative Sisteme constructive de cuplare Sisteme de protecție Produse multi-funcționale Decoratiuni interioare Placari si acoperiri exterioare Detalii de montaj	Fixarea de zidarie cu glaf 350-351	Fixarea de zidarie cu glaf 453-362	zidarie cu glaf 455-351
	Fixarea de zidarie cu glaf 385-347	Profil 1735 suport pentru fixare glaf din aluminiu	Placare contur: aplica 467 cu 271/272/274
	Placare contur: aplica 831 cu 271/272/274	Placare contur: aplica cu 426/429/430	Placare contur: aplica cu 770/775/776
	Placare contur: aplica 467/831 cu Deeplas	Conexiunea intre masca pentru draperie si placarea 271/272/274	Conexiunea intre masca pentru draperie si placarea 271/272/274 (fixare de tavan)

Fig.35 Prin selectarea din meniul de navigare se generează conținutul corespunzător al paginii. Accesarea detaliilor din catalog se face prin selectarea acestora din tabelul generat, ele fiind prezentate într-o nouă fereastră.



Informațiile au fost structurate pe mai multe niveluri, accesul la ele făcându-se prin intermediul unor pagini preformate în funcție de context. Aceste pagini realizate în HTML se compun dintr-o parte statică superioară cu coordonatele firmei și două părți generate dinamic în funcție de contextul de prezentare: partea din stânga conține un meniu pentru navigare, iar în dreapta acestei zone se afișează conținutul propriu-zis al părții selectate (figura 34). Meniul de navigare este structurat în trei părți: informații generale (*Istoric si profil; De la materia primă la profilul din PVC-U; Certificare ISO; Câteva caracteristici tehnice; PVC-U -- omul și mediul*), informații referitoare la prezentarea sistemelor oferite (*Sisteme de tâmplării; Sisteme pentru construcții; Închideri; Produse multi-funcționale; Decorațiuni interioare; Placări și acoperiri exterioare; Mobilier urban și de grădină*) și catalogul propriu-zis (sub capitolul *Detalii pentru arhitecți*). Pentru a ușura navigarea, meniul este prezent pe fiecare pagină generată, marcând totodată și capitolul accesat. În cazul accesării detaliilor (figura 35), conținutul generat în dreapta meniului de navigare devine un submeniu tabelar contextual (conținând iconițe sugestive și descrieri succinte). Prezentarea detaliilor grafice selectate din acest submeniu se face printr-o fereastră dinamică nouă, pentru a ușura manevrarea lor, profitând și de facilitățile oferite de Whip.

Detaliile grafice de specialitate sunt prezentate sub formatul DWF, deci pot fi accesate cu ușurință și prin rețea, iar pentru includerea lor în diverse documente se pot salva sub formă rastru (salvarea lor vectorială presupunând descărcarea variantelor DXF sau DWG alăturate DWF-urilor). În comparație cu catalogul inițial tipărit, sistemul realizat prezintă mai multe avantaje evidente:

- Volum fizic mult mai redus față de catalogul inițial, transportul, stocarea și distribuirea mult mai facile, datorită și greutateii neglijabile a produsului.
- Posibilitatea combinării unor informații foarte diferite ca formă: text, imagini fotografice, desene, secvențe video, sunet înregistrat, adrese și legături (adrese de poștă electronică, URL-uri) etc.
- Posibilitatea înglobării în sistem a unor proceduri sau aplicații.
- Accesul performant la resursele oferite, prin legăturile și meniurile realizate respectiv prin structurarea informațiilor variate oferite.

Având în vedere dorința limitării resurselor soft și hard necesitate, precum și volumul relativ restrâns al cunoștințelor înmagazinate, am preferat realizarea unui sistem fără a apela la baze de date. Regăsirea informațiilor dorite se realizează astfel prin tabele cu adrese concepute static, accesate în momentul generării paginilor prin interfața realizată. Acest aspect însă nu exclude posibilitatea unor transformări ulterioare sau extinderea sistemului cu alte facilități dorite. Având în vedere structura clară și interfața simplă dar eficientă a sistemului, nu s-a considerat necesară dezvoltarea și includerea unor componente suplimentare cu facilități de explicare, ghidare sau ajutor.

CAP.5 CONCLUZII

5.1 DOCUMENTAREA ȘI SINTEZA REALIZATĂ, SISTEMUL EXPERT BPC

Documentarea extinsă realizată de-a lungul anilor de pregătire dedicate prezentei teze m-a ajutat să pătrund în lumea fascinantă a cercetărilor legate de una din ramurile inteligenței artificiale cu aplicabilitate în domeniul construcțiilor. Dacă la început eram interesat în cunoașterea, stăpânirea și aplicarea unor asemenea instrumente mai degrabă în calculul construcțiilor, pe parcursul cercetărilor întreprinse mi-am dat seama de potențialul extraordinar oferit de cuplarea teoriilor din acest domeniu cu instrumentele actuale din tehnologia informatică. Modalitățile practice existente de calcul structural sunt instrumente bazate în general pe teorii cunoscute, de cele mai multe ori consacrate, înnoirea lor constând de fapt din înglobarea unor interfețe grafice deosebite și extinderea calității acestor pachete în funcție de resursele din ce în ce mai extinse ale echipamentelor de calcul utilizate. În asemenea condiții, aplicarea sistemelor expert poate avea un rol foarte important în problemele cu natură neclară, nedeterminată (folosind și mulțimile vagi), la analize și expertizări extinse sau la aplicații cu caracter educațional, aliniindu-ne însă la condițiile descrise mai sus. Dacă la începutul anilor '80 dezvoltarea unor sisteme expert necesita resurse umane și financiare importante, pe durate de timp considerabile, azi datorită dezvoltării "firești" din domeniul tehnologiei informațiilor, realizarea acestora este mult mai facilă, existând chiar instrumente dedicate pentru așa ceva.

Strângerea informațiilor necesare pregătirii acestei teze, a publicațiilor și a realizărilor conexe exemplificatoare, au necesitat un efort susținut mai ales în primii ani dedicați pregătirii mele. Procurarea unor asemenea produse soft și familiarizarea mea cu ele a necesitat un timp semnificativ, dedicat unor preocupări fără rezultate aparente imediate. Dintre shell-urile comerciale testate și utilizate aș aminti doar câteva: VP-Expert, MicroPS, M1, Expert System Toolkit (TurboExpert), EXSYS.

Sper ca sinteza prezentată în această teză, referitoare atât la instrumentele existente dedicate realizării sistemelor expert cât și la conceptele realizării și implementării unor sisteme în domeniul construcțiilor, să fie de folos celor care doresc să se pregătească în această direcție. Nu puteam prezenta doar stadiul curent al acestor preocupări fără a face referiri la realizările anterioare, mai ales în ipostaza în care cele mai multe realizări actuale se bazează pe teorii și instrumente apărute în anii '80. Domeniul sistemelor expert și al bazelor de cunoștințe este foarte extins, îmbogățindu-se cu trecerea timpului, deci deși am încercat realizarea unei priviri cât mai cuprinzătoare (cel puțin pentru domeniul construcțiilor) nu pot susține că am reușit în totalitate. Pentru a compensa într-un fel eventualele lipsuri, am inclus și cele mai actuale referințe publice accesibile prin Internet (în ANEXA 4).

Poate constui o surpriză lipsa formulelor din lucrare și lipsa surselor complete ale programelor scrise, dar am considerat mai necesară abordarea și prezentarea în prim plan

ale unor concepte în locul conținutului poate sec și monoton al codificărilor (exemplificate totuși prin conținutul relativ bogat al anexelor). Prin realizarea celor 3 variante ale sistemului expert BPC (destinat documentării în cadrul Bibliotecii de programe destinate construcțiilor, editată de IPCT București) am avut o ocazie deosebită de a aborda practic același domeniu cu concepte și instrumente diferite. Dintre contribuțiile și realizările personale la acest capitol aș aminti:

- analiza detaliată prin diagrame cu fluxuri de date și prin întocmirea dicționarelor de date aferente a domeniului abordat, în prima etapă printr-o descompunere de sus în jos (de la general la particular) a situației existente, iar apoi prin recompunerea de jos în sus a modelului noului sistem imaginat;
- organizarea programelor din domeniul construcțiilor în clase și subclase, conceperea unui model fizic viabil și eficient pentru acesta;
- realizarea prototipului sistemului expert BPC (în 3 versiuni succesive) și implementarea acestuia (în medii de lucru diferite: de pe minicalculatoare la PC-uri), ceea ce a presupus conceperea și dezvoltarea unor baze de cunoștințe, motoare de inferență, interfețe utilizator precum și alte instrumente specifice (analizorul semantic, generatorul de răspuns, pachetul destinat generării și administrării sistemului realizat etc);
- valorificarea rezultatelor atât prin lucrări prezentate la sesiuni științifice cât și prin contracte de cercetare.

Deși aceste prime experimentări și rezultate obținute mă purtau în direcția dezvoltării sistemului BPC prezentat și a altor sisteme expert similare, o avarie nefericită la echipamentul meu de calcul (cedarea discului într-un moment sensibil) m-a pus în situația de a lua în considerare aspecte mai practice legate de piața echipamentelor de calcul din țară. Renunțând oarecum la caracterul avantgardist din punctul de vedere al funcționării aplicațiilor dorite, am decis să insist în realizarea unui sistem prin care să pot contribui într-o manieră practică și eficientă și la dezvoltarea construcțiilor din țara noastră. Din păcate, impresia mea subiectivă referitoare la incapacitate susținerii bugetare corespunzătoare a cercetărilor a rămas neschimbată până azi. Astfel, am renunțat la perfecționarea și dezvoltarea sistemului BPC (Biblioteca Programelor pentru Construcții) în cadrul universității, rezultatele obținute fiind totuși valorificate prin două părți de contracte cu Ministerul Învățământului și Științei (în 1991 [78] și 1992 [79]).

5.2 CONTRIBUȚII LA PERFECTIONAREA PROIECTĂRII, INSTRUMENTE "INTELIGENTE"

Având în vedere resursele disponibile limitate, am încercat asigurarea susținerii materiale a cercetărilor dorite pe căi personale. Astfel, după o întrerupere de șapte luni a pregătirii doctoratului prin participarea la lucrări de execuție în Germania (în urma unei detașări cu durată determinată la firma COMECO), am reușit achiziționarea unui calculator personal și a unui volum important de documentații. Pentru a putea continua perfecționarea în domeniul vizat am încercat împletirea cercetării cu organizarea și desfășurarea activității de proiectare în cadrul firmei CONSORT Construcții. Din activitatea mea din această

perioadă aş dori să scot în evidență:

- organizarea și structurarea activităților legate de proiectare în cadrul firmei amintite;
- realizarea și întreținerea legăturii între diverse echipamente și componente soft;
- realizarea și dezvoltarea unor elemente parametrizate de bibliotecă pentru proiectare arhitecturală asistată de calculator (de exemplu: cadre metalice spațiale modulare care furnizează cotele de montaj rezultate în urma poziționării, chiuvete de colț cu unghi adaptiv etc.);
- conceperea unor instrumente software pentru dimensionare și desenare (inclusiv a unui algoritm de alegere automată a diametrelor barelor de armătură) și participarea la codificarea, implementarea și dezvoltarea lor;
- conceperea unui prototip dedicat modelării virtuale a halelor standard LINDAB.

Atât obiectele de bibliotecă parametrizate cât și instrumentele dezvoltate au fost utilizate în mod curent în activitatea de proiectare din cadrul firmei CONSORT Construcții, și valorificate prin proiectele realizate. Din cauza evoluțiilor economice și sociale, în momentul în care activitatea de proiectare amenința cu acapararea timpului meu disponibil în dauna pregătirii dorite (având în vedere și relativa stagnare în dezvoltarea echipamentelor la firma menționată), am fost nevoit să pășesc într-o altă direcție. Astfel, în 1998 am renunțat și la dezvoltarea instrumentelor practice de proiectare și desenare realizate prin concursul colegilor entuziaști de la firma menționată. Trebuie să recunosc că aceste renunțări au fost oarecum forțate, fiind rezultatul deciziilor mele în ceea ce priveau perspectivele conjuncturale de pregătire și de realizare profesională, referindu-se doar la participarea mea efectivă și nemijlocită în dezvoltarea sistemelor amintite și nu la întreruperea totală a interesului sau a sprijinului acordat lor.

5.3 BAZA DE CUNOȘTINȚE AIC INFO ȘI REALIZĂRI CONEXE

Participând la foarte multe manifestări legate de construcții (expoziții, conferințe etc.) am fost oarecum surprins de ineficiența comunicării și de natura de cele mai multe ori neperformantă ce caracteriza schimbul de informații. Practic, mi s-a întâmplat să fiu căutat și contactat de mai multe ori cu probleme referitoare la subiecte tratate sau prezentate cu un an (sau chiar mai mult) înainte. Am avut și eu ocazia să caut informații diferite și să nu știu cum să găsesc surse competente. De foarte multe ori mi se adresau persoane diferite cu aceleași solicitări și întrebări:

- Ce echipamente sunt necesare pentru proiectare asistată?
- Care sunt programele cele mai performante, ce pregătire necesită folosirea lor?
- Care sunt pachetele de proiectare cu prețul cel mai accesibil?
- Cine este furnizorul materialului folosit în proiectul X?
- etc.

În acest context mi s-a părut mult mai interesantă abordarea sistemelor bazate pe

cunoștințe dintr-un alt punct de vedere. Având în vedere multitudinea formelor sub care se pot prezenta cunoștințele referitoare la construcții, modelarea acestora pentru a fi incluse în cadrul unui sistem performant și gestionarea lor performantă reprezenta o provocare incitantă. Indiferent dacă este vorba de schițe, imagini, descrieri textuale sau mai formale, ele trebuie să-și îndeplinească menirea, adică să ajungă la cei vizați și să fie comprehensibile de o manieră cât mai utilă. Cunoscând echipamentele limitate, de multe ori depășite, existente în instituțiile de învățământ superior de construcții, situația pieței de calculatoare din țară (într-o oarecare contradicție cu evoluția pieței din domeniul comunicațiilor), am preferat în final orientarea către realizarea unor instrumente care să contribuie în mod pozitiv, în primul rând la diseminarea cunoștințelor într-o manieră cât mai performantă în domeniul construcțiilor.

După ani de cercetări și experimentări a început să se cristalizeze posibilitatea realizării unui proiect de anvergură, care pe lângă necesitatea aplicării cunoștințelor acumulate să asigure și materializarea intențiilor într-o formă modernă și performantă, ținând totodată cont și de cerințele unei piețe în formare. Pentru a putea dezvolta un sistem într-adevăr practic și util, a trebuit să recurg la o formulă comercială. Prin acesta se putea asigura nu numai suportul material și financiar absolut necesar, dar și eco-ul real al impactului și utilității sistemului. În aceeași timp, fiind conștient că un asemenea proiect are nevoie de mai mult decât resursele unei persoane, trebuia să-mi găsesc colaboratori interesați și pricepuți. Succesul din acest punct de vedere pot să spun că a fost atins în momentul în care am reușit să mă înțeleg cu colegul ing. Dan Turda, inițial de a participa la strângerea și transpunerea în format electronic al unor informații, în scopul creării unei baze de cunoștințe autohtone dedicate construcțiilor. Deși discuțiile inițiale și punerea teoretică la punct a unor probleme a necesitat câteva luni, o doză apreciabilă de curaj și un oarecare spirit de sacrificiu, am reușit să pornim un tăvălug care promitea schimbarea (de altfel inevitabilă) mentalității participanților. Însăși generarea practică a primei versiuni a sistemului presupunea rezolvarea unor aspecte esențiale cu privire la perspectivele dezvoltării (pe lângă problemele inerente legate de crearea și materializarea sistemului, trebuia efectuat un studiu de piață corespunzător, trebuiau identificate și valorificate resursele cele mai accesibile etc.).

Deci, fiind confruntat la sfârșitul lui 1997 și începutul lui 1998 cu un nou proiect, realizarea și dezvoltarea acestuia a necesitat implicarea mea totală, nemairămânându-mi resurse pentru continuarea activităților anterioare. Însă experiența și cunoștințele acumulate prin acestea au reprezentat un factor foarte important (cel puțin din punctul meu de vedere) în conceperea și realizarea reușită a noului sistem. Un studiu personal pe care l-am făcut în cursul anului 1998 a revelat o serie de fapte, printre care:

- programele de calculator gratuite (inclusiv cele din categoria *Open Source* — programe la care este inclusă și sursa în distribuție) sunt din ce în ce mai răspândite, de la sisteme de operare (cum ar fi Linux) până la servere pentru baze de date;
- împreună cu o serie de anexe, programele client dedicate inițial numai

vizualizării de fișiere în format HTML tind să devină sisteme care cuprind utilități de vizualizare multiple ale multor formate (aplicabile entităților de tip text, grafică, sunet și imagine);

- programele client amintite mai sus încep să suporte în mod fiabil anumite limbaje de programare (JavaScript, VBScript) care tind să ofere o instrumentație sub o manieră unificată pentru mai multe platforme;
- infrastructura existentă în România, împreună cu o relativ slabă cunoaștere a caracteristicilor și posibilităților de utilizare a rețelelor de calculatoare, îndepărtează momentul utilizării intensive a tehnologiilor client-server.

În urma realizării unei colecții semnificative de informații, ținând cont de observațiile prezentate, mi-am pus problema dacă este posibilă realizarea de aplicații informatice cu următoarele caracteristici:

- să poată fi utilizate fără conectare la Internet;
- să imite o tehnologie client-server pentru a beneficia de avantajele oferite de acesta (și pentru a putea fi ulterior transformate pentru utilizarea resurselor oferite prin Internet fără mari eforturi);
- să fie construite utilizând software gratuit și care poate fi distribuit liber (pentru a stimula activitățile de acest fel);
- să nu necesite în mod obligatoriu calculatoare superechipe sau rapide pentru exploatare;
- să nu depindă, dacă este posibil, de tipul sistemului de operare (Windows, Unix, MacOS) al platformei client utilizate.

Dacă aceste condiții ar putea fi îndeplinite simultan, o serie de probleme legate de distribuția de baze de cunoștințe sistematizate și structurate ar putea fi rezolvate în condiții economice și performante, iar răspândirea lor ar putea fi mult mai largă, contribuind eventual la formarea utilizatorilor în vederea folosirii de tehnologii client-server. Aceste rațiuni m-au condus la realizarea unor aplicații implementate sub sistemul AIC Info (prezentate în paragraful 4.2.3). Rezultatele obținute au fost valorificate atât prin lucrările prezentate și publicate la diverse manifestări (expoziția Ambient din 1999, conferințele internaționale de construcții ÉPKO'98 și ÉPKO'99), cât și prin contracte comerciale (indispensabile asigurării resurselor materiale necesare).

Deși sistemul AIC Info întrunește majoritatea caracteristicilor unui sistem expert (bază de cunoștințe, prelucrarea "inteligentă" a acestora etc.) fără a fi totuși în sens tradițional așa ceva, datorită tehnicilor și instrumentelor informatice existente cred că poate depăși performanțele unui sistem expert "clasic", deci nu știu dacă există rațiunea sau necesitatea să fie încorsetat sub forma unui asemenea sistem. Având în vedere anvergura actuală la care a ajuns, sistemul probabil depășește limitele unei teze de doctorat, concurând cu sisteme comerciale dezvoltate de către profesioniști. La apariția pe orizontul pieței românești de construcții, foarte mulți considerau că este prea devreme pentru un asemenea

sistem, eu însă eram convins că lansarea acestuia era necesară, nu numai pentru acoperirea unui gol (mai mult sau mai puțin existent la acea dată), dar mai ales pentru că eram încredințat în ceea ce privea utilitatea acestuia și știam că maturizarea și impunerea lui va cere timp. Beneficiind și de ajutorul și concursul unor oameni deosebiți am reușit realizarea unui sistem bazat pe cunoștințe care să beneficieze de avantajele oferite de către sistemele deschise, să nu depindă într-o mare măsură de echipamentele disponibile, să nu necesite resurse locale nejustificate și să poată fi utilizabil cât mai performant adresându-se unui public cât mai larg. Această bază de cunoștințe dispune de o serie de instrumente cu caracter mai mult sau mai puțin expert și ceea ce mi se pare foarte important: am avut ocazia să constat și personal că sistemul reușea substituirea cel puțin parțială a unor specialiști umani în ceea ce privește informarea unor utilizatori cu privire la diverse materiale și tehnologii existente în țară. Poate par lipsit de modestie afirmând acest lucru, însă nu pot descrie bucuria care m-a cuprins văzându-mi studenții folosind sistemul realizat, cu ușurință firească (neavând nici o informație premergătoare referitoare la sistem) sau surpriza unor colegi prin aflarea unor informații nebănuite sau anterior căutate, fără a se lovi de greutate în utilizarea sistemului.

Realizarea și dezvoltarea acestui sistem mi-a adus foarte multe satisfacții (în primul rând morale). De la interesul manifestat de către diverse persoane și firme, nu numai în ceea ce privește accesul la resursele oferite cât mai ales la participarea în dezvoltarea și diversificarea acestora, până la oferte de colaborare și comenzi apărute chiar și de peste hotare. Astfel, după o primă etapă (destul de scurtă) de colaborare cu AEC Media din S.U.A. (axată pe activități publicitare legate de construcții), am ajuns la o colaborare care funcționează și în prezent, cu Ceramlinks, materializată prin schimburi dinamice de legături și adrese cu alte situri dedicate diverselor activități din construcții. Spre mândria mea și a colaboratorilor, consultând rapoartele statistice oferite de Ceramlinks, în primele șase luni de colaborare, situl AIC Info era printre primele opt cele mai vizitate domenii și deși în ultima vreme ne-a crescut traficul în cifre absolute, de circa trei luni ne aflăm pe poziția a 12-a din clasamentul publicat (<http://www.ceramlinks.com/cerambanners/us/top20visited.asp>). Din păcate trebuie să recunosc că majoritatea dominantă a vizitatorilor sunt din afara țării, în special din America și Asia. Cunoscând însă potențialul materiei cenușii din țara noastră, prezent inevitabil și în domeniul construcțiilor, sperând și în transformarea benefică a mediului economic general, sunt convins că sistemul realizat va juca un rol mai însemnat în viitor. Susținerea lui comercială, precum și interesul unor "sponsori" capabili, pot asigura menținerea unui nivel performant și chiar dezvoltarea și diversificarea sistemului în continuare.

Din punctul meu de vedere, îmbinarea tehnologiilor informatice moderne cu teoriile privind reprezentarea și manevrarea cunoștințelor prin realizarea bazei de cunoștințe prezentate înseamnă un succes semnificativ. Utilizarea capacităților oferite de HTML, JavaScript, Pearl, serverul Apache, Postgres și alte instrumente, exploatarea mediului de lucru oferit de Internet conferă sistemului realizat trăsături care evidențiază actualitatea sistemului. Prin modelarea cunoștințelor oferite am reușit îmbinarea vechiului și a noului într-o manieră care, acum când scriu aceste rânduri, tinde să fie deja considerată clasică (dar la începutul

anului 1998 reprezenta încă o noutate). Structurarea realizată în vederea generării dinamice a paginilor de web, prin utilizarea bazelor de date și codificarea informațiilor accesibile, a necesitat o muncă de pionierat, atât din partea mea cât și din partea colegilor implicați, ținând cont de faptul că învățarea, asimilarea și aplicarea cunoștințelor necesare s-a făcut de cele mai multe ori în paralel cu realizarea sistemului, în mod autodidact. Realizările conexe sistemului AIC Info subliniază eficiența modelării abordate mai ales din punctul de vedere al legăturii dintre costul și exploatarea resurselor, subliniind și importanța abordării multidisciplinare a unor probleme legate de informare și cunoaștere. Prin rezultatele activității prezentate cred că am reușit deschiderea unui nou capitol în ceea ce privește performanța în activitatea de construcții.

Din activitatea mea de cercetare aplicată, precum și din rezultatele obținute în urma realizării și dezvoltării sistemului pe bază de cunoștințe AIC Info (dedicat exclusiv domeniului construcțiilor), aș dori să punctez următoarele:

- strângerea și prelucrarea informațiilor specifice, modelarea și structurarea acestora în funcție de conținut și context;
- conceperea și implementarea unei topologii informaționale în cadrul sistemului dezvoltat;
- codificarea paginilor accesibile prin sit și a capitolului *Software* de pe CD-ROM;
- conceperea și realizarea unor aplicații specifice (prin JavaScript) utilizând resursele disponibile în soluții client-server și soluții locale (dintre care aș reaminti: OPS, instrumentul dezvoltat pentru Rigips România, mini-sistemul expert pentru alegerea vopselelor);
- realizarea catalogului electronic al firmei Deceuninck pentru reprezentanțele din România și Rusia.

În încheiere, metafranzând scopul realizării și dezvoltării unui sistem bazat pe cunoștințe dedicat exclusiv construcțiilor și deocamdată unic de felul lui în țara noastră, pot să spun că dacă la început am dorit plantarea și creșterea unui copăcel prin conceperea sistemului AIC Info, în prezent am ajuns să-i purtăm de grijă unei întregi grădini prin care sperăm să-i ajutăm pe toți cei interesați, indiferent dacă fac parte din speciile de constructori care se târăsc sau care doresc să-și ia zborul către înălțimi.

Firește, finalizarea acestei teze nu înseamnă și sfârșitul contribuțiilor mele la dezvoltarea sistemelor expert în domeniul construcțiilor. Ca dovadă, prototipul sistemului expert pentru alegerea vopselelor a fost contractat cu firma NIGE Impex s.r.l. și implementarea lui este aproape de finalizare, iar sistemul AIC Info este dezvoltat în continuare, atât din punctul de vedere al îmbogățirii bazei de cunoștințe cât și din perspectiva extinderii instrumentației oferite. Interesul real pentru sisteme informatice performante, manifestat și de către specialiștii din domeniul construcțiilor și de către studenți, mă îndeamnă la continuarea activității în domeniul sistemelor expert, prin elaborarea de noi sisteme dedicate și valorificarea cunoștințelor dobândite în procesul didactic și educațional.

BIBLIOGRAFIE

- 1 * AIC Info, <http://www.aic.info.ro>, 1999
- 2 * AutoLISP Programmer's Reference, Autodesk Ltd., London, 1988
- 3 * CAD report, Computer Press Agora (1997 - 1998)
- 4 * CLIPS, <http://www.ghg.net/clips/CLIPS.html>, 1999
- 5 * CMU Artificial Intelligence Repository, <http://www.cs.cmu.edu/Groups/AI/html/repository.html>
- 6 * Expert System Toolkit (Turbo Expert), Knowledge Engineering Manual, Thinking Software Inc., Woodside (N.Y.), 1986
- 7 Interview mit Peter Raulefs über "Künstliche Intelligenz" in Deutschland, Diebold Management Report, Nr. 11/12, 1983
- 8 * JESS - Expert System Shell for Java Programming, <http://herzberg.ca.sandia.gov/jess>, 1997
- 9 * KI – ein Market im Aufbruch, Diebold Management Report, Nr. 11/12, 1983, p.14
- 10 * Personal Communication Form., John W. Lewis General Electric Corporate Research and Development Center, Schenectady, New York 12345, 1983
- 11 * Új Alaplap, Új Alaplap Kiadó Kft, Budapest, 1997 - 1999
- 12 ADELI, H.(ed.): Expert Systems in Construction & Civil Engineering, Chapman Hall, London, 1988
- 13 ADELI, H. - AL-RIJLEH, M.M.: A Knowledge-based Expert System for Design of Roof Trusses, Microcomputers in Civ. Eng., No. 2, 1987, p.179-195
- 14 ADELI, H. - BALASUBRAMANYAM, K.V.: Expert Systems for Structural Design A New Generation, Prentice Hall, 1988
- 15 AGHA, G.: ACTORS: A Model of Concurrent Computation in Distributed Systems, MIT Press, Cambridge, 1986
- 16 AGHA, G. - HEWITT, C.E.: Concurrent Programming Using Actors, In: Object-Oriented Concurrent Programming (YONEZAWA, A. - TOKORO, M. eds.), MIT Press, 1987, p.37-54
- 17 AHUJA, H.N. - NANDAKUMAR, V.: Simulation Model to Forecast Project Completion Time, Journal of Construction Engineering and Management 111, (4), 1985, p.325-342
- 18 ALIA, C. - CANUT, M.F. - VINCENT, B.: Des machines apprenantes pour une modélisation dynamique des systèmes de décision. Al Treilea Congres European de Sistemică, Roma, 1-4 octombrie, 1996
- 19 ALKASS, S.T. - HARRIS, F.C.: An Expert System for Claim Analysis, Proc. of 3rd Int. Conf. on Structure and Civil Engineering Computing, CIVIL-COMP Press, Edinburgh, England, 1987, p.39-44
- 20 ALKASS, S. - HARRIS, F.: Expert System for Earthmoving Equipment Selection in Road Construction, ASCE JCEM, V114, No.3, Sept.1988, p.426-440
- 21 ALKASS, S.T. - HARRIS, F.C.: An Expert System for the Earthmoving Plant Selection in Road Construction, ASCE Journal of Construction Engineering and Management, Vol.110, No.4, Sept. 1988
- 22 ALKASS, S.T.: An Expert System Applied to Earthmoving Operations and Equipment Selection, Ph.D. thesis, Civ. Eng. Dept., Loughborough University, England, 1988
- 23 ARDITI, D - RIAD, N.: Selection of an Expert System Shell for a Computerized Mediator - A Knowledge Based Expert System for Claims Management, Report, State of Illinois Technology Commercialization Center, 1988
- 24 ARDITI, D. - PATEL, B.K.: Expert Systems in Claims Management, International Journal of Project Management, 1989
- 25 ARDITI, D. - PATEL, B.K.: Impact Analysis of Owner-Directed Acceleration, Journal of Construction Engineering and Management, ASCE, Vol. 115, No. 1, 1989, p.144-160
- 26 ASHLEY, K.D. - RISSLAND, E.L.: A Case-based Approach to Modeling Legal Expertise, IEEE Expert, Fall 1988, p.70-77
- 27 BAILLY, C. - CHALLINE, J. F. - FERRI, H. C. - GEOESS, P. Y. - MARCHESIN, B.: Les langages orientés objets, Cepadues, Toulouse, 1987
- 28 BANERJEE, J. - CHOU, H.T. - GARZA, J.F. - KIM, W. - WOELK, D. - BALLOU, N.: Data Model Issues for Object-Oriented Applications, ACM Transactions on Office Information Systems, Jan. 1987, Vol. 5, No. 1, p.3-26
- 29 BARTH, S.W.: ERS User Manual, PAR Technology Corp., PAR Technology Park, 220 Seneca Turnpike, New Hartford, New York 13413, 1983
- 30 BEECH, D. - MAHBOD, B.: Generalized Version Control in an Object-Oriented Database, Proc. of the Fourth Int. Conf. on Data Engineering, USA, Feb. 1-5 1988, p.14-22

BIBLIOGRAFIE

- 31 BENCHIMOL, G. - LEVINE, P. - POMEROL, J. C.: Sisteme expert în întreprindere, Editura Tehnica, Bucuresti, 1993
- 32 BENNET, J. - CREARY, L. - ENGELMORE, R.S. - MELOSH, R.: SACON. A Knowledge-Based Consultant in Structural Analysis. Heuristic Programming, Project Report No. HPP-78-28, also Report No. STAN-CS-78-699, Computer Science Department, Stanford University, 1978
- 33 BENNET, J.K.: Distributed SmallTalk: Inheritance and Reactiveness in Distributed Systems, Ph.D. Dissertation, University of Washington, Seattle, USA, 1988
- 34 BIRTWISTLE, G. - DAHL, O. - MYRHAUG, B. - NYGAARD, K.: Simula Begin, Petrocelli/Charter, 1973
- 35 BLAKE, E. - COOK, S.: On Including Part Hierarchies in Object-Oriented Languages, with an Implementation in Smalltalk, Proc. of ECOOP 87, Paris, June 1987, p.41-50
- 36 BOBROW, D.G. - KAHN, K. - KICZALES, G. - MASINSTER, L. - STEFIK, M. - ZDYBEL, F.: Common Loops Merging Lisp and Object-Oriented Programming, Proc. of OOPSLA 86, USA, Sept. 29 -Oct. 2, p.17-29
- 37 BOBROW, D. - DeMICHIEL, L.G. - GABRIEL, R.P. - KEENE, S.E. - KICZALES, G. - MOON, D.: Common Lisp Object System Specification, ANSI X3 J.13 Document 88-002R, American National Standards Institute, Washington, D.C., 1988
- 38 BONISSONE, P.P. - JOHNSON, H.E.: An Expert System for Fault Diagnosis in Diesel Electric Locomotives, General Electric Corporate Research and Development Center, Schenectady, New York 12345, 1983
- 39 BOOCH, G.: Ingénierie du logiciel avec ADA, InterEditions, Paris, 1988
- 40 BOOSE, J. H.: Expertise Transfer for Expert System Design, Elsevier, Amsterdam, 1986
- 41 BOTEZAT, C. - CONSTANTINESCU, D. - DINU, V. - MALITA, M.: SEDAN - Manual de utilizare, I.P.C.T. Bucuresti, 1986 (BNP)
- 42 BRANDON, P.S.: Expert Systems - Modelling Professional Expertise? Proc. of 5th CIB-W65 Intl Symp., London, Sept. 1987, p.580-590
- 43 BRETON, Ph.: Histoire de l'informatique, Editions La Découverte, Paris, 1987
- 44 BRYANT, N.: Szakértői rendszerek, Novotrade Rt., Budapest, 1989
- 45 BUCHANAN, B.G. - FEIGENBAUM, E.A.: DENDRAL and meta-DENDRAL: Their Applications Dimension, Journal of Artificial Intelligence, 11, 1978, p.5-24
- 46 CARR, R.I. - CHANG, D.Y. - MORUA-PADILLA, E.: Knowledge Based Simulation of Construction, Proc. of 4th Int. Conf. on Robotics and AI in Building, Haifa, Izrael, June 1987, Technion, p.190-207
- 47 CHAN, W.T. - PAULSON, B.: Constraint-Based Model of Design, In: Microcomputer-Based Expert Systems in Civil Engineering (ADELI, H. ed.), ASCE Publications, New York, 1988
- 48 CHANG, T.C.: Network Resource Allocation Using an Expert System with Fuzzy Logic Reasoning, Ph.D. thesis, Construction Engineering and Management Division, Department of Civil Engineering, University of California, Berkley, Aug. 1987
- 49 CLOCKSIN, W.F. - MELLISH, C.S.: Programming in PROLOG, Springer-Verlag, New York, 1981
- 50 CORRADI, A. - LEONARDI, L.: How to Embed Concurrency Within an Object Environment: Parallel Objects, In: Parallel Processing and Applications, Proc. of the Int. Conf. on Parallel Processing and Applications, L'Aquila, Italia, Elsevier, Amsterdam, 1988, p.79-84
- 51 COX, B. - HUNT, B.: Objects, Icons, and Software IC's, Byte, 11 (8), 1986
- 52 DAVIS, R.: Metarules: Reasoning about Control, Artificial Intelligence, 1980, 15, p.179-222
- 53 DAVIS, R.: Content Reference: Reasoning about Rules, Artificial Intelligence, 1980, 15, p.223-239
- 54 DAVIS, R. - LENAT, D. B.: Knowledge-based Systems in Artificial Intelligence, McGraw-Hill, New York, 1982
- 55 DELAHAYE, J. P.: Outils logiques pour l'intelligence artificielle, Eyrolles, Paris, 1987
- 56 DELAHAYE, J. P.: Systems experts: organisation et programmation des bases de connaissances en calcul propositionnel, Eyrolles, Paris, 1987
- 57 DIEKMANN, J.E.: Claims Analysis & Computer Reasoning, ASCE JCEM, V110, No.4, Dec. 1984, p.391-407
- 58 DUBIOS, D. - PRADE, H.: Possibility Theory – Approach to Computerized Processing of Uncertainty, Plenum Press, New York, 1988
- 59 DUDA, R.O. - HART, P.E. - GASCHNIG, J.: Model Design, in The PROSPECTOR Consultant System for Mineral Exploration. Expert Systems in the Microelectronic Age (ed. D. Michie), Edinburgh

BIBLIOGRAFIE

- University Press, Edinburgh, 1979
- 60 ECHEVERRY, D. - IBBS, C.W. - KIM, S.: Generation of Construction Schedules – A Knowledge-Based Approach, Proc. of 6th Int. Symp. on Automation and Robotics in Construction, S. Francisco, California, SUA, June 6-9 1989
- 61 ERMINE, J.L.: Les systèmes de connaissances, Ed. Hermès, Paris, 1996
- 62 FAGHRI, A. et al.: Expert System for the Evaluation of Rail/Highway Crossings, ASCE JCCE, V2, Â1, Jan. 1988, p.21-37
- 63 FARRENY H. - GHALLAB, M.: Eléments d'intelligence artificielle, Hermes, Paris, 1990
- 64 FEIGENBAUM, E.A. - McCORDUCK, P.: The 5th Generation, Addison-Wesley, Reading, Massachusetts, 1983
- 65 FEIJO, B.: XLOG –A Logic Programming Environment, Technical Report ESL 1/86, Expert Systems Laboratory, Department of Civil Engineering, Imperial College, London, 1986
- 66 FENVES, S.J. et al.: An Integrated Software Environment for Building Design & Construction, Proc. 5th Conf. on Cptg. in Civ. Eng. Alexandris, Virginia, USA, March 1988, p.21-32
- 67 FENVES, G.L.: Object-Oriented Models for Engineering Data, Proc. of the ASCE 6th Conf. on Computing in Civil Engineering, Atlanta, Georgia, SUA, Sept. 11-13, 1989
- 68 FIGGINS, T. - BARTH, S.W. - GATES, K.: Functional Description for the Duplex Army Radio/Radar Targeting Aid (DART), PART Report 83-117, written for RADC Contract F30602-81-C-0263, PAR Technology Corp., PAR Technology Park, 220 Seneca Turnpike, New Hartford, New York 13413, 1983
- 69 FRUCHTER, R. - GLUCK, J. - GOLD, Y.I.: Application of AI Programming Techniques to the Analysis of Structures, Proc. of 3rd Int. Conf. on Civil and Structural Engineering Computing, Vol. AI, 1987
- 70 FUCHI, K.: Aiming for Knowledge Information, Processing Systems, Proc. of the Int. Conf. on Fifth Generation Computer Systems, Tokio, Japan, Oct. 19-22, 1981
- 71 FURUTA, H. - OZAKI, Y. - SHIRAISHI, N.: Fatigue Analysis of Reinforced Concrete Decks Based on Fuzzy Sets Theory, Proc. of ICASP-5, Vol.1, 1987, p.198-205
- 72 GARDNER, H.: Histoire de la révolution cognitive: la nouvelle science de l'esprit., Bibliothèque scientifique Payot, 1993
- 73 GARRETT, J.H. - FENVES, S.J.: A Knowledge-Based Standards Processor for Structural Component Design, Engineering with Computers, 1987, Vol. 2, p.219-238
- 74 GOBESZ, F. - COCHECI T. - MAN, M. - GOBESZ, F.ZS. - SCHIR, R. - KATONA, M.: Programul DOC. Consultarea documentațiilor de pe suport magnetic. Lito. I.P.Cluj-Napoca, 1987; contract cu I.P.C.T.București nr.10174b/1987 fazele PTh și DE
- 75 GOBESZ F. - BOLOGA I. - MAN M. - GOBESZ, F.ZS.: Manuale de utilizare pe suport magnetic: generare și exploatare. Lucrări S.N.I.C. 6, pag.143-146, Timișoara, 1988
- 76 GOBESZ F.ZS.: Produsul informatic conversațional pentru consultarea bibliotecii de programe destinate proiectării în construcții, editată de ICCPDC=IPCT București, Proiect de absolvire curs F1A: Analiza și proiectarea sistemelor informatice de gestiune Felix-Mini, C.T.C.E.-Cluj-Napoca, 1989
- 77 GOBESZ, F. - BOLOGA I. - GOBESZ, F.ZS.: Subprograme pentru generarea caracteristicilor geometrico-mecanice ale secțiunilor compuse. Lucrările celei de a VI-a Conf.de Construcții Metalice, vol. 1, pag.136-143, Timișoara, 1991
- 78 GOBESZ, F. - BOLOGA I. - GOBESZ, F.ZS.: Dezvoltarea de sisteme expert în domeniul calculului construcțiilor. Sistemul expert BPC (Documentare în Biblioteca Programelor de Construcții). Model matematic; Baza de date. Raport tehnic; contract cu M.I.S. 1990; I.P.Cluj-Napoca 101 F, faza I/1991
- 79 GOBESZ, F. - BOLOGA I. - GOBESZ, F.ZS.: Dezvoltarea de sisteme expert în domeniul calculului construcțiilor. Sistemul expert BPC (Documentare în Biblioteca Programelor de Construcții). Interfața utilizator-sistem. Raport tehnic; contract cu M.I.S. 1990; I.P.Cluj-Napoca 101 F, faza II/1992
- 80 GOBESZ, F.ZS.: Metode și modele în analiza structurilor. (Referat nr.1) U.T.Cluj-Napoca, Facultatea de Construcții, 1992
- 81 GOBESZ, F.ZS.: Sisteme expert pentru analiza structurilor. (Referat nr.2) U.T.Cluj-Napoca, Facultatea de Construcții, 1993
- 82 GOBESZ, F.ZS.: Generarea sistemelor expert. (Referat nr.3) U.T.Cluj-Napoca, Facultatea de Construcții, 1994

BIBLIOGRAFIE

- 83 GOBESZ, F.ZS. - TÉGLÁS, A.: Tervezni a tervezést. În lucrările conferinței internaționale de construcții "ÉPKO'97: A századvég építészete - hogyan tovább?", ÉMT, Sovata, 6-8 iunie 1997, p.20-21
- 84 GOBESZ, F.ZS.: Számítógép és tervezés. Kritikus magasság. În lucrările conferinței internaționale de construcții ÉPKO'98, ÉMT, Odorheiu Secuiesc, 5-7 iunie 1998, p.57-62
- 85 GOBESZ, F.ZS. - TURDA, D.V.: AIC Info. În lucrările conferinței internaționale de construcții ÉPKO'99, ÉMT, Odorheiu Secuiesc, 4-6 iunie 1999, p.12-15
- 86 GRAY, C. - COLES, E.: A predictive model of the building design process, Proc. of 4th Int. Conf. on Robotics and AI in Building, Haifa, Izrael, June 1987, Technion, p.442-446
- 87 GRAY, C.: Intelligent Construction Time and Cost Analysis, Construction Management and Economics, 1986, No. 4
- 88 GOLDBERG, A. - ROBSON, D.: SmallTalk-80: The Language and its Implementation, Addison-Wesley, 1983
- 89 GRUMBACH, A.: Cognition artificielle, du réflexe à la réflexion, Addison-Wesley, 1994
- 90 HADDEN, W.J. - HADDEN, S.G.: Expert Systems for Environmental Regulation, Proc. Expert Systems in Government Symposium, IEEE, McLean, Virginia, (ed. K.N. Karna) 1985, p.558-565
- 91 HADDEN, W.J. - HADDEN, S.G.: An Intelligent Advisory System for Private Sewage Facility Permits: Two Years in the Trenches. Proc. 3rd Annual Expert Systems in Government Conf., IEEE, Washington, (ed. H.J. Antonisse, J.W. Benoit, B.G. Silverman) 1987, p.280-284
- 92 HAMIANI, A.: CONSITE: A Knowledge-Based Expert System Framework for Construction Site Layout, Ph.D. thesis, Department of Civil Engineering, University of Texas at Austin, Austin, Texas, USA, 1987
- 93 HAMIANI, A. - POPESCU, C.: CONSITE: A Knowledge-Based Expert System for Site Layout, Proc. of 5th Conf. on Computing in Civ. Eng., Washington, D.C., USA, March 1988
- 94 HAMIANI, A.: Knowledge Representation for the Site Layout Problem, Proc. of 6th Conf. on Cmpgt in Civ. Eng., Atlanta, GA, USA, Sept. 1989
- 95 HAMMOND, P.: APES: A Detailed Description. Research Report 82/10, Department of Computing, Imperial College of Science and Technology, University of London, 180 Queens Gate, London, SW7 2BZ, UK, 1982
- 96 HASLING, D.W. - CLANCEY, W.J. - RENNELS, G.: Strategic Explanations For a Diagnostic Consultation System, Int. J. Man-Machine Studies, 1984, No. 20, p.3-19
- 97 HAYES-ROTH, F. - WATTERMAN, D. A. - LENAT, D. B.: Building Expert Systems, Addison-Wesley, Reading, 1983
- 98 HENDRICKSON, A.M. - GOROSTIZA, C. - REHAK, D. - MILLER, E. - LIM, P.: Expert System for Construction Planning, Journal of Computing in Civil Engineering, ASCE, Vol. 1, No. 4, Oct. 1987
- 99 HOSKING, J.G. - MUGRIDGE, W.B. - BUIS, M.: FireCode: A Case Study in the Application of Expert-Systems Techniques to a Design Code. Environment and Planning B: Planning and Design, 1987, vol. 14, p.267-280
- 100 HUNT, E.B. - MARTIN, J. - STONE, P.: Experiments in Inductive Learning, Academic Press, New York, 1966
- 101 IBBS, C.W. - DE LA GARZA, J.M.: Knowledge Engineering for Construction Scheduling Analysis System, in: Expert Systems in Construction and Structural Engineering (H.Adeli ed.), Chapman and Hall Ltd., New York, 1988, p.137-159
- 102 ISHII, K. - HAYAMI, S.: Expert Systems in Japan, IEEE Expert, Summer 1988, p.69-74
- 103 JENSEN, K. - WIRTH, N.: Pascal User Manual and Report, Springer-Verlag, New York, 1974
- 104 JINTAE, L.: Design rationale systems: understanding the issues. IEEE Journal, May-June 1997
- 105 JUL, E.: Object Mobility in Emerald, Ph.D. Dissertation, University of Washington, Seattle, 1988
- 106 KARTAM, N. - LEVITT, R.E.: Intelligent Planning of Construction Projects with Repeated Cycles of Operation, ASCE Journal of Computing in Civil Engineering, Special Issue: Knowledge-based Approaches to Planning and Design, Fall 1989
- 107 KAUFMANN, A.: Nouvelles logiques pour l'intelligence artificielle, Hermes, Paris, 1987
- 108 KAVANAGH, D.P.: SIREN: A repetitive construction simulation model, Journal of Construction Engineering and Management 111, (3), 1985, p.308-323
- 109 KLEMENTIS, O.: Contribuții privind realizarea unui sistem expert pentru proiectarea itinerarului tehnologic și a sculelor aferente în cazul procesului de deformare la rece pentru gama pieselor axial

BIBLIOGRAFIE

- simetrice pline, realizabile cu ajutorul matritelor. Teză de doctorat, U.T. Cluj-Napoca, 1999
- 110 KORNELL, J.: A VAX Running Expert Built Using Automated Knowledge Acquisition, Proceedings of the First Conference on AI Applications, IEEE Computer Society, December 1984, p.38-41
- 111 KOWALSKI, R.: Logic for Problem Solving, Elsevier, New York, 1979
- 112 KRISTENSEN, B.B. - MADSEN, O.L. - MOLLER-PEDERSEN, B. - NYGAARD, K.: The BETA Programming Language, In: Research Directions in Object-Oriented Languages, (SHRIVER, B. - WEGNER, P. eds.), MIT Press, 1987
- 113 KUMAR, B.: Knowledge Processing for Structural Design, Ph.D. dissertation, University of Edinburgh, May 1989
- 114 KUMAR, B. - CHUNG, P.W.H. - RAE, R.H. - TOPPING, B.H.W.: A Knowledge Based Approach to Structural Engineering, Proceedings, CIVIL-COMP 1987, London, p.79-88
- 115 KUMAR, B. - CHUNG, P.W.H. - TOPPING, B.H.V.: Approaches to FORTRAN-PROLOG Interfacing for an Expert System, Proceedings, CIVIL-COMP 1987, London, p.15-20
- 116 LABIDI, S. - LEJOUAD, W.: De l'intelligence artificielle distribuée aux systèmes Multi-Agents. Rapport de recherche de l'INRIA, no.2004, august 1993
- 117 LANDIS, G.S.: Design Evolution and History in an Object-Oriented CAD/CAM Database, Proc. of COMP-CON 86, USA, Mar. 3-6 1986, p.297-303
- 118 LANSDOWN, J.: Expert Systems: their impact on the Construction Industry, Report to RIBA Conference Fund, England, 1982
- 119 LAW, K.H. - JONANEH, M.K. - SPOONER, D.L.: Abstraction Database Concept for Engineering Modeling, Engineering with Computers, Vol. 2, No. 2, 1987, p.79-94
- 120 LEHNER, P.E.: - DONNELL, M.L.: Maintainability Design Expert System, PAR Technology Corporation Report, 1983
- 121 LESTER, J.: Lawyers on a Microchip, Civil Engineering, ASCE, Vol. 57, No. 6, 1987
- 122 LETIA, I. A. - GRECU, D. - TATAR, M.: Inteligența artificială (îndrumător de laborator), U.T. Cluj-Napoca, 1992
- 123 LEVITT, R.E. - KUNZ, J.C.: Using Knowledge of Construction and Project Management for Automated Schedule Updating, Project Management Journal, V 16, No.5, Dec. 1985, p.57-76
- 124 LEVITT, R.E. - KUNZ, J.C. - KARTAM, N.A.: Using Artificial Intelligence Techniques for Automated Planning and Scheduling, Proc. of 4th Int. Conf. on Robotics and AI in Building, Haifa, Izrael, June 1987, Technion, p.27-52
- 125 LIEBERMAN, H.: Using Prototypical Objects to Implement Shared Behavior in Object-Oriented Programming Systems, Languages and Applications, Portland, Oregon, SUA, Special Issue of SIGPLAN Notices, 21, 1986
- 126 LINDSAY, P. H. - NORMAN, D. A.: Traitement de l'information et comportement humain, Editions Etudes Vivantes, Montréal, 1980
- 127 LLOYD, J. W.: Foundations of Logic Programming, Springer Verlag, Heidelberg, 1984
- 128 LOGCHER, R.D.: Adding Knowledge Based Systems Technology to Project Control Systems, Project Controls: Needs and Solutions, ASCE, 1987
- 129 MAHER, M.L. - FENVES, S.J.: HI-RISE - An Expert System for the Preliminary Design of High Rise Buildings, Knowledge Engineering in Computer-Aided Design (GERO, J. ed.), Elsevier Science Publishers, B.V. (North Holland), 1985, p.125-146
- 130 MAHER, M.L.: Expert Systems for Structural Design, Expert Systems in Engineering (ed. D.T.Pham), IFS Publications, Springer Verlag, 1988, p.147-161
- 131 MAHER, M.L.: HI-RISE: An Expert System for Preliminary Structural Design, Expert Systems for Engineering Design (ed. M.D.Rychener), Academic Press Inc., 1988, p.37-52
- 132 MAWDESLEY, M.J. et al: An Approach to Automatic Project Planning, 4th International Symposium on Robotics and Artificial Intelligence in Building Construction, Haifa, Israel, June 1987, p.897-917
- 133 McCABE, F.G.: Micro-PROLOG Reference Manual, Logic Programming Associates, 10 Burntwood Close, London, SW18 3JU, UK, 1981
- 134 McCORDUCK, P.: Machines Who Think, Freeman and Comp., New York, 1979
- 135 McDERMOTT, J.: R1: A Rule-Based Configurer of Computer Systems, Report CMU-CS-80-119, Department of Computer Science, Carnegie-Melon University, 1980

BIBLIOGRAFIE

- 136 McLAUGHLIN, S. - GERO, J.S.: Learning from Characterized Design, Proc. of 2nd Int. Conf. on Application of AI in Engineering, 1987
- 137 MEYROWITZ, N.: Intermedia. The architecture and construction of an object-oriented hypermedia system. Proceedings of the 1st Conference on Object-Oriented Programming, Portland, 1986, p.186-201
- 138 MICHAELSEN, R. - MICHIE, D.: Expert Systems in Business, Datamation, 1983, p.240
- 139 MICHIE, D. - MUGGLETON, S. - RIESE, C. - ZUBRICK, S.: RuleMaster: A Second Generation Knowledge Engineering Facility. Presented at the First Conference on Artificial Intelligence Applications, sponsored by the IEEE Computer Society, Denver, Colorado 5-7. December 1984, Radian Corp., P.O.Box 9948, Austin TX 78766
- 140 MILLER, G.R.: A LISP-Based, Object-Oriented Approach to Structural Analysis, Engineering with Computers, 4, 1988, p.197-203
- 141 MINSKY, M.: A framework for representing knowledge, In: P.H.Winston (ed.), The Psychology of Computer Vision, McGraw-Hill, New York, 1975.
- 142 MOON, D.A.: Object-Oriented Programming with Flavors, OOPSLA '86 Proceedings, SIGPLAN Notices, 21, 1986, p.1-8
- 143 NAGY, T. - GAULT, D. - NAGY, M.: Így készül a szakértői rendszer, Novotrade Rt., Budapest, 1988
- 144 NATKE, H.G. - DOLL, H. - HILDEBRANDT, P. - SCHÜTZE, M.: Bridge Condition Assessment using an Expert System, Structural Engineering Review, Vol.7, No.3, Elsevier Science Ltd., 1995, p.165-180
- 145 NAVINCHANDRA, D.S. - LOGCHER, R.D.: GHOST: Project Network Generator, Journal of Computing in Civil Engineering, ASCE, Vol. 2, No. 3, July 1988
- 146 NEWELL, A. - SIMON, H.: Human Problem Solving, Prentice-Hall, Englewood Cliffs, 1972
- 147 NIERSTRASZ, O.M.: Active Objects in Hybrid, OOPSLA '87 Proceedings, SIGPLAN Notices, 22, 1987, p.243-253
- 148 NILSSON, N. J.: Problem-solving Methods in Artificial Intelligence, McGraw-Hill, New York, 1971
- 149 O'BRIEN, P.D. - HALBERT, D.C. - KILIAN, M.F.: The Trellis Programming Environment, OOPSLA '87 Proceedings, SIGPLAN Notices, 22, 1987, p.91-102
- 150 O'CONNOR, M.J. - DE LA GARZA, J.M. - IBBS, C.W.: An Expert System for Construction Schedule Analysis, 1st Symposium on the Expert Systems in Civ. Eng., ASCE Spring Convention, Seattle, Washington, 1986
- 151 PAEK, Y. - ADELI, H.: SDL: An Environment for Building Integrated Structural Design Expert Systems, Microcomputer Knowledge-Based Expert Systems in Civil Engineering, Proc. of a Symposium, American Society of Civil Engineers, Nashville, Tennessee, 1988, p.40-52
- 152 PARSAYE, K - CHIGNELL, M.: Expert Systems for Experts, John Wiley, New York, 1988
- 153 PATERSON, A.: AL/X User Manual, Intelligent Terminals Ltd., 15 Canal St. Oxford, OX2 6BH, UK, 1981
- 154 PATERSON, A.: ACLS User Manual, Intelligent Terminals Ltd., 15 Canal St. Oxford, OX2 6BH, UK, 1982
- 155 PAULSON, B.C. - SOTOODEH-KHOO, H.: Expert Systems in Real-time Construction Operations, Proc. of 5th CIB-W65 Intl Symp, London, UK, Sept. 1987, p.554-565
- 156 PETCU, V. - SVASTA, C.: Calculul automat al elementelor din beton armat. Programe Basic, Editura Tehnică, București, 1986
- 157 POON, P. - HOSAIN, M.: Guru2: an Electronic Teaching Assistant, Proc. 5th Conf. on Cptg. in Civ. Eng. Alexandris, Virginia, USA, March 1988, p.608-617
- 158 PRINCE, V.: Vers une informatique cognitive dans les organisations. Ed. Masson, 1996
- 159 QUE Development Group: AutoCAD 12, Editura Teora, București, 1995
- 160 QUINLAN, J.R.: Discovering Rules by Induction from Large Collections of Examples, In: Expert Systems in the Micro-electronic Age (ed. D.Michie), Edinburgh University Press, Edinburgh, 1979
- 161 REITER, J.E.: AL/X: An Inference System for Probabilistic Reasoning, M.S. Thesis, Department of Computer Science, University of Illinois, Urbana, Illinois 61801, 1981
- 162 RICH, E.: Artificial Intelligence, McGraw-Hill, New York, 1983
- 163 ROUHANI, S. - KANGARI, R.: Landfill Site Selection, Microcomputers in Civil Engineering, No. 2, 1987, p.47-53

BIBLIOGRAFIE

- 164 ROSENMAN, M.A.: BUILD Expert System Shell Users Manual, Design Computing Unit, Department of Architectural Science, University of Sydney, Australia, 1988
- 165 ROSENMAN, M.A. - TOLHURST, S. - POSTMUS, A.: SOLAREXPERT Users Manual, Design Computing Unit, Department of Architectural Science, University of Sydney, Australia, 1988
- 166 SÁNTÁNÉ-TÓTH, E.: Tudásalapú technológia, szakértő rendszerek. Miskolci Egyetem Dunújvárosi Főiskolai Kar Kiadói Hivatala, Dunaújváros, 1998
- 167 SCHMUCKER, K.: Object-Oriented Programming for the Macintosh, Hayden Book Co., 1986
- 168 SERGOT, M.: A Query-the-User Facility for Logic Programming, In: Integrated Interactive Computing Systems (eds. P.Degano and E.Sandwell), North-Holland, New York, 1983
- 169 SHIRAIISHI, N. - FURUTA, H. - UMANO, M. - KAWAKAMI, K.: An Expert System for Damage Assessment of Reinforced Concrete Bridge Deck, Preprints of Second IFSA Congress, Vol.1, 1987, p.160-163
- 170 SHORTLIFFE, E.H. - BUCHANAN, B.G. - FEIGENBAUM, E.A.: Knowledge Engineering for Medical Decision Making: A Review of Computer-Based Clinical Decision, Aids Proc.IEEE, 67, 1979, p 1207-1224
- 171 SIMON, H. A.: The Sciences of Artificial, M.I.T. Press, Cambridge (Ma),1981
- 172 SLOMAN, A.: The Computer Revolution in Philosophy (Harvester Studies in Cognitive Science), The Harvester Press, Ltd., (1978) 2 Stanford Terrace, Hassocks, Sussex.
- 173 SOH, C.K. - SOH, A.K.: Example of Intelligent Structural Design System, ASCE JCCE, V2, Á4, Oct. 1988, p.329-345
- 174 SOTIRESCU, D.N. - STEFANESCU, M.: Raport asupra sistemului interactiv de programare DM-LISP, versiunea 1.0, In: Raport, Vol. 111, Lab. de calcul, Fac. de constructii, U.T. din Cluj-Napoca
- 175 SPRAGUE, R. H. Jr. - CARLSON, E. D.: Building Effective Decision Support Systems, Prentice-Hall, Englewood Cliffs, 1982
- 176 SRIRAM, D.: DESTINY - A Modell for Integrated Structural Design, Knowledge Engineering and Computer Modelling in CAD, Proc. of CAD'86, London, 1986, p.226-235
- 177 STEFIK, M. - BOBROW D.G.: Object-Oriented Programming: Themes and Variations, AI Magazine, Jan. 1986, p.40-62
- 178 STROUSTRUP, B.: The C++ Programming Language, Addison-Wesley, 1986
- 179 SZOKOLAY, S.V.: Thermal Design of Buildings, RAI Education Division, Canberra, Australia,1987
- 180 TARANATH, B.S.: Structural Analysis and Design of Tall Buildings, McGraw Hill, 1988
- 181 TIUCĂ, T.L.: Îndrumător de lucrări la tehnica prelucrării informațiilor, Lito. U.T. Cluj-Napoca, 1993
- 182 TOMMELEIN, I.D. et al.: Using Expert Systems for the Layout of Temporary Facilities on Construction Sites, Proc. of 5th CIB-W65 Intl Symp, London, UK, Sept. 1987, p.566-577
- 183 TOPPING, B.H.V. (ed.): Artificial Intelligence Techniques and Applications for Civil and Structural Engineers, CIVIL-COMP Press, Edinburgh, UK, 1989
- 184 TOPPING, B.H.V. (ed.): The Application of Artificial Intelligence Techniques to Civil and Structural Engineering, CIVIL-COMP Press, Edinburgh, UK, 1987
- 185 TOPPING, B.H.V. - KUMAR, B.: Knowledge Representation and Processing for Structural Engineering Design Codes, Engineering Applications of Artificial Intelligence, 1989
- 186 TYRVÄINEN, H.: Introduction of Finnish Ratu production Files, Technical Research Centre of Finland, Laboratory of Urban Planning and Building Design, 1988, 3p.+app. 15p.
- 187 VÁMOS T.: Expert Systems – Information Technology. Working Paper, Summer School, June 1983, Bulgaria
- 188 VAN MELLE, W.: A Domain-Independent System that Aids in Constructing Knowledge-Based Consultation Program, Ph.D. Dissertation, Report No. STAN-CS-80-820, Computer Science Department, Stanford University, 1980
- 189 WATANABE, T. - YONEZAWA, A.: Reflection in an Object-Oriented Concurrent Language, OOPSLA '88 Proceedings, SIGPLAN Notices, 23, 1988, p.306-315
- 190 WATERMAN, D. A.: A Guide to Expert Systems, Addison-Wesley, Reading, 1986
- 191 WEGNER, P.: Dimensions of Object-Based Language Design, OOPSLA '87 Proceedings, SIGPLAN Notices, 22, 1987, p.168-182
- 192 WILKINS, D.: Domain-Independent Planning: Representation and Plan Generation, Artificial Intelligence, Vol. 22, No. 3, April 1984

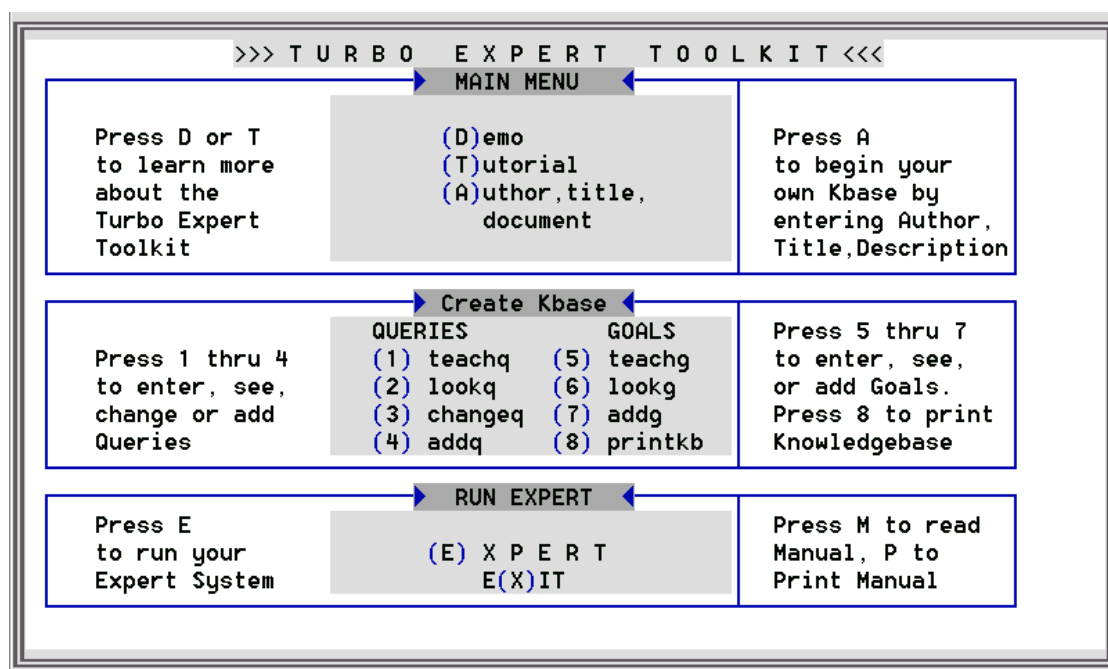
BIBLIOGRAFIE

- 193 WINSTON, P. H.: Inteligenta artificiala, Editura Tehnica, Bucuresti, 1981
- 194 ZABILSKI, R.J. - HALL, H.E.: Presented in 3-D, Civil Engineering, 59 (6), 1989, p.48-50
- 195 ZIMÁNYI, M. - KÁLMÁN, L. - FADGYAS, T.: A LISP programozási nyelv, Műszaki Könyvkiadó, Budapest, 1989
- 196 ZOZAYA-GOROSTIZA, C. - HENDRICKSON, C. - REHAK, D. - LIM, P.: CONSTRUCTION PLANEX: A Knowledge Intensive Planner for Construction Management, The 5th Int. Symp. on Robotics in Construction, June 6-8 1988, Tokyo, Japan, Japan Industrial Robot Association (JIRA), Tokyo, p.511-520
- 197 YOKOTE, Y. - TOKORO, M.: Concurrent Programming in Concurrent SmallTalk, In: Object-Oriented Concurrent Programming (YONEZAWA, A. - TOKORO, M. eds.), MIT Press, 1987

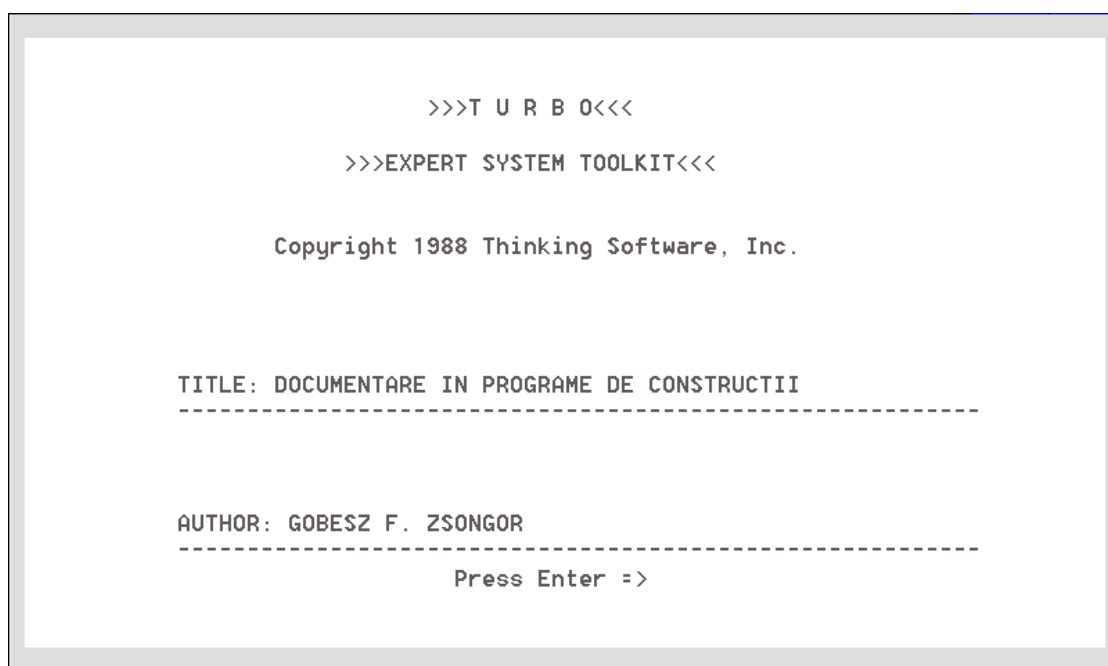
SISTEMUL EXPERT BPC

A. Sistemul expert BPC generat sub *Expert System Toolkit*

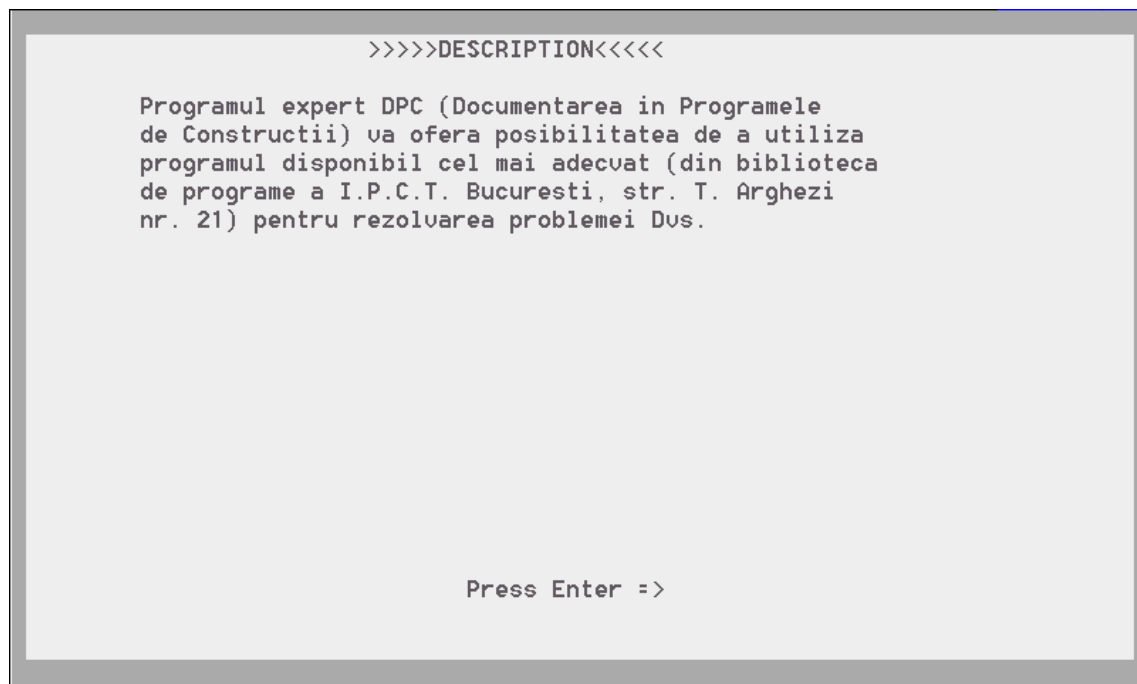
1. Captură de ecran realizată la lansarea versiunii din 1986 a mediului EXPERT SYSTEM TOOLKIT (cunoscut și sub numele TURBO EXPERT), realizat de către Thinking Software Incorporated din S.U.A.:



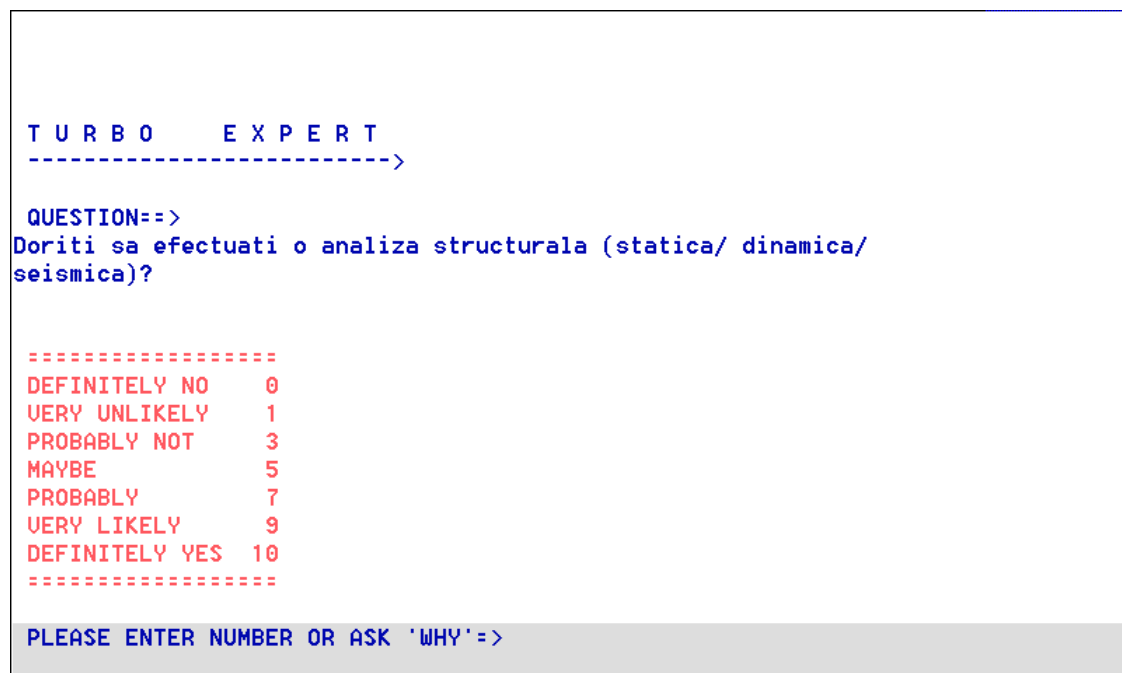
2. Mediul de lucru permite atât generarea și dezvoltarea prototipurilor limitate, cât și lansarea unei sesiuni de consultare a bazei de cunoștințe. Consultarea sistemului expert BPC (variantea transpusă sub acest mediu în anul 1990) începe cu ecranul următor:



3. Următorul ecran de prezentare oferă o descriere sumară a sistemului lansat:

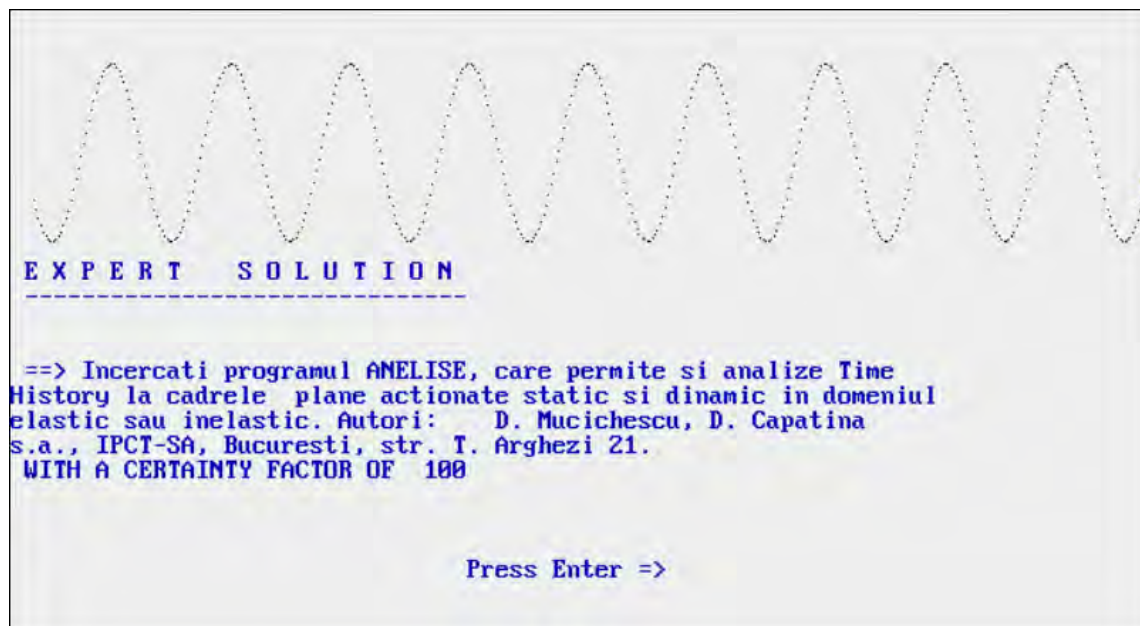


4. Sesiunea de consultare (accesibilă tot prin rularea mediului TURBO EXPERT, dar cu baza de cunoștințe corespunzătoare încărcată în prealabil) se face sub sistemul de operare DOS în regim text, elementele grafice fiind realizate cu ajutorul unor caractere din partea extinsă a setului de caractere încărcate. Regulile din baza de cunoștințe se execută în funcție de răspunsurile furnizate sistemului pe parcursul sesiunii desfășurate, în final obținând recomandările prin ecrane de genul următor:



Răspunsurile utilizatorului sunt exprimate prin cifrele din dreptul probabilităților prevăzute, existând și posibilitatea chestionării pentru obținerea unor explicații referitoare la rațiunea întrebării afișate.

5. Rezultatul este afișat grafic, fiind însă generat și un fișier text (ASCII) care conține atât recomandările sistemului cât și gradele de suitabilitate (la cerințele exprimate) a soluției oferite.



6. Baza de cunoștințe este compusă din trei părți principale:

- KBASED - conținând informații generale (afișate în cadrul ecranelor de prezentare a sistemului încărcat);
- KBASEA - baza de cunoștințe propriu-zisă, conținând atât scopurile, cât și regulile necesare (admite și precizarea probabilității într-o manieră simplă);
- KBASEQ - conținând setul complet al criteriilor considerate (întrebările ce vor fi afișate pe parcursul sesiunii de consultare)

În urma consultării se generează fișierul KBASEX conținând recomandările rezultate. În continuare se prezintă conținutul primelor 3 fișiere:

Conținutul fișierului KBASED:

```
"DOCUMENTARE IN PROGRAME DE CONSTRUCTII","GOBESZ F. ZSONGOR"
"Programul expert DPC (Documentarea in Programele de Constructii) va ofera
posibilitatea de a utiliza programul disponibil cel mai adecvat (din biblioteca
de programe a I.P.C.T. Bucuresti, str. T. Arghezi nr. 21) pentru rezolvarea
prob- lemei Dvs. "
""
""
```

Conținutul fișierului KBASEA:

```
"Incercati programul ANELISE, care permite si analize "Time History" la cadrele
plane actionate static si dinamic in domeniul elastic sau inelastic. Autori:
D. Mucichescu, D. Capatina s.a., IPCT-SA, Bucuresti, str. T. Arghezi 21.",.01
1,.75,.01
3,.6,.01
```

4,.65,.01
5,.6,.01
6,.6,.01
7,.6,.01
8,.85,.01
9,.6,.01
10,.7,.01
13,.85,.01
14,.65,.01
15,.7,.01
19,.6,.01
20,.6,.01
22,.65,.01
23,.8,.01
25,.9,.01
26,.8,.01
27,.75,.01
30,.85,.01
33,.9,.01
999

"Va propunem programul CADRE, care face analiza statica si seismica (dupa P100) a cadrelor plane. Autori: D. Cocora, T. Cornea, V. Dinu si N. Balosache, IPCT-SA, Bucuresti, str. T. Arghezi 21, tel. 117850/170.",.01

1,.75,.01
3,.65,.01
4,.65,.01
5,.6,.01
6,.6,.01
7,.6,.01
8,.85,.01
9,.6,.01
10,.65,.01
11,.7,.01
12,.65,.01
13,.85,.01
14,.65,.01
15,.8,.01
17,.7,.01
19,.6,.01
20,.6,.01
22,.65,.01
27,.75,.01
999

"Programul CASE pare cel mai convenabil. El va permite analiza liniara spatiala a cadrelor cu diafragme si plansee actionate static/ dinamic/ seismic. Autori: D. Capatina, D. Cocora, T. Cornea, D. Giurgea, N. Balosache, IPCT-SA, Buc. Tel. 117850.",.01

1,.75,.01
3,.6,.01
4,.65,.01
5,.6,.01
6,.6,.01
7,.6,.01
8,.85,.01
9,.6,.01
10,.65,.01
11,.7,.01
12,.65,.01
13,.85,.01
14,.65,.01
15,.7,.01
16,.8,.01

17,.7,.01
18,.78,.01
19,.6,.01
22,.65,.01
27,.8,.01
28,.86,.01
999

"Cu programul DELTA puteti stabili relatii "P-Delta" prin analiza statica neli-
niara a structurilor din bare. Autori: D.Constantinescu, D. Giurgea, IPCT-SA,
Bucuresti, str. T. Arghezi 21, tel. 117850/170.",.01

1,.75,.01
3,.6,.01
4,.65,.01
5,.6,.01
6,.6,.01
8,.85,.01
9,.6,.01
19,.6,.01
26,.78,.01
33,.9,.01
36,.95,.01
999

"Cu programul GIPSI puteti analiza prin substructurare structuri extinse din ba-
re, actionate static sau seismic. Autori: T. Cornea, D. Cocora,D. Giurgea,IPCT-
SA, Bucuresti, str. T. Arghezi 21, tel. 117850/170.",.01

1,.75,.01
3,.6,.01
4,.65,.01
5,.6,.01
6,.6,.01
7,.6,.01
8,.85,.01
9,.6,.01
11,.7,.01
12,.65,.01
13,.85,.01
14,.65,.01
15,.75,.01
16,.8,.01
17,.7,.01
18,.78,.01
20,.6,.01
28,.85,.01
31,.9,.01
999

"Probabil GRIRUL va convine. El va permite determinarea eforturilor din grinzi-
le cailor de rulare. Autor: V. Dinu, IPCT-SA, Bucuresti, str. T. Arghezi 21, tel.
117850.",.01

1,.75,.01
3,.7,.01
4,.65,.01
5,.7,.01
6,.6,.01
7,.6,.01
8,.85,.01
9,.6,.01
11,.7,.01
12,.65,.01
28,.8,.01
999

"Cu programul RETGRI puteti analiza static retele de grinzi cu sau fara torsiu-
nea barelor. Autori: T. Cornea,D. Giurgea, IPCT-SA, Bucuresti, str. T. Arghezi

21, tel. 117850/170.",.01

1,.75,.01

3,.7,.01

4,.8,.01

5,.6,.01

6,.65,.01

7,.6,.01

8,.85,.01

9,.6,.01

11,.7,.01

12,.65,.01

20,.65,.01

22,.65,.01

999

"Va propunem programul SECON, care permite analiza statica si seismica (P100-91) a cadrelor plane. Autori: D. Capatina, D. Cocora, T. Cornea, N. Balosache, IPCT SA, Bucuresti, str. T. Arghezi 21, tel. 117850/170.",.01

1,.75,.01

3,.7,.01

4,.7,.01

5,.65,.01

6,.7,.01

7,.65,.01

8,.85,.01

9,.6,.01

10,.65,.01

11,.65,.01

12,.65,.01

13,.85,.01

14,.6,.01

15,.8,.01

17,.7,.01

19,.6,.01

22,.65,.01

999

"Programul SPAN (Space Analysis) pare convenabil. El permite analiza spatiala liniara a structurilor solicitate static sau dinamic. Autor: St. Betea, ICB-FCCIA Bucuresti, B-dul Lacul Tei 124, tel. 877030.",.01

1,.75,.01

3,.6,.01

5,.6,.01

6,.6,.01

7,.6,.01

9,.6,.01

10,.6,.01

11,.6,.01

12,.65,.01

13,.85,.01

14,.65,.01

15,.7,.01

16,.7,.01

17,.79,.01

18,.8,.01

19,.6,.01

22,.65,.01

23,.7,.01

24,.7,.01

27,.75,.01

29,.85,.01

31,.9,.01

999

"Programul SAP6 va permite analiza liniara prin MEF a structurilor oarecari so-

licitate static sau dinamic. Adaptarea dupa SAP6-Berkeley de D. Cretu, ICB, Bucuresti, B-dul Lacul Tei 124, tel. 877030/108.",.01

1,.75,.01
4,.65,.01
7,.6,.01
8,.85,.01
9,.6,.01
12,.6,.01
14,.6,.01
15,.7,.01
17,.7,.01
18,.8,.01
21,.6,.01
23,.8,.01
24,.85,.01
25,.9,.01
29,.85,.01
30,.8,.01
31,.9,.01
32,.75,.01
35,.9,.01
37,.95,.01
999

"Programul ADINA va permite analiza (liniara/ neliniara) prin MEF a structurilor oarecare solicitate static sau dinamic. Adaptat de D. Cretu ICB-FCCIA,Bucuresti B-dul Lacul Tei 124, tel. 877030/108.",.01

1,.75,.01
3,.6,.01
4,.65,.01
7,.6,.01
8,.85,.01
12,.65,.01
14,.65,.01
15,.75,.01
17,.7,.01
18,.8,.01
21,.6,.01
23,.75,.01
24,.85,.01
25,.9,.01
26,.78,.01
29,.85,.01
30,.8,.01
31,.9,.01
32,.75,.01
33,.9,.01
35,.9,.01
36,.95,.01
999

"Programul COSURI_BA permite calculul cosurilor de fum industriale din b.a., legile constitutive fiind furnizate de utilizator. Autori: V. Rosetnic, M. Munteanu, ICB, Catedra CBA, Bucuresti, B-dul Lacul Tei 124, tel. 877030.",.01

1,.75,.01
3,.6,.01
4,.65,.01
6,.6,.01
10,.65,.01
13,.85,.01
14,.65,.01
15,.7,.01
17,.7,.01
26,.8,.01

28,.8,.01

33,.9,.01

37,.8,.01

999

"Programul COSURI_ZID permite calculul cosurilor de fum industriale din zidarie.

Autori: V. Rosetnic, M. Munteanu, ICB, Catedra CBA, Bucuresti, B-dul Lacul Tei

124, tel. 877030.",.01

1,.75,.01

3,.6,.01

4,.65,.01

6,.6,.01

10,.65,.01

13,.85,.01

14,.65,.01

15,.7,.01

17,.7,.01

26,.78,.01

28,.8,.01

37,.8,.01

999

"Programul STRUC va permite calculul liniar al structurilor formate din bare.

Autori: V. Rosetnic, M. Munteanu, ICB, Catedra CBA, Bucuresti, B-dul Lacul Tei

124, tel. 877030.",.01

1,.75,.01

3,.6,.01

4,.65,.01

5,.6,.01

7,.6,.01

10,.65,.01

11,.65,.01

13,.85,.01

14,.65,.01

15,.7,.01

19,.6,.01

24,.8,.01

999

"Programul MICROSAP de tip MEF, permite analiza liniara statica, dinamica si a
campului de temperaturi a structurilor oarecari. Autori: D. Cocora (Carpati-Pro
iect, Buc. 153368), O. Coman (IRNE) si D. Cretu (ICB-FCCIA, Buc. tel.

877030)",.01

1,.75,.01

4,.65,.01

5,.6,.01

7,.6,.01

8,.85,.01

10,.7,.01

11,.7,.01

12,.65,.01

13,.85,.01

14,.65,.01

15,.75,.01

16,.8,.01

17,.7,.01

18,.78,.01

19,.6,.01

20,.6,.01

21,.6,.01

22,.65,.01

23,.8,.01

24,.8,.01

25,.85,.01

29,.85,.01

30,.8,.01
32,.75,.01
34,.88,.01
35,.9,.01
999

"Cu programul CADN puteti efectua analiza statica/ dinamica a cadrelor plane. Autor: Iosub Ioji, PRODOMUS-SA, Buc., str. N. Filipescu 53-55, tel. 117840/171.",.01

1,.75,.01
3,.6,.01
4,.65,.01
5,.6,.01
6,.6,.01
7,.7,.01
8,.85,.01
9,.6,.01
10,.6,.01
11,.65,.01
12,.65,.01
13,.85,.01
17,.7,.01
21,.6,.01
28,.8,.01
999

"Cu programul CPND puteti face analize statice/ dinamice la structurile plane din bare, prin MEF. Autor: Iosub Ioji, PRODOMUS-SA, Buc., str. N. Filipescu 53-55, tel. 117840/171.",.01

1,.75,.01
3,.6,.01
4,.65,.01
5,.65,.01
7,.6,.01
8,.85,.01
9,.6,.01
10,.65,.01
11,.65,.01
13,.85,.01
20,.6,.01
21,.6,.01
22,.65,.01
24,.75,.01
999

"Cu programul DIAT puteti analiza static si dinamic structuri in cadre, diafragme sau mixte. Autor: Iosub Ioji, PRODOMUS-SA, Buc., str. N. Filipescu 53-55, tel. 117840/171.",.01

1,.75,.01
3,.6,.01
6,.6,.01
7,.6,.01
8,.85,.01
9,.6,.01
10,.65,.01
13,.85,.01
14,.65,.01
15,.7,.01
17,.7,.01
18,.78,.01
21,.65,.01
22,.65,.01
23,.78,.01
27,.75,.01
999

"Cu programul EFORT puteti analiza static si dinamic cadre plane inclusiv dimensionarea elementelor (stalpi, grinzi de b.a.). Autor: V. Mager, Iosub Ioji, PRODOMUS-SA, Buc., str. N. Filipescu 53-55, tel. 117850/171.",.01

1,.75,.01
3,.6,.01
4,.65,.01
6,.65,.01
7,.6,.01
8,.85,.01
9,.6,.01
10,.65,.01
11,.65,.01
12,.65,.01
14,.65,.01
15,.7,.01
17,.7,.01
18,.78,.01
21,.6,.01
22,.65,.01
23,.75,.01
27,.7,.01
29,.85,.01
31,.9,.01
999

"Va propunem programul CYROM, care permite calculul spectrelor de nivel. Autori: V. Mustata, I. Santu, ISPE, Buc., tel. 107080/1712.",.01

1,.75,.01
3,.6,.01
5,.6,.01
8,.85,.01
9,.6,.01
10,.7,.01
11,.65,.01
16,.8,.01
29,.95,.01
999

"Va recomandam LAPROC, care permite analiza statica, analiza spectrala, combinatii ipoteze si dimensionare elemente de b.a. Autori: V. Mustata, A. Ungureanu, ISPE, Buc., tel. 107080/1712.",.01

1,.75,.01
3,.6,.01
5,.6,.01
7,.6,.01
11,.65,.01
12,.65,.01
14,.65,.01
16,.8,.01
17,.7,.01
18,.78,.01
20,.6,.01
21,.6,.01
22,.65,.01
23,.7,.01
24,.7,.01
29,.9,.01
32,.75,.01
999

"Va recomandam SUBSMD, care permite calculul static si dinamic aplicand tehnica substructurarii. Autori: G. Munteanu (Univ. Transilv. Brasov, tel. 172236), V. Mustata, R. Canarache, D. Costan (ISPE-Buc., tel. 107080/1712).",.01

1,.75,.01
3,.6,.01

4, .65, .01
5, .6, .01
7, .6, .01
11, .7, .01
12, .65, .01
14, .65, .01
15, .7, .01
16, .8, .01
17, .7, .01
18, .78, .01
20, .6, .01
21, .6, .01
23, .75, .01
24, .8, .01
31, .9, .01
32, .75, .01

999

"Cu programul CAD puteti efectua calculul static si seismic al cadrelor plane.

Autor: I. Olariu, IPROTIM-SA, Timisoara, str. Paris 2a, tel. 136311/173.", .01

1, .75, .01
3, .6, .01
4, .65, .01
5, .6, .01
6, .6, .01
7, .6, .01
10, .65, .01
11, .65, .01
12, .65, .01
13, .85, .01
14, .65, .01
15, .7, .01
17, .7, .01
22, .65, .01

999

"Cu programul GRI puteti efectua calculul static al grinzilor continue. Interfa-
ta moderna. Autor: I. Olariu, IPROTIM-SA, Timisoara, str. Paris 2a, tel.136311/
173.", .01

1, .75, .01
3, .6, .01
4, .65, .01
5, .6, .01
6, .6, .01
7, .7, .01
10, .65, .01
11, .7, .01
12, .65, .01
13, .85, .01
19, .6, .01

999

"Cu programul PLACA puteti face calculul static si dimensionarea placilor de be-
ton armat incovoiate cu MEF. Autori: V. Stoian, R. Friedrich (INCERC-Timis.),
D. Giurgea, E. Pecol si I. Olariu.", .01

1, .75, .01
4, .7, .01
5, .6, .01
7, .65, .01
8, .85, .01
10, .65, .01
11, .65, .01
12, .65, .01
13, .85, .01
19, .65, .01

21,.6,.01
23,.85,.01
24,.8,.01
34,.88,.01
999

"Cu programul CPF puteti determina caracteristicile dinamice ale cladirilor inguste si lungi cu 1-3 nivele, avand plansee flexibile. Autori: F. Gobesz, I. Bologna, UTC-N, Cluj-N., str. C. Daicoviciu 15, tel. 194816/146.",.01

1,.75,.01
3,.6,.01
4,.65,.01
5,.6,.01
8,.85,.01
10,.65,.01
14,.9,.01
999

"Va propunem programul EFEX pentru calculul static al grinzilor continue actionate de convoaie de sarcini mobile. Autor: F. Gobesz, UTC-N, Cluj-N., str. C. Daicoviciu 15, tel. 194816/146.",.01

1,.75,.01
3,.6,.01
4,.65,.01
5,.6,.01
6,.6,.01
7,.65,.01
8,.85,.01
10,.65,.01
11,.7,.01
999

"Cu INTERCON puteti analiza interactiunea neliniara teren-constructie pe modele plane MEF in cazul unei acceleratii date la nivelul rocei de baza. Autori: F. Gobesz (UTC-N), D. Capatina (IPCT-SA), D. Cocora (Carpati-Proiect, Buc.).",.01

1,.75,.01
4,.65,.01
5,.6,.01
9,.6,.01
16,.8,.01
21,.6,.01
22,.7,.01
23,.75,.01
24,.8,.01
25,.8,.01
30,.95,.01
33,.9,.01
999

"Cu interpretorul SICAM puteti rezolva secvente de operatii matriceale si analize statice/ dinamice la structuri spatiale din bare. Autor: F. Gobesz, UTC-N, Cluj-N., str. C. Daicoviciu 15, tel. 194816/146.",.01

1,.75,.01
3,.65,.01
5,.65,.01
7,.65,.01
9,.7,.01
14,.65,.01
16,.8,.01
18,.8,.01
25,.9,.01
999

"Cu interpretorul SIC puteti rezolva secvente de operatii matriceale si analize statice/ dinamice la structuri spatiale din bare. Autor: F. Gobesz, UTC-N, Cluj-Napoca, str. C. Daicoviciu 15, tel. 194816/146.",.01

1,.75,.01

3,.7,.01
5,.6,.01
6,.6,.01
7,.7,.01
8,.85,.01
9,.65,.01
10,.7,.01
14,.7,.01
18,.8,.01
25,.9,.01
999

"Cu interpretorul SIP puteti rezolva secvente de operatii matriceale si analiza static/ dinamic structuri plane din diverse tipuri (10) de bare. Autor: F. Gobesz, UTC-N, Cluj-N., str. C. Daicoviciu 15, tel. 194816/146.",.01

1,.75,.01
3,.65,.01
4,.75,.01
5,.6,.01
6,.6,.01
7,.65,.01
8,.85,.01
9,.7,.01
10,.65,.01
14,.65,.01
22,.7,.01
25,.9,.01
27,.75,.01
28,.85,.01
34,.88,.01
999

"Cu interpretorul SICAM-PC puteti rezolva secvente de operatii matriceale si analiza static/ dinamic structuri plane din diverse (10) tipuri de bare. Autor: F. Gobesz, UTC-N, Cluj-N., str. C. Daicoviciu 15, tel. 194816/146.",.01

1,.75,.01
3,.65,.01
4,.7,.01
5,.6,.01
6,.6,.01
7,.7,.01
9,.65,.01
10,.7,.01
13,.85,.01
14,.7,.01
22,.7,.01
25,.9,.01
27,.75,.01
28,.85,.01
34,.9,.01
999

"Cu START1 puteti face analiza statica a structurilor articulate (plane, spatiale pretensionate). Autor: F. Gobesz, UTC-N, Cluj-N., str. C. Daicoviciu 15, tel 194816/146.",.01

1,.75,.01
3,.65,.01
4,.65,.01
5,.65,.01
6,.6,.01
7,.7,.01
8,.85,.01
9,.65,.01
16,.8,.01
18,.78,.01

999

"Cu START2 puteti efectua analize statice neliniare (de ordinul 2) la structuri articulate (plane, spatiale pretensionate). Autor: F. Gobesz, UTC-N, Cluj-N., str. C. Daicoviciu 15, tel. 194816/146.",.01

3,.7,.01

4,.65,.01

5,.7,.01

6,.65,.01

8,.85,.01

9,.7,.01

10,.6,.01

16,.8,.01

18,.78,.01

33,.9,.01

999

"Cu programul START3 puteti efectua analiza statica neliniara (de ordinul 3) la structurile articulate (plane, spatiale pretensionate). Autor: F. Gobesz, UTC-N Cluj-N., str. C. Daicoviciu 15, tel. 194816/146.",.01

1,.75,.01

3,.65,.01

4,.65,.01

5,.65,.01

6,.6,.01

9,.65,.01

16,.8,.01

18,.78,.01

26,.9,.01

999

"Cu programul START4 puteti efectua analize statice neliniare (de ordinul 3) la structuri articulate (spatiale pretensionate) incluzand efecte de membrana. Autor: F. Gobesz, UTC-N, Cluj-N., str. C. Daicoviciu 15, tel. 194816/146.",.01

1,.75,.01

3,.65,.01

4,.65,.01

5,.65,.01

9,.7,.01

16,.8,.01

18,.78,.01

23,.75,.01

24,.8,.01

26,.8,.01

999

"Va propunem CADRE10 pentru analiza statica liniara a structurilor din bare, legate elastic in noduri de dimensiuni finite. Autor: M. Petrina, UTC-N, Cluj-N., str. C. Daicoviciu 15, tel. 194816/155.",.01

1,.75,.01

3,.65,.01

4,.7,.01

6,.6,.01

7,.6,.01

8,.85,.01

10,.65,.01

11,.7,.01

12,.65,.01

13,.85,.01

15,.7,.01

16,.8,.01

20,.6,.01

27,.8,.01

28,.8,.01

999

"Cu programul LINIINFL1 puteti determina linii de influenta la grinzi continue

si cadre. Autor: M. Petrina, UTC-N, Cluj-N., str. C. Daicoviciu 15, tel.194816/155.",.01
1,.75,.01
3,.65,.01
4,.65,.01
5,.6,.01
6,.6,.01
7,.65,.01
8,.85,.01
16,.8,.01
19,.7,.01
20,.65,.01
999

"Cu programul LINIINFL2 puteti determina linii de influenta la retele de grinzi. Autor: M. Petrina, UTC-N, Cluj-N., str. C. Daicoviciu, tel. 194816/155.",.01
1,.75,.01
3,.6,.01
4,.65,.01
5,.6,.01
6,.6,.01
7,.65,.01
8,.85,.01
16,.8,.01
20,.65,.01
999

"Cu programul STRART puteti calcula static structuri articulate plane si spatiale de mari dimensiuni. Autor: M. Petrina, UTC-N, Cluj-N., str. C. Daicoviciu 15 tel. 194816/155.",.01
1,.75,.01
3,.65,.01
4,.6,.01
5,.6,.01
6,.65,.01
7,.65,.01
8,.85,.01
9,.65,.01
18,.78,.01
999

"Va recomandam programul DIAF pentru calculul static al diafragmelor la sarcini verticale si orizontale (vant, seism P100). Autor: I. Moga, UTC-N, Cluj-N.,str. C. Daicoviciu 15, tel. 194816/107.",.01
1,.75,.01
3,.6,.01
4,.65,.01
5,.6,.01
6,.6,.01
8,.85,.01
9,.6,.01
10,.7,.01
11,.65,.01
12,.65,.01
13,.85,.01
15,.65,.01
16,.8,.01
17,.7,.01
19,.7,.01
20,.6,.01
21,.65,.01
999

"Cu programul GME puteti calcula static grinzi pe mediu elastic (model Winkler). Dispune de pre si post procesare grafica. Autor: I. Moga, UTC-N, Cluj-N., str. C. Daicoviciu 15, tel. 194816/107.",.01

1,.75,.01
 3,.6,.01
 4,.65,.01
 5,.6,.01
 6,.6,.01
 8,.85,.01
 9,.6,.01
 10,.65,.01
 11,.65,.01
 12,.65,.01
 13,.85,.01
 16,.8,.01
 19,.8,.01
 20,.65,.01
 34,.95,.01
 999

"Programul STRMIX permite calculul static si la seism (dupa P100-91) a structurilor mixte (cadre-diafragme). Autor: I. Moga, UTC-N, Cluj-N., str. C. Daicoviciu 15, tel 194816/107.",.01

1,.75,.01
 3,.6,.01
 4,.65,.01
 5,.6,.01
 6,.6,.01
 8,.85,.01
 9,.6,.01
 10,.65,.01
 11,.65,.01
 12,.65,.01
 13,.85,.01
 15,.7,.01
 16,.8,.01
 17,.7,.01
 19,.65,.01
 20,.6,.01
 21,.65,.01
 999

"Va propunem programul PFRAME al firmei Softek din Vancouver, Canada. Puteti analiza (static, dinamic, biografic etc) structuri plane complexe din bare si diafragme. Dispune de pre si post procesare grafica avansata (animatie etc).",.01

1,.75,.01
 2,.65,.01
 3,.6,.01
 4,.7,.01
 7,.6,.01
 10,.65,.01
 11,.65,.01
 12,.65,.01
 13,.85,.01
 14,.65,.01
 17,.7,.01
 19,.7,.01
 20,.6,.01
 21,.65,.01
 22,.65,.01
 25,.8,.01
 26,.8,.01
 29,.85,.01
 32,.75,.01
 36,.95,.01
 37,.95,.01
 71,.95,.01

999

"Va recomandam IMAGES-3D al firmei Celestial Software (de tip MEF), care permite analiza complexa (statica, dinamica, seismica) a structurilor spatiale formate din bare si elemente bi si tri-dimensionale. Dispune de pre si postprocesare moderna.",.01

1,.75,.01

2,.65,.01

3,.6,.01

4,.65,.01

7,.6,.01

10,.65,.01

11,.65,.01

12,.65,.01

13,.85,.01

14,.7,.01

17,.7,.01

18,.8,.01

19,.65,.01

20,.6,.01

21,.65,.01

22,.65,.01

23,.7,.01

24,.8,.01

25,.8,.01

26,.78,.01

29,.85,.01

32,.75,.01

72,.95,.01

999

"Cu SAP90 (varianta SAP pentru PC E.L. Wilson, Berkeley, Calif.)puteti efectua analize complexe (structuri si actiuni oarecari) liniare si neliniare. Dispune de pre si postprocesare grafica avansata. Permite si verificari de tensiuni.",.01

1,.75,.01

2,.65,.01

4,.6,.01

7,.65,.01

10,.65,.01

11,.65,.01

12,.65,.01

13,.85,.01

14,.65,.01

17,.7,.01

18,.8,.01

19,.65,.01

20,.6,.01

21,.6,.01

22,.65,.01

23,.65,.01

24,.82,.01

25,.75,.01

26,.78,.01

27,.7,.01

29,.85,.01

32,.75,.01

33,.9,.01

35,.9,.01

37,.9,.01

73,.95,.01

999

Conținutul fișierului KBASEQ:

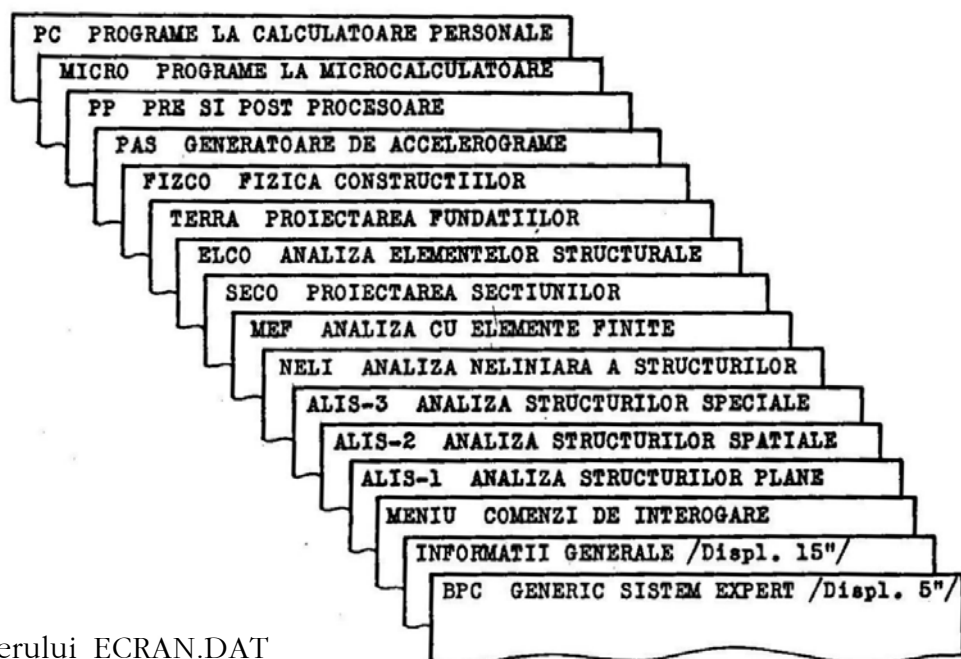
Doriti sa efectuati o analiza structurala (statica/ dinamica/ seismica)?
Doriti o analiza de sectiune/element (problema e de dimensionare/verificare) ?
Structura de analizat e alcatuita numai din elemente liniare (1D) ?
Doriti sa analizati o structura plana (eventual retea) ?
Nodurile structurii sunt considerate ideale (punctuale) ?
Preferati un program care rezolva numai structuri din bare ?
Doriti o analiza statica liniara ?
Doriti ca rulara programului sa se faca pe minicalculatoare ?
Incarcarile sunt aplicate (reduse) numai pe noduri ?
Doriti o introducere interactiva (preprocesata) a datelor ?
Exista si incarcari standard (P si q) in campul barelor ?
Doriti ca programul sa faca si combinatii de ipoteze statice ?
Doriti ca programul sa fie rulat pe PC-uri (compatibile IBM) ?
Doriti o analiza dinamica modala ?
Doriti o analiza seismica (dupa cod P-100/91) ?
Doriti rulara programului pe sisteme de calcul FELIX ?
Doriti si combinatii de ipoteze statice cu dinamice ?
Structura analizata este spatiala (3D) ?
Doriti o postprocesare (eventual grafica) a rezultatelor ?
Structura analizata are si reazeme elastice ?
Doriti ca programul sa genereze automat noduri/elemente ?
Doriti sa surprindeti si efectul forfecarii in analiza ?
Structura are in alcatuire si elemente de suprafata (2D) ?
Acceptati pentru analiza si un program mai general tip MEF ?
Doriti o analiza dinamica de tip "Time History" ?
Doriti un calcul geometric neliniar (de ordinul 3) ?
Unele noduri ale structurii se considera cu dimensiuni finite?
Exista in structura si bare cu I variabil (cu vute) ?
Doriti o analiza dinamica spectrala ?
Doriti si surprinderea interactiunii teren-structura ?
Doriti facilitatea analizei prin substructurare ?
Doriti si renumerotarea automata optima a nodurilor ?
Doriti un calcul neliniar de ordinul 2 (fizic) ?
Aveti in structura si bare rezemate pe mediu elastic ?
Structura are in alcatuire si elemente masive (3D) ?
Doriti o analiza statica biografica (cu P crescator) ?
Doriti si verificari la stabilitate ale structurii ?
Elementul analizat este din beton armat ?
Doriti stabilirea capacitatii portante a elementului de b.a. ?
Sectiunea elementului analizat e formata din dreptunghiuri ?
Elementul analizat e de tip "stalp" ?
Doriti introducerea interactiva a datelor pentru analiza ?
Doriti ca legea "sigma-epsilon" (curba caracteristica a materialului) sa fie impusa prin program ?
Doriti si calculul ariilor de armatura ?
Doriti rulara programului pe PC-uri (compatibile IBM) ?
Doriti rulara programului pe minicalculatoare ?
Elementul analizat este de tip "grinda" ?
Doriti ca legea "sigma-epsilon" (curba caracteristica a materialului) sa fie furnizata de utilizator ?
Elementul analizat este de tip "perete-diafragma" ?
Doriti dimensionarea sectiunilor de beton armat ?
Doriti si calculul distributiei deformatiilor specifice din sectiune ?
Doriti prin program desene interactive + calcule ?
Doriti si calculul unor curbe de interactiune (N-M) ?
Doriti trasarea prin program a curbelor calculate ?
Sectiunea elementului analizat are o forma oarecare ?
Doriti si trasarea unor diagrame de eforturi si deplasari ?
Doriti si calculul diagramelor "moment-curbura" ?

Doriti prin program si realizarea unor planse de executie ?
Doriti rularea programului pe calculatoare sistem FELIX ?
Elementul analizat este metalic (din otel) ?
Doriti prin program si optimizarea sectiunii ?
Doriti si calculul diagramelor "moment-rotire" ?
Doriti si calculul curbelor de interactiune (N-Mx-My) ?
Elementul analizat est de tip placa (planseu) ?
Elementul analizat este din beton armat precomprimat ?
Doriti alegerea sectiunii metalice (dimensionarea) prin program ?
Doriti calculul capacitatii portante a sectiunii/elementului metalic analizat ?
Doriti si verificari la stabilitate locala a elementului metalic analizat ?
Elementul analizat este de tip "grinda perete" ?
Doriti prin program si calculul solidarizarilor (suruburi, nituri) la elementele metalice analizate ?
Doriti o analiza complexa la PC a unei structuri liniare, plane, cu verificarea tensiunilor din elemente ?
Doriti o analiza complexa la PC a unei structuri liniare, spatiale, cu verificarea tensiunilor din elemente ?
Doriti o analiza complexa la PC, a unei structuri oarecari cu verificarea tensiunilor din elemente ?

B. Completări la sistemul expert BPC scris în Fortran/2 IBM

1. Structura generală a fișierului INDEX.DAT (GEN.DAT), numărul înregistrărilor fiind cu unu mai mare decât numărul programelor catalogate în BPC:

- poz. 1 1 col. - liber;
- poz. 2 1 col. - clasa (categoria) programului;
- poz. 3 1 col. - subclasa programului;
- poz. 4 1 col. - pachetul în care e catalogat programul;
- poz. 5 - 12 8 col. - denumirea programului
- poz. 13 - 16 4 col. - autor(ii) programului (de la 1 la maxim 4);
- poz. 17 - 18 2 col. - anul elaborării/revizuirii programului
- poz. 19 - 20 2 col. - limbajul/structura programului;
- poz. 21 - 22 2 col. - implementări ale programului;
- poz. 23 - 24 2 col. - starea și documentarea programului;
- poz. 25 - 60 26 col. - domeniul de aplicabilitate, din care:
 - 4 col. - tip (model) structura;
 - 4 col. - tip (model) elemente;
 - 4 col. - tip secțiuni;
 - 4 col. - tip (model) legături;
 - 4 col. - tip acțiuni (încărcări);
 - 4 col. - tip material;
 - 4 col. - tip (metodă) analiză;
 - 4 col. - ipoteze acceptate în model;
 - 4 col. - norme acceptate, opțiuni, calcule speciale;
- poz. 61 - 68 8 col. - performanțe, restricții, din care
 - 4 col. - gabarite aplicații, limite, timpi de rulare;
 - 4 col. - rezultate și informații furnizate de program;
- poz. 69 - 78 12 col. - interfața cu utilizatorul, din care:
 - 2 col. - modul de operare;
 - 4 col. - structura datelor de intrare;
 - 4 col. - pre/post-procesare, I/O grafic etc.
- poz. 79 - 80 2 col. - observații + număr de ordine program.



2. Specificațiile fișierului ECRAN.DAT

3. Conținutul fișierului ECRAN.DAT:

M.I.S. Bucuresti * UNIV. TEHNICA Cluj-Napoca * I.P.C.T.-S.A. Bucuresti

V A O F E R A S I S T E M U L E X P E R T

BBBBBBBBBBBBBB	PPPPPPPPPPPPPP	CCCCCCCC
BBBBBBBBBBBBBB	PPPPPPPPPPPPPPPP	CCCCCCCCCCCCCCCC
BBBBBBBBBBBBBB	PPPPPPPPPPPPPPPP	CCCCCCCCCCCCCCCC
BBBBB BBBB	PPPPP PPPPP	CCCCC CCCCC
BBBBB BBBB	PPPPP PPPPP	CCCCC CCCCC
BBBBB BBBB	PPPPP PPPPP	CCCCC
BBBBBBBBBBBBBB	PPPPPPPPPPPPPPPP	CCCCC
BBBBBBBBBBBBBB	PPPPPPPPPPPPPPPP	CCCCC
BBBBBBBBBBBBBB	PPPPPPPPPPPPPP	CCCCC
BBBBB BBBB	PPPPP	CCCCC
BBBBB BBBB	PPPPP	CCCCC CCCCC
BBBBB BBBB	PPPPP	CCCCC CCCCC
BBBBBBBBBBBBBB	PPPPP	CCCCCCCCCCCCCCCC
BBBBBBBBBBBBBB	PPPPP	CCCCCCCCCCCCCCCC
BBBBBBBBBBBBBB	PPPPP	CCCCCCCC

pentru consultarea "BIBLIOTECII DE PROGRAME DE CALCUL" in Constructii

AUTORI: Prof.dr.ing. GOBESZ FERDINAND, UTCluj-N.Str.C.Daicoviciu 15, tel.134565
ing. GOBESZ F.ZSONGOR, UTCluj-N.Str.C.Daicoviciu 15, tel.134565

P R E Z E N T A R E

DENUMIRE : BPC (Biblioteca de Programe in Constructii).

OBIECT : Produsul informatic conversational BPC permite utilizatorilor sa consulte documentatiile de prezentare (fisele tehnice ale manualelor de utilizare) referitoare la pachetele de programe pentru cercetarea si proiectarea constructiilor descrise in "CATALOGUL PROGRAMELOR DE CALCUL" IPCT-ICCPDC Bucuresti, editia 1988.

DIALOG : Conversational pe baza de MENIU avind o gama larga de facilitati.

DOMENIU : Limitat in versiunea actuala la programele gestionate de IPCT-SA. Exista posibilitatea extinderii (actualizarii) sistemului *BPC* .

OUTPUT : Oferă raspunsuri (construite) la cereri variate de tipul:
- tabele sinoptice pentru pachetele de programe comune;
- fise tehnice complete sau parțiale;
- explicatii termeni tehnici pe baza unui GLOSAR;
- informatii de sinteza (inclusiv posibilitati de listare) etc.

RELATII : Dr.ing. F.GOBESZ, UNIV. TEHNICA Cluj-Napoca, tel. (95)134565-146
Ing. Zs.F.GOBESZ, UNIV. TEHNICA Cluj-Napoca, tel. (95)134565-112
** PENTRU AFISARE MENIU TASTATI <CR> **

M E N I U P R I N C I P A L

MENIUL programului "BPC" accepta următoarele forme de interogare si comenzi:

?termen <CR> se solicita explicarea "termen"-ului din GLOSAR.
 <CR> sau _<CR> se solicita afisarea MENIU-lui.
 \$ <CR> Interogare libera, dirijata pe baza de submeniuri. Sistemul BPC
 construiește raspunsul dupa interpretarea cererilor.
 !id.fis. <CR> se deschide fisierul specificat pentru retinerea dialogului.
 & <CR> se renunta la fisierul pentru salvarea dialogului.
 ^ <CR> se reia executia comenzii precedente (valabil doar odata!).
 + <CR> se afiseaza urmatorul tabel sinoptic de programe (ciclic).
 - <CR> se afiseaza tabelul sinoptic precedent de programe (ciclic).

#termen <CR> Comenzi speciale:
 #ALIS01 <CR> se va afisa tabelul sinoptic solicitat (ALIS-01) etc.
 #PAUSE <CR> se suspenda temporar executia (si se inchid fisierele).
 #STOP <CR> terminarea normala a executiei (si inchiderea fisierelelor).
 CTRL/S, SCROLL opreste defilarea afisarii pe ecran.
 CTRL/Q, SCROLL se reia defilarea afisarii pe ecran.

NOTA: In cazul unei cereri eronate va fi reafisat MENIUL PRINCIPAL !

I-1	ALIS-01	Analiza Liniara plana a Structurilor din bare.					
		AUTOR			A C T I U N E A	SISTEM	
COD	DENUMIRE	INSTI- TUTIE	CLS.	STRUCTURA	STATICA FIX.MOB.	DINAMICA A.M.P100	FE- MI- OBSERV. LIX NI
101	GRINDA	IPCT	A4	Grinda continua	*		* +Med.el.
102	GRILIN	IPCT	A4	Grinda continua		*	*
103	GRIRUL	IPROMET	A4	Grinda continua		*	*
104	ANGI	IPCT	A1	Grinzi de rulare		*	*
105	EFEX	IPCN	B5	Grinda continua		*	*
106	GREF	IPCT	A4	Gr.simplu rezem.		*	*
107	LICA	IPCN	B6	Cadru plan		*	*
108	CAPLAN	IPCT	A4	Cadru plan	*	*	* *
109	SECON	IPCT	A2	Cadru plan	*	*	* * O/grafic
110	CADRE	IPCN	A4	Cadru plan	*	*	* * +leg.el.
111	SECPAN	IPCT	A5	Cadru plan	*	*	* * +zidarii
112	SEDAN	IPCT	B2	Cadru plan	*	*	* * LISP+F77
113	SEGRAF	IPCT	A1	Cadru plan	*	*	* * O/grafic
114	RETGRI	IPCT	A2	Retea de grinzi	*		* * +Med.el.
115	SIP	IPCN	A2	Orice tip	*	*	* * L. SICAM

I-2	ALIS-02	Analiza Liniara Spatiala a Structurilor din bare.					
		AUTOR			A C T I U N E A	SISTEM	
COD	DENUMIRE	INSTI- TUTIE	CLS.	STRUCTURA	STA D I N A M I C A TIC AM. P100 SP. TH	FE- MI- OBSERV. LIX NI	
201	GIPSI	IPCT	A2	Orice tip	* * *	* * O/graf.Y	
202	SICAM	IPCN	B5	Orice tip	* *	* L. SICAM	
203	SIC	IPCN	B5	Orice tip	* *	* L. SICAM	
204	CASPA	IPCT	A4	Cadre spatiale	*	*	
205	SPAT	IPCT	A4	Orice tip	* * * * *	* O/graf.Y	
206	HALCONS	IPCT	A4	Hale cu pod rul.*	* * *	* +Dimens.	
207	CASE	IPCT	A2	Cadre/diafragme	* * * * *	* Dim.stilp	
208	CPF	IPCN	B5	Cu plansee flex.*	* *	*	
209	HALES	IPCT	B1	Hale ind.parter	* *	* Mediu XRL	
210	LAPROC	ISPE	C6	Cadre din b.a.	* * *	Dim.stilp	
211	START1	IPCN	B5	Cu zabrele	*	*	

I-3	ALIS-03	Analiza Liniara Spatiala a Structurilor de tip special.						
		AUTOR	A C T I U N E A SISTEM					
COD	DENUMIRE	INSTI- TUTIE	CLS.	STRUCTURA	STA- TICA	DINAMICA A.M. P100	FE- LIX	MI- NI OBSERV.
301	NELSAS	IPCN	B5	Spatiala cu zabrele	*		*	Gen.date
302	TREZE	IPCT	A5	Rezerv. cilindrice	*		*	
303	CASAP	IPCT	A1	Castele de apa	*	*	*	+Dimens.
304	FUNCAS	IPCT	A1	Fund. ptr. cast.apa	*		*	+Dimens.
305	SILOTURN	INCERC	A5	Silozuri	*	*	*	
306	SMASH	COCC	C6	Pereti mulati	*		*	
307	MTUB	INCERC	C6	C-tii cu nucl. port.	*	*	*	

I-4	NELI	Analiza NELIniara a structurilor din bare sau asimilabile.						
		AUTOR	A C T I U N E A SISTEM					
COD	DENUMIRE	INSTI- TUTIE	CLS.	STRUCTURA	STATICA CT CR CD	DINAM. P100 TH	FE- LIX	MI- NI OBSERV.
401	START2	IPCN	B5	Articul.,cabluri	*		*	
402	START4	IPCN	B5	Cabluri/membrane	*		*	Membrana
403	UNIFE	IPCN	A4	Articul.,cabluri	*		*	
404	THACS	IPCN	B5	Articul.,cabluri		*	*	
405	NELIN	ICCPDCT	B5	Articulate	*	(*)	*	Stabilit
406	STAGRUL	ICCPDCT	B5	Grinzi de rulare		(*)	*	Stabilit
407	CRISTIAN	IPT	B5	Hale cu pod rul.		(*)	*	Stabilit
408	DELTA	IPCT	A2	Cadre/diafragme	*	*	*	Din b.a.
409	CANEL	IPCT	A4	Cadre/diafragme	*	*	*	Din b.a.
410	ABIS	IPCT	A5	Cadre	*	*	*	
411	CASEP	ICB	B5	Cadre	*	*	*	
412	STASEL	ICB	B5	Cadre		(*)	*	Stabilit
413	THNEL	INCERC	B5	Str.tip forfecat		*	*	*
414	ANELISE	IPCT	A2	Cadre	*	*	*	O/grafic
415	START3	IPCN	B5	Articul.,cabluri	*			

LEGENDA: (*) Cedarea prin pierderea stabilitatii.

I-5	MEF	Analiza Structurilor cu Metoda Elementelor Finite (de Frontiera)						
		AUTOR	A C T I U N E A SISTEM					
COD	DENUMIRE	INSTI- TUTIE	CLS.	S T R U C T U R A	STATI. CT CR	DINAMICA AM SP TH	FE- LIX	MI- NI OBSERV.
501	SAP05	IPCT	A2	Orice tip	*	*	*	*
502	SAP05M	ICB	A2	Orice tip	*	*	*	*
503	MINSAP	IPCT	A2	Orice tip	*	*	*	*
504	PALES	ICIMBv	A4	Placi,diafrag.,tuburi	*		*	*
505	BIOGRAF	IPT	A4	Diafragme plane	*	*	*	*
506	BIOSPAT	IPT	B5	Diafragme spatiale	*	*	*	*
507	PANEL	IPT	B5	Diafragme plane	*		*	*
508	FISFIN	IPT	B5	Placi plane / cutate	*		*	*
509	KYO	IPI	B5	Structuri plane	*		*	*
510	INTERCON	IPCN	B5	Structuri plane	*		*	*
511	SIAS314	INCERC	C6	Structuri din bare	*	*	*	*
512	SIAS262	INCERC	C6	Placi curbe de rotatie		*	*	*
513	SAPROM	ISPE	C6	Orice tip	*	*	*	*

I-6	SECO	Analiza Sectiunilor elementelor de Constructii													
		AUTOR	SECT.	MA-	SOLI-	D ANALIZA SEC.			SISTEM						
COD	DENUMIRE	INSTI-	CLS.	D D A	TE-	CI-	I RE-	RI-	DUC-	FE-	MI-	OBSERV.			
		TUTIE		R C R	RIAL	TARE	M ZIS.	GID.	TIL.	LIX	NI				
601	MODIAF	IPCT	A4	* *	BA	N, M		*		*	*	Mpl, Mul.			
602	COBLIK	IPCT	A4	*	BA	N, Mx, My		*			*	CI.Mx-My			
603	ANCEXO	ICB	A2	*	BA	N, Mx, My	*	*			*	Dif.norm			
604	JUMBO	ICB	A1	*	BA	N, Mx, My	*	*			*	ST.10107			
605	DEFBAR	ICB	A2	* *	BA	N, M		*	*	*	*	O/grafic			
606	DEFINE	IPCT	A2		* BA	N, M		*	*	*	*	O/grafic			
607	COBRA	IPCT	A4	* * *	BA	N, Mx, My	*	*			*				
608	ARDIAF	IPCT	A4	* *	BA	N, M, T	*	*			*				
609	SET01	IPCarp.	c6		* BA (R)	N, Mx, My		*	*	*	*				
610	GG05	COCC	A4		* BA (R)	N, M		*	*	*	*	O/grafic			
611	MORSEP	IPCT	A5		* BA/BP	N, M		*	*	*	*	CI.M-rot			
612	DIA	IPCN	C5	*	BA/BP	N, M		*	*	*	*				
613	DEVAST	ICB	C6		BA	toate	*	*	*		*	Dim. Aa			
614	VERIN	IPCT	A1	*	BA			*			*	Verf.nod			

I-7	ELCO	Analiza Elementelor de Constructie.																	
		AUTOR		TIP ELEMENT							SECT.	MA-	SOLI-	TIP ANALIZA			SISTEM		
COD	DENUMIRE	INSTI- TUTIE	CLS.	S	D	P	P	G	F	D	D	A	TE- RIAL	CI- TARE	RE- ZIS.	RA- GID.	DUC- TIL.	FE- LIX	MI- NI
604	JUMBO	IPCT	A1	*						*			BA	N,Mx,My,T	*		*	*	*
605	DEFBAR	IPCT	A2	*	*			*		*	*		BA	N,M	*	*	*		*
606	DEFINE	IPCT	A2	*							*		BA	N,M	*	*	*		*
608	ARDIAF	IPCT	A4	*						*			BA	N,M	*				*
701	PLOBA	IPT	A4				*			*			BA	M	*	*		*	
702	PLAGRI	IPT	C5				*	*		*			BA	M	*	*		*	
703	PREST	IPCT	B5				*				*		BP-T	N,M	*				*
704	BEAM	COCC	C6				*			*	*		BA	M	*			*	*
705	GG11,21	COCC	C6	*	*		*				*		BA-R	N,M	*	*		*	*
706	AJURATE	CCPHun	B5				*				*		OL	M	*			*	
707	OPTICON	IPI	B5				*				*		BP-T	M	*			*	
708	OPTICRUD	IPI	B5				*				*		BP-T	M	*			*	
709	CARP	IPCT	B5				*			*			BA	N,M	*				*
710	ATHENA	IPT	B1					*		*			OL	N,M	*	*		*	
711	STILP	IPROMET	C6	*				*			*		OL	N,Mx,My	*			*	
712	GRIPIC	IPCN	B5				*				*		BP-C	M,T	*			*	*
713	STIGOL	IPCT	B2	*						*			BA	N,M		*	*		*

I-8	TERRA	Calcul de geotehnica si fundatii									
		AUTOR		TIP		ACTIUNE	DIMEN-	CALCUL	SISTEM		
COD	DENUMIRE	INSTI-	CLS.	DE	STA-	DINA-	SIONA-	PRESI-	FE-	MI-	OBSERV.
		TUTIE		FUNDATIE	TICA	MICA	RE	UNI	LIX	NI	
801	FUNDA	IPCT		izolata	*		*	*	*	*	
802	STFMAT	ICB	A4	radier rig.	*			*	*		
803	RADIER	ICB	C6	radier rig.	*			*	*	*	Crbe M-Q
804	FLXMAT	ICB	A4	rad.flexib.	*			*	*		
805	MASIV	IIP TIC	A5	masiva		*		*	*		
806	FUMAS	IIP TIC	A5	cadru		*		*	*		

I-9	FIZCO	Calcul de FIZica Constructiilor									
		AUTOR		TIP ANALIZA	CIMP TERM.		REGIM TERMIC	CALC	SISTEM		
COD	DENUMIRE	INSTI- TUTIE	CLS.	LINI- ARA	NELI- NIARA	PLAN SPA- TIAL	STATI- ONAR	NESTA- TION.	CON- DENS	FE- MI LIX NI	
901	CIMPPLAN	IPCN	A1	*		*	*			* *	
902	CIMPSPAT	IPCN	A1	*		*	*			* *	
903	RENESTP	IPCN	B5	*		*		*		*	
904	CIMPNELP	IPCN	B5		*	*		*		*	
905	CIMPNELS	IPCN	C5		*	*		*		*	
906	SPATERM	IPI	B5	*		*	*			*	
907	CONDENS	IPI	B5	*		*	*		*	*	
908	PLUNUP	IPI	C5	*				*	*	*	
909	SPASCOND	IPI	A4	*		*	*		*	*	

I-10	PAS	Prelucrarea Accelerogramelor Seismice									
COD	DENUMIRE	AUTOR	INSTI- CLS.	GENE- RARE	ACTU- ALI- ZARE	PRE- LUC- RARE	SISTEM FE- MI- LIX NI	O B S E R V A T I I			
1001	SERVAC	IPCT	A2		*						
1002	SISGEN	INCERC	B5	*							
1003	SPEL	IPCT	A5			*					
1004	SPETAJ	INCERC	B5			*					
1005	PRES	IPCT	B1			*					
1006	SPECTRE	INCERC	C6			*					

Spectre medii (cu asig.)

I-11	PP	Pre si Postprocesoare.									
COD	DENUMIRE	AUTOR	INSTI- CLS.	PROCESA- LA	T DAF	E DAF	R MD	M DIGIGR.	I KOVO	N SCAMP	A L 2030
1101	PCASE	IPCT	A1	CASE	*						
1102	IGREC	CTCECN	A2	GIPSI, SAP	*	*	*	*	*		+MINSAP
1103	SAS	INMT	A2	SAP, MINSAP	*					*	
1104	PANE	IPCT	C5	ANE (V5)			*			*	
1105	PLO	IPCT	B5	ANE (V6)	*					*	
1106	DEMFI	IPCT	B2	DEFBAR	*					*	+DEFINE
1107	DELDES	IPCT	B2	DELTA	*					*	
1108	ABC	IPCT	A1	SECON	*						
1109	DIRECT	IPCT	A1	SECON	*		*	*	*	*	
1110	ANTEPAN	IPCT	A1	PREMAS	*						
1111	DESPLAN	IPCT	A1	CIMPPLAN	*		*	*	*	*	
1112	TRANS	IPCT	A1	CASE	*		*	*	*	*	
1113	CURBE	IPCT	A1	MORSEP	*		*	*	*	*	
1114	INTAR2	IPCT	A1	ARDIAF	*						
1115	DESASP	IPCT	A1	SPECTRE	*				*	*	+PRES
1116	GEOM	IPCN	B5	NELSAS	*				*		

II-M MICRO		Programe de calcul pentru MICROcalculatoare TPD, CUBZ si JUNIOR				TIP CALCULATOR	
COD DENUMIRE		INSTI-	CLS.	F U N C T I U N I L E	PROGRAMELOR	NECESAR	
		TUTIE				TPD CUBZ JUNIO	
M01-	CBA1-	CPJSi-	A2	Calculul de rezistenta al sectiuni-		*	*
M17	CBA17	biu		lor de beton armat			
M18	CPLAN1	CPJSi-	A2	Calculul cimpului plan de tempera-		*	*
		biu		tura in elementele de constructie			
M19	COMPLET	IPCT	A2	Analiza modala la hale.	*		
M20	CSIC	IPCT	A2	Calculul stilpilor centrifugati cu			*
				sectiune inelara.			
M21	TBP	IPCT	A2	Verificarea elementelor de b.a.			*
				precomprimat la forta traietoare.			
M22	SEGURA	IPCT	A2	Calculul caracteristicilor geome-		*	
				trice ale sectiunilor.			
M23	CALISB	IAMN	A2	Calculul sectiunilor de beton ar-		*	
				mat la stari limite ultime.			

III-1 PC		Programe de calcul pentru microcalculatoare(Personal Computer)					
		AUTOR					
COD DENUMIRE		INSTI-	CLS.	F U N C T I U N I	A L E	P R O G R A M E L O R	
		TUTIE					
P101	CARD	IPCT	A1	Calculul caracteristicilor geometrice sectionale.			
P102	MINC	IPCT	A1	Calculul momentelor de incastrare perfecta.			
P103	GRIM	IPCT	A1	Calculul momentelor maxime pe bara.			
P104	GC	IPROTIM+	A1	Analiza statica la grinzi continue.			
P105	SPA	IPROTIM+	A1	Analiza statica a fermelor plane.			
P106	CP	IPROTIM+	A1	Analiza statica a cadrelor plane oarecari.			
P107	CPO	IPROTIM+	A1	Analiza statica a cadrelor plane rectangulare.			
P108	CPDIN	IPROTIM+	A1	Analiza seismica (P100) la cadre plane oarecari.			
P109	CPSTAB	IPROTIM+	A1	Analiza la stabilitate a cadrelor plane oarecari.			
P110	DINO	IPCT	A1	Dimensionarea nodurilor de cadru (din b.a)			
P111	VERA	IPCT	A1	Verificarea agrafelor la panourile de perete.			
P112	TERMO	IPCT	A1	Calculul termotehnic al panourilor de perete.			

NOTA: IPROTIM+ Se va intelege ca autori IPROTIM+ICCPDC Timisoara

4. Conținutul fișierului MENU.DAT (meniurile înlănțuite de interogare):

OPTIUNEA Dvs. (A1) :

X = EXIT BPC; Y = REVENIRE MENIU PRECEDENT; Z = REVENIRE MENIU PRINCIPAL;

D O R I T I :

- 1 - INFORMATII DESPRE UN PROGRAM AFLAT IN *BPC* ?
- 2 - SA STITI DACA UN ANUMIT PROGRAM ESTE INCLUS IN *BPC* ?
- 3 - INFORMATII DESPRE O CLASA DE PROGRAME CATALOGATE IN *BPC* ?

I N D I C A T I N U M E L E P R O G R A M U L U I (max. 8 caract. MARI)

D O R I T I URMATOARELE INFORMATII DESPRE PROGRAMUL SPECIFICAT :

- 1 - TOATE INFORMATIILE POSIBILE !
- 2 - TIPUL, CLASIFICAREA PROGRAMULUI !
- 3 - OBIECTUL, SCOPUL PROGRAMULUI !
- 4 - AUTOR(I) (NUME, NR. TELEFON / INSTITUTIE, ADRESA) !
- 5 - ANUL ELABORARII / REVIZUIRII PROGRAMULUI !
- 6 - DOMENIUL DE APLICABILITATE !
- 7 - PERFORMANTE - LIMITARI (RESTRICTII) !
- 8 - INTERFATA (DIALOGUL) CU UTILIZATORUL !
- 9 - LIMBAJELE DE PROGRAMARE FOLOSITE LA ELABORAREA PROGRAMULUI !
- A - IMPLEMENTARI PE TIPURI DE CALCULATOARE !
- B - STAREA PROGRAMULUI - NIVELUL DOCUMENTARII !

D O R I T I ANUMITE INFORMATII DESPRE DOMENIUL DE APLICABILITATE :

- 1 - TOATE INFORMATIILE POSIBILE !
- 2 - TIPUL (MODELUL) STRUCTURII !
- 3 - TIPUL (MODELUL) ELEMENTELOR !
- 4 - TIPUL SECTIUNILOR ELEMENTELOR !
- 5 - TIPUL (MODELUL) LEGATURILOR !
- 6 - TIPURILE DE MATERIALE ACCEPTATE !
- 7 - TIPURILE DE ACTIUNI CONSIDERATE !
- 8 - TIPUL (MODELUL) ANALIZEI APLICATE !
- 9 - IPOTEZE ACCEPTATE, METODE DE CALCUL FOLOSITE !
- A - NORME CONSIDERATE, OPTIUNI OFERITE, CALCULE SPECIALE !

D O R I T I ANUMITE INFORMATII CU PRIVIRE LA "PERFORMANTE - LIMITARI" :

- 1 - TOATE INFORMATIILE POSIBILE !
- 2 - GABARITELE MAXIMALE ALE APLICATIILOR CARE POR FI REZOLVATE !
- 3 - DURATELE MEDII DE EXECUTIE LA EXEMPLELE TEST !
- 4 - REZULTATELE SI INFORMATIILE FURNIZATE DE PROGRAM !
- 5 - RESTRICTII - LIMITARI IN APLICAREA PROGRAMULUI !

D O R I T I INFORMATII PRIVIND DIALOGUL "UTILIZATOR - CALCULATOR" :

- 1 - TOATE INFORMATIILE POSIBILE !
- 2 - MODUL DE OPERARE (DIALOGARE) !
- 3 - STRUCTURA DATELOR DE INTRARE !
- 4 - POSIBILITATI DE PRE / POST-PROCESARE A DATELOR / REZULTATELOR !
- 5 - INTRARI / IESIRI GRAFICE ACCEPTATE !

D O R I T I :

- 1 - LISTA PROGRAMELOR DINTR-O ANUMITA CLASA (CATEGORIE) SPECIFICATA !
- 2 - LISTA PROGRAMELOR DINTR-UN ANUMIT PACHET SPECIFICAT !
- 3 - LISTA PROGRAMELOR ELABORATE DE ANUMITI AUTORI (1 la max. 4 autori) !

- 4 - RELATII CU PRIVIRE LA ANUMITI AUTORI DE PROGRAME (TELEFON - ADRESE) !
- 5 - LISTA PROGRAMELOR ELABORATE / REVIZUITE INTRE ANUMITE DATE (ANI) !
- 6 - LISTA PROGRAMELOR CARE POT FI RULATE PE UN CALCULATOR SPECIFICAT !
- 7 - LISTA PROGRAMELOR SIMILARE, DEDICATE UNOR APLICATII SPECIFICATE !
- 8 - IDENTIFICAREA PROGRAMELOR CU DOMENIUL DE APLICABILITATE SPECIFICAT !

P R E C I Z A T I C L A S A P R O G R A M E L O R D O R I T E :

S P E C I F I C A T I T I P U L P R O G R A M E L O R D O R I T E :

S E L E C T A T I O B I E C T U L P R O G R A M U L U I :

- | | | | |
|---|-------------------------------------|---|---------------------------------------|
| 1 | CALCULUL STRUCTURILOR PORTANTE. | H | PROIECTAREA TUNELELOR. |
| 2 | CALCULUL LA STABILITATE. | I | GEOTEHNICA, MECANICA PAMINTURILOR. |
| 3 | ANALIZA ELEMENTELOR DE CONSTRUCTII. | J | CALCULE DE OPTIMIZARE. |
| 4 | DIMENSIONAREA ELEMENTELOR PORTANTE. | K | ARHITECTURA IN CONSTRUCTII. |
| 5 | VERIFICAREA ELEMENTELOR STRUCTURALE | L | PRODUSE GRAFICE, DESEN AUTOMAT. |
| 6 | PROIECTAREA IMBINARILOR. | M | PROIECTAREA INSTALATIILOR INTERIOARE. |
| 7 | ANALIZA ELEMENTELOR SI SECTIUNILOR. | N | PROIECTAREA INSTALATIILOR EXTERIOARE. |
| 8 | ANALIZA SECTIUNILOR. | O | CALCULE HIDRAULICE - HIDROTEHNICE. |
| 9 | DIMENSIONAREA SECTIUNILOR. | P | ANALIZA LA TRANSFER (FLUX) TERMIC. |
| A | VERIFICAREA SECTIUNILOR. | Q | CALCULE HIGROTERMICE. |
| B | TRATAREA DIVERSELOR INTERACTIUNI. | R | ACUSTICA, VIBRATII IN CONSTRUCTII. |
| C | TRATAREA ACTIUNILOR SEISMICE. | S | PRE / POST-PROCESARE, SISTEME EXPERT. |
| D | TRATAREA ACTIUNILOR SPECIALE. | T | CALCULE ECONOMICE IN CONSTRUCTII. |
| E | PROIECTAREA PODURILOR. | U | CALCULE DE PLANIFICARE / EVIDENTA. |
| F | PROIECTAREA DRUMURILOR. | V | |
| G | PROIECTAREA CAILOR FERATE. | W | |

P R E C I Z A T I S U B C L A S A P R O G R A M E L O R D O R I T E :

P R E C I Z A T I O B I E C T U L P R O G R A M U L U I :

- | | | | |
|---|--------------------------------------|---|---------------------------------------|
| 1 | CALCUL LINIAR, STRUCTURI PLANE. | 6 | CALCULUL STRUCTURILOR COMPLEXE. |
| 2 | CALCUL LINIAR, STRUCTURI SPATIALE. | 7 | CALCULUL STRUCTURILOR PRETENSIONATE. |
| 3 | CALCUL LINIAR, PRIN SUBSTRUCTURARE. | 8 | CALCUL CU ELEMENTE FINITE/FRONTIERA. |
| 4 | CALCUL NELINIAR, STRUCTURI PLANE. | 9 | |
| 5 | CALCUL NELINIAR, STRUCTURI SPATIALE. | A | |
| 1 | ANALIZA STABILITATII GENERALE | 6 | |
| 2 | CALCULE DE STABILITATE LOCALA. | 7 | |
| 3 | PROBLEME DE FLAMBAJ. | 8 | |
| 4 | PROBLEME DE VOALARE. | 9 | |
| 5 | INSTABILITATEA DINAMICA. | A | |
| 1 | ANALIZA REZISTENTEI ELEMENTELOR. | 6 | CURBE DE INTERACTIUNE LA SOLICITARI. |
| 2 | ANALIZA RIGIDITATII ELEMENTELOR. | 7 | |
| 3 | ANALIZA DUCTILITATII ELEMENTELOR. | 8 | |
| 4 | ANALIZA LA REZIST.-RIGID.-DUCTILIT. | 9 | |
| 5 | ANALIZA LA OBOSEALA. | A | |
| 1 | GRINZI METALICE. | 6 | GRINZI DE RULARE. |
| 2 | STILPI METALICI. | 7 | PLACI, PLANSEE, DALE. |
| 3 | GRINZI DE BETON ARMAT/PRECOMPRIMAT. | 8 | PERETI, PANOURI, DIAFRAGME, SAIBE. |
| 4 | STILPI DE BETON ARMAT. | 9 | FUNDATII. |
| 5 | ELEMENTE DIN LEMN. | A | |
| 1 | VERIFICARI GRINZI METALICE. | 6 | VERIFICARI GRINZI DE RULARE. |
| 2 | VERIFICARI STILPI METALICI. | 7 | VERIFICARI PLACI, PLANSEE, DALE. |
| 3 | VERIFICARI GRINZI BET.ARM./PRECOMPR. | 8 | VERIFICARI PERETI, PANOURI, DIAFRAGME |
| 4 | VERIFICARI STILPI DE BETON ARMAT. | 9 | VERIFICARI FUNDATII. |
| 5 | VERIFICARI ELEMENTE DIN LEMN. | A | |

1 PROIECTAREA IMBINARILOR NITUITE.	6 PROIECTAREA APARATELOR DE REAZEM.
2 PROIECTAREA IMBINARILOR BULONATE.	7 PROIECTAREA ARTICULATIILOR.
3 PROIECTAREA IMBINARILOR SUDATE.	8 PROIECTAREA ANCORAJELOR, TIRANTILOR.
4 PROIECTAREA NODURILOR FERMELOR.	9
5 PROIECTAREA NODURILOR DE CADRE.	A
1 ANALIZA LA REZISTENTA.	6 CURBE DE INTERACTIUNE LA SOLICITARI.
2 ANALIZA LA RIGIDITATE, FISURARE.	7
3 ANALIZA LA DUCTILITATE.	8
4 ANALIZA LA REZIST.-RIGID.-DUCTILIT.	9
5 ANALIZA LA OBOSEALA.	A
1 ANALIZA SECTIUNILOR METALICE.	6
2 ANALIZA SECTIUNILOR DE BETON ARMAT.	7
3 ANALIZA SECTIUNILOR DE BETON PRECOMP.	8
4 ANALIZA SECTIUNILOR DIN LEMN.	9
5 ANALIZA SECTIUNILOR DIN ALTE MATER.	A CALCULUL CARACTERISTICILOR SECTIONALE
1 SECTIUNI METALICE.	6
2 SECTIUNI DE BETON ARMAT.	7
3 SECTIUNI DE BETON PRECOMPRIMAT.	8
4 SECTIUNI DIN LEMN.	9
5 SECTIUNI DIN ALTE MATERIALE	A CALCULUL CARACTERISTICILOR SECTIONALE
1 VERIFICARI SECTIUNI METALICE.	6
2 VERIFICARI SECTIUNI DE BETON ARMAT.	7
3 VERIFICARI SECTIUNI DE BETON PRECOMP.	8
4 VERIFICARI SECTIUNI DIN LEMN.	9
5 VERIFICARI SECTIUNI DIN ALTE MATER.	A CALCULUL CARACTERISTICILOR SECTIONALE
1 INTERACTIUNI TEREN-FUNDATIE.	6 INTERACTIUNI PANOU INFILAT-CADRU.
2 INTERACTIUNI TEREN-FUNDATIE-STRUCTURA	7 INTERACTIUNI INCARCARI - STRUCTURA.
3 SISTEME FUNDATIE-AMORTIZOR-STRUCTURA.	8
4 SISTEME CU DISIPATORI ENERGETICI.	9
5 SISTEME CU CONTROL ACTIV.	A
1 DECONVOLUTII SEMNAL SEISMIC.	6 GENERARI SPECTRE DE RASPUNS.
2 GENERARI ACCELEROGRAME ARTIFICIALE.	7
3 PRELUCRARI DE ACCELEROGRAME.	8
4 ACTUALIZARI DE ACCELEROGRAME.	9
5 TRATARI COMPLEXE (GEN.+PREL.+ACTUAL.)	A BANCII DE ACCELEROGRAME / SPECTRE.
1 ACTIUNI DIN CUTREMURE.	6 EXPLOZII CONVENTIONALE / ATOMICE.
2 ACTIUNI DIN ALUNECARI / INUNDATII.	7 EXPLOZII IN INTERIORUL CLADIRILOR.
3 ACTIUNI DIN VINT (RAFALE, EF.KARMAN).	8 EFECTE HISTERETICE (Curbe reologice).
4 EFECTE DE FLUTURARE (FLUTTER).	9
5 EFECTE DIN FLUAJ/CONTRACTIE/ECRUISAJ.	A
1 PROIECTAREA TRAVERSARILOR SIMPLE.	6 PROIECTAREA PODURILOR HOBANATE.
2 PROIECTAREA PODURILOR METALICE.	7 PROIECTAREA PODURILOR SUSPENDATE.
3 PROIECTAREA PODURILOR DE BETON ARMAT.	8
4 PROIECTAREA PODURILOR DE LEMN.	9
5 PROIECTAREA PODURILOR MIXTE.	A
1 BAZELE DINAMICII TRAFICULUI RUTIER.	6 TRASAREA DRUMURILOR.
2 PROIECTAREA AUTOSTRAZILOR.	7 CONFIGURAREA SECTIUNII TRANSVERSALE.
3 PROIECTAREA SUPRATRAVERSARILOR.	8 MODERNIZARI, CONSOLIDARI DE DRUMURI.
4 PROIECTAREA INTERSECTIILOR.	9 CONSTRUCTII AUXILIARE AFERENTE.
5 PROIECTAREA PISTELOR (AVIOANE/CURSE).	A INSTALATII DE REGLARE A TRAFICULUI.
1 BAZELE DINAMICII TRAFICULUI PE C.F.	6 TRASAREA CAILOR FERATE.

2 PROIECTAREA CAILOR FERATE NORMALE.	7 CONFIGURAREA SECTIUNII TRANSVERSALE.
3 PROIECTAREA SUPRATRAVERSARILOR.	8 MODERNIZARI, CONSOLIDARI DE C.F.
4 PROIECTAREA STATIILOR (GARILOR).	9 CONSTRUCTII SPECIFICE CAILOR FERATE.
5 PROIECTAREA CAILOR FERATE SPECIALE.	A INSTALATII AFERENTE TRAFICULUI C.F.
1 PROBLEME DE MECANICA ROCILOR.	6 TRASAREA TUNELELOR.
2 PROIECTAREA GALERIILOR HIDROTEHNICE.	7 CONFIGURAREA SECTIUNII TRANSVERSALE.
3 PROIECTAREA TUNELELOR CLASICE.	8 MODERNIZARI, CONSOLIDARI DE TUNELE.
4 PROIECTAREA METROURILOR.	9 CONSTRUCTII SUBTERANE SPECIALE.
5 PROIECTAREA TUNELELOR SPECIALE.	A INSTALATII AFERENTE TUNELELOR.
1 DETERMINARI CARACTERISTICI TEREN.	6
2 CONSOLIDARI TEREN FUNDATIE.	7
3 CALCUL IMPINGERI ACTIVE / PASIVE.	8
4 CALCUL TASARI.	9
5 CALCUL SISTEME CU PILOTI.	A BANCII CU DATE GEOTEHNICE.
1 METODE DE OPTIMIZARE UNICRITERIALE.	6 OPTIMIZARI DE STRUCTURI.
2 METODE DE OPTIMIZARE MULTICRITERIALE.	7 OPTIMIZARI DE CONSTRUCTII.
3 OPTIMIZARI DE SECTIUNI.	8
4 OPTIMIZARI DE ELEMENTE.	9
5 OPTIMIZARI DE SUBANSAMBLURI.	A
1 PERSPECTIVE, AXONOMETRII, UMBRIRI.	6 TRATARI USI, FERESTRE, FINISAJE.
2 LUMINA, CULOARE, VIZIBILITATE.	7 SISTEMATIZARI (MICRO) TERITORIALE.
3 MODULARI ORIZONTALE / VERTICALE.	8
4 TRATAREA PARTIURILOR / OPTIMIZARI.	9
5 TRATAREA SUPRAFETELOR / VOLUMELOR.	A
1 TRASARI FUNCTII, GRAFICE, DIAGrame.	6
2 REPREZENTARI GRAFICE "2D".	7
3 REPREZENTARI GRAFICE "3D".	8
4 INTERFETE GRAFICE.	9
5 PRE / POST-PROCESOARE GRAFICE.	A BIBLIOTECI DE MODULE GRAFICE.
1 INSTALATII APA, CANAL, SANITARE.	6 SISTEME ENERGETICE NECONVENTIONALE.
2 INSTALATII ELECTRICE (LUMINA, FORTA).	7
3 INSTALATII GAZE, INCALZIRI.	8
4 VENTILARI, AER CONDITIONAT.	9
5 ASCENSOARE, ELEVATOARE ETC.	A
1 RELETE APA, CANALIZARE.	6 SISTEME ENERGETICE NECONVENTIONALE.
2 RELETE ELECTRICE (LUMINA, FORTA).	7
3 RELETE GAZE, TERMOFICARE.	8
4 RELETE INDUSTRIALE.	9
5 TELEFERICE, FUNICULARE ETC.	A
1 ALIMENTARI CU APA (CAPTARI).	6
2 CANALE, IRIGATII, DIGURI, PORTURI.	7
3 PROIECTAREA BARAJELOR.	8
4 CONSTRUCTII HIDROTEHNICE.	9
5	A
1 SISTEME CU FLUX UNIDIRECTIONAL.	6
2 SISTEME CU FLUX BIDIRECTIONAL.	7
3 SISTEME CU FLUX SPATIAL.	8
4 CALCULE GLOBALE (PE CLADIRE).	
5 VALORIFICAREA ENERGIEI SOLARE.	
1 PERETI, PANOURI INCHIDERE.	6
2 PLACI, PLANSEE TERASA.	7
3 PUNTI TERMICE SIMPLE.	8

4 ZONE INTERSECȚIE ELEMENTE COMPLEXE.
5 PROBLEME DE CONDENS.

1 ACUSTICA ARHITECTURALA.
2 PROTECTIA (IZOLAREA) LA ZGOMOTE.
3 IZOLAREA LA VIBRAȚII / SOCURI.
4
5

1 PREPROCESOARE.
2 POSTPROCESOARE.
3 PRE / POST-PROCESOARE.
4 INTERPRETOARE - COMPILATOARE.
5 LINII AUTOMATE DE PROIECTARE.

6 MEDII DE PROIECTARE INTEGRATE.
7 SISTEME CAD (PAC) DEDICATE.
8 SISTEME EXPERT.
9

1 ANTEMASURATORI, DEVIZE, EXTRASE.
2 GESTIUNEA RESURSELOR.
3
4
5

6
7 BAZE DATE INDICI TEHNICO-ECONOMICI.
8 BAZE DATE INDICATORI DE PERFORMANTA.
9 EVALUARE EFICIENTA LUCRARI.
A BAZE DE DATE GENERALE.

1 PROGRAMAREA RESURSELOR.
2 PROGRAMAREA PROCESELOR.
3 OPTIMIZARE FLUX PRODUCTIE.
4
5

6
7
8
9
A

1
2
3
4
5

6
7

1
2
3
4
5

6
7

S P E C I F I C A T I P A C H E T U L D E P R O G R A M E :

1 - * ALIS-01 *	7 - * ELCO *	D - * ALIS-PC *	J - * PAS-PC *
2 - * ALIS-02 *	8 - * TERRA *	E - * NELI-PC *	K - * PP-PC *
3 - * ALIS-03 *	9 - * FIZCO *	F - * MEF-PC *	L - *
4 - * NELI *	A - * PAS *	G - * SECO-PC *	M - *
5 - * MEF *	B - * PP *	H - * ELCO-PC *	N - *
6 - * SECO *	C - * MICRO *	I - * FIZCO-PC *	O - *

SELECTATI NUMERELE DE ORDINE ALE NUMELOR DE AUTORI [1 - max. 4 numere (4I4)]

1 Capatina D.	22 Enescu C.	42 Pecol E.	63 Cenariu A.	18 Constantinescu D
2 Linaru M.	23 GospodinovN.	43 Sabareanu E.	64 Pacoste C.	53 Stanciulescu D.
3 Cocora D.	24 Borcia I.	44 Bota V.	65 Noaghiu A.	74 Constantinescu G
4 Pop R.	25 Sandi H.	45 Mihalache N.	66 Olariu I.	84
5 GoncerencoT.	26 Pirv B.	46 Filipescu C.	67 Marinov R.	85
6 Bodea I	27 AppeltauerI.	47 PostelnicuT.	68 Ciomocos T.	86
7 Dinu V.	28 Cuteanu E.	48 Dogaru L.	69 Gluck N.	87
8 Gobesz F.	29 Giurgea D.	49 Cismigiu P.	70 Moga I.	88
9 Gobesz Zs.	30 Dutulescu E.	50 Cristea I.	71 Comsa E.	89
10 Cristea Gh.	31 Crainic L.	51 Onet T.	72 Balosache N.	90
11 Catarig Al.	32 Banut V.	52 Man M.	73 Nedelcu M.	91
12 Petrina M.	33 Cotescu A.	54 Clipii T.	75 Rosca M.	92
13 Coheci T.	34 Stancu O.	55 Mihaila N.	76 DumitrescuI.	93

14 Olariu I.Cj	35 Olariu O.	56 CuciureanuF.	77	94
15 Kopenetz L.	36 MucichescuD.	57 Iliesci Fl.	78	95
16 Cornea T.	37 Cretu D.	58 Bogos C.	79	96
17 Ciongradi I.	38 Dosa A.	59 Brosteanu M.	80	97
19 Malita M.	39 Stoian V.	60 Luchian S.	81	98
20 Botezat C.	40 Friedrich R.	61 Cristea D.	82	99
21 Chisalita A.	41 FurtunescuA.	62 Vornicu T.	83	100

RASPUNSUL Dvs. (4I4):

X = EXIT BPC; Y = REVENIRE MENU PRECEDENT; Z = REVENIRE MENU PRINCIPAL;

P R E C I Z A T I (CONFORM INDICATIILOR DIN EXEMPLIFICARI) :

- 1 - INTERVALUL DE ELABORARE (Expl. 1,76-82 pentru intervalul 1976-1982) ;
- 2 - INTERVALUL DE REVIZUIRE (Expl. 2,80-90 pentru intervalul 1980-1990) ;
- 3 - INTERVALUL DE ELABORARE/REVIZUIRE (Expl. 3,75-91 pentru 1975-1991) etc

RASPUNSUL Dvs. (2I2,1X,I2):

X = EXIT BPC; Y = REVENIRE MENU PRECEDENT; Z = REVENIRE MENU PRINCIPAL;

S P E C I F I C A T I T I P U L C A L C U L A T O R U L U I D v s :

- 1 - SISTEME FELIX C-256 / 512 / 1024 ;
- 2 - MINICALCULATOARE: PDP-11/34; I100; I102-F; CORAL 4030 etc ;
- 3 - PC-XT/AT COMPATIBILE IBM (286, 386, 486 etc) ;
- 4 - MICROCALCULATOARE: M18; M118; TPD; CUB-Z; JUNOIR etc ;
- 5 - MICROCALCULATOARE: TIM-S; aMIC; PRAE-M; SINCLAIR; COMMODORE etc ;
- 6 - HEWLETT - PACKARD (9830-A etc) ;
- 7 - ALTE TIPURI.

S P E C I F I C A T I T I P U L S T R U C T U R I I :

- | | |
|---|--|
| 1 GRINZI, SISTEME DE GRINZI. | G SUSPENDATE, HOBANATE. |
| 2 STRUCTURI ARTICULATE PLANE. | H TUBULARE (TUB-IN-TUB, TUB PERFORAT). |
| 3 STRUCTURI ARTICULATE SPATIALE. | I STRUCTURI CELULARE, CASETATE. |
| 4 CADRE PLANE. | J STRUCTURI CU TENSIUNI INITIALE. |
| 5 CADRE SPATIALE. | K PNEUMATICE (AERO-PURTATE/PORTANTE). |
| 6 HALE INDUSTRIALE CU PODURI RULANTE. | L FUNDATII, SISTEME DE FUNDARE. |
| 7 RELETE (GRINZI, BARE, CABLURI). | M ZIDURI (PORTANTE, DE SPRIJIN ETC). |
| 8 OARECARI (ARTIC/CADRE-PLANE/SPATIALE) | N |
| 9 STRUCTURI CU DIAFRAGME, PANOURI. | O |
| A STRUCTURI MIXTE (CADRE-DIAFRAGME). | P |
| B PINZE SUBTIRI, CUPOLE, ARCE. | Q |
| C PLACI, DALE, PLANSEE. | R |
| D TURNURI, COSURI, CASTELE DE APA. | S |
| E REZERVOARE, BUNCARE, SILOZURI. | T |
| F AUTOPORTANTE (CU CABLURI, MEMBRANE). | U |

S P E C I F I C A T I T I P U L L E G A T U R I L O R :

- 1 - NODURI IDEALE (PUNCTIFORME, FARA FRECARI).
- 2 - NODURI RIGIDE DE DIMENSIUNI FINITE.
- 3 - LEGATURI ELASTICE LA CAPETELE BARELOR.
- 4 - NODURI DE DIMENSIUNI FINITE RIGIDE, CU LEGATURI ELASTICE.
- 5 - NODURI NEPUNCTIFORME, TRATATE CU METODA ELEMENTELOR FINITE.

S P E C I F I C A T I S I S T E M U L D E G R I N Z I :

- 1 - GRINZI SIMPLU REZEMATE.
- 2 - GRINZI CONTINUE.
- 3 - GRINZI CONTINUE PE REAZEME ELASTICE.

- 4 - GRINZI REZEMATE PE MEDIU ELASTIC (Model Winkler etc).
- 5 - GRINZI PRECOMPRIMATE / PRETENSIONATE.
- 6 - GRINZI CU ZABRELE (FERME).
- 7 - GRINZI DE RULARE.
- 8 - GRINZI AJURATE.
- 9 - GRINZI GERBER (CU CONSOLE SI ARTICULATII).
- A - GRINZI VIERENDEEL (TIP CADRU, SCARA).
- B - GRINZI CURBE PLANE, INELARE.
- C - RETELE GRINZI.
- D - GRINZI HELICOIDALE.
- E -
- F -

S P E C I F I C A T I T I P U L A R C U L U I / P I N Z E I :

- 1 - ARCE CIRCULARE CU 3 ARTICULATII.
- 2 - ARCE PARABOLICE CU 3 ARTICULATII.
- 3 - ARCE CIRCULARE DUBLU ARTICULATE.
- 4 - ARCE PARABOLICE DUBLU ARTICULATE.
- 5 - ARCE CIRCULARE DUBLU INCASTRATE.
- 6 - ARCE PARABOLICE DUBLU INCASTRATE.
- 7 - ARCE CU TIRANT (CIRCULARE / PARABOLICE).
- 8 - ARCE OARECARI (INCLUSIV PRECOMPRIMATE).
- 9 -
- A - PINZE SUBTIRI AXIAL SIMETRICE.
- B - PINZE SUBTIRI CILINDRICE.
- C - PINZE SUBTIRI TIP HYPAR (PARABOLOID HIPERBOLIC).
- D - PINZE SUBTIRI DE TRANSLATIE.
- E - PINZE SUBTIRI CONOIDALE.
- F - CUPOLE GEODEZICE / CU ZABRELE ETC.

S P E C I F I C A T I T I P U L F U N D A T I E I :

- 1 - FUNDATII IZOLATE RIGIDE (CU CUZINET).
- 2 - FUNDATII IZOLATE FLEXIBILE (TIP PAHAR).
- 3 - FUNDATII CONTINUE UNIDIRECTIONALE.
- 4 - RETELE DE GRINZI.
- 5 - FUNDATII CONTINUE / RETELE GRINZI PE MEDIU ELASTIC.
- 6 - RADIERE RIGIDE.
- 7 - RADIERE FLEXIBILE.
- 8 - FUNDATII CIRCULARE, INELARE.
- 9 - FUNDATII TIP PINZE SUBTIRI.
- A - FUNDATII DE MASINI.
- B - FUNDATII PE PILOTI.
- C -
- D -
- E -
- F - FUNDATII SPECIALE.

S P E C I F I C A T I T I P U L A N A L I Z E I D O R I T E :

- 1 - CALCUL STATIC.
- 2 - CALCUL BIOGRAFIC (ELASTO-PLASTIC).
- 3 - CALCUL SEISMIC CONVENTIONAL (PE BAZA DE COD P-100).
- 4 - CALCUL POSTELASTIC (LA CEDARE).
- 5 - CALCUL DINAMIC.
- 6 - CALCUL LA OBOSEALA.
- 7 - CALCUL LINII / SUPRAFETE / ZONE / COEFICIENTI DE INFLUENTA.
- 8 -
- 9 -
- A -
- B -

- C -
- D -
- E - CALCUL LA STABILITATE GENERALA.
- F - CALCUL LA STABILITATE LOCALA.

S P E C I F I C A T I M O D U L D E A N A L I Z A D O R I T :

- 1 - ANALIZA LINIAR ELASTICA.
- 2 - ANALIZA PLASTICA (LA CEDARE).
- 3 - ANALIZA ELASTO-PLASTICA.
- 4 - ANALIZA FIZIC NELINIARA (REOLOGICA).
- 5 - ANALIZA NELINIARA DE ORDINUL 2.
- 6 - ANALIZA NELINIARA DE ORDINUL 3.
- 7 - ANALIZA MODALA.
- 8 - ANALIZA PE BAZA DE COD (P-100/81/91 ETC).
- 9 - ANALIZA PE BAZA DE SPECTRE.
- A - ANALIZA PE BAZA DE ACCELEROGRAME.
- B - ANALIZA TIP "TIME-HISTORY" (INCREMENTALA, HISTERETICA).
- C - ANALIZA INCREMENTAL-ITERATIVA (ACTIUNI NECONSERVATIVE).
- D - ANALIZA CU ELEMENTE FINITE (SI/SAU DE FRONTIERA).
- E -
- F -

S P E C I F I C A T I C O M P O R T A R E A M A T E R I A L E L O R :

- 1 - LINIAR ELASTICA (HOOKE).
- 2 - NELINIAR ELASTICA (Curba SIGMA-EPSILON specificata).
- 3 - ELASTO-PLASTICA (Model PRANDTL).
- 4 - ELASTICA BILINIARA (Curba SIGMA-EPSILON / P-Delta, data).
- 5 - ELASTICA TRILINIARA (Curba SIGMA-EPSILON / P-Delta, data).
- 6 - VISCO-ELASTICA (Curbele reologice fiind specificate).
- 7 - VISCO-ELASTO-PLASTICA (Curbele reologice fiind date).
- 8 -
- 9 -
- A -

S P E C I F I C A T I T I P U L A C T I U N I L O R :

- 1 - INCARCARI STATICE FIXE (FORTE, MOMENTE).
- 2 - INCARCARI STATICE CRESCATOARE (FORTE, MOMENTE).
- 3 - INCARCARI INERTIALE, PSEUDOSTATICE.
- 4 - INCARCARI STATICE MOBILE (CONVOAIE SARCINI).
- 5 - VARIATII DE TEMPERATURA (STATIONARE).
- 6 - CEDARI DE REAZEME / DEPLASARI IMPUSE.
- 7 - EFORTURI INITIALE (PRETENSIONARE etc).
- 8 - INCARCARI DINAMICE FIXE [DIGITIZATE $P(t)$].
- 9 - INCARCARI DINAMICE MOBILE [CONVOAIE $P(x,t)$].
- A - RAFALE VINT.
- B - EXPLOZII.
- C - INCARCARI NECONSERVATIVE [$P(x,y,z,t)$].
- D -
- E -
- F -

S P E C I F I C A T I M O D U L D E A P L I C A R E A L A C T I U N I L O R :

- 1 - IN NODURI (REDUSE LA NODURI).
- 2 - PE ELEMENTE (BARE).
- 3 - IN NODURI SI PE ELEMENTE.
- 4 - IN AXUL ELEMENTELOR.
- 5 - INTRE FETELE ELEMENTELOR.
- 6 - IN AX SI INTRE FETE ELEMENTE.

- 7 - PRIN ACCELEROGRAME DIGITIZATE LA BAZA STRUCTURII.
 8 -
 9 -
 A -

SPECIFICATI DESCRIEREA TIPULUI STRUCTURII (max. 10I4):

G R U P A 1		G R U P A 2	
1 MIXTE	16 NERVURATE	1 GRINZI	16 OARECARI
2 RETELE	17 AUTOPORTANTE	2 CU ZABRELE	17 PLANE
3 CU ZABRELE	18 HOBANATE	3 DE RULARE	18 SPATIALE
4 CADRE	19 SUSPENDATE	4 CONTINUE	19 SIMPLU
5 PARTER	20 CU CABLURI	5 GERBER	20 REZEMATE
6 ETAJATE	21 CU MEMBRANE	6 VIERENDEEL	21 ELASTIC
7 SIMETRICE	22 ZID PORTANT	7 CURBE	22 PE MEDIU ELAST
8 BIDIMENSIONALE	23 ARTICULATE	8 HELICOIDALE	23 CU I CONSTANT
9 PLANARE	24 INCOVOIATE	9 CASETATE	24 CU I VARIABIL
10 ORTOGONALE	25 FORFECATE	10 HOBANATE	25 IN TREPTE
11 OARECARI	26 AEROPURTATE	11 PORTAL	26 CU VUTE
12 CU DIAFRAGME	27 AEROPORTANTE	12 CU INIMA PLINA	27 DREPTE
13 CU BARE	28 PLANE	13 AJURATE	28 PARABOLICE
14 PLACI, DALE	29 SPATIALE	14 IN RETEA	29 CONSOLIDATE
15 CASETATE	30 PRETENSIONATE	15 SPECIALE	30 PRETENSIONATE

RASPUNSUL Dvs. (Nr.GRUPA + Specificatii):

G R U P A 3		G R U P A 4	
1 PINZE	16 OARECARI	1 TURNURI, COSURI	16 MASIVE
2 CUPOLE	17 SIMETRICE	2 CASTELE DE APA	17 DE MASINI
3 ARCE	18 CU TIRANT	3 REZERVOARE	18 CONTINUE
4 SUBTIRI	19 CU ZABRELE	4 BUNCARE	19 RIGIDE
5 DE ROTATIE	20 CU INIMA PLINA	5 SILOZURI	20 FLEXIBILE
6 DE TRANSLATIE	21 DUBLU STRAT	6 FAGURE	21 PE MEDIU ELAST
7 CIRCULARE	22 MULTI STRAT	7 CELULARE	22 PE PILOTI
8 PARABOLICE	23 PRECOMPRIMATE	8 CU PANOURI	23 CHESONATE
9 HIPERBOLICE	24 SIMPLU	9 ZIDURI SPRIJIN	24 SPECIALE
10 CILINDRICE	25 REZEMATE	10 TUBULARE	25 PORTANTE
11 SFERICE	26 ARTICULAT	11 TUB IN TUB	26 AERIENE
12 ELIPSOIDALE	27 INCASTRAT	12 TUB PERFORAT	27 SUBTERANE
13 CONOIDALE	28 CU 3 ARTICUL.	13 FUNDATII	28 HALE PARTER
14 CUTATE	29 ELASTIC	14 IZOLATE	29 TIP RETEA
15 GEODEZICE	30 IN PUNCTE	15 RADIERE	30 CU POD RULANT

RASPUNSUL Dvs. (Nr.GRUPA + Specificatii):

SPECIFICATI DESCRIEREA TIPULUI DE ELEMENTE (max. 10I4):

G R U P A 1		G R U P A 2	
1 SUBSTRUCTURI	16 DE RULARE	1	16
2 GRINZI	17 AJURATE	2	17
3 RIGLE	18 PLACI, DALE	3	18
4 STILPI	19 FUNDATII	4	19
5 BARE	20 ZIDURI	5	20
6 CABLURI, FIRE	21 PLINE	6	21
7 DIAFRAGME	22 CU GOLURI	7	22
8 PANOURI	23 INTINSE	8	23
9 CHESON	24 COMPRIMATE	9	24
10 CU INIMA PLINA	25 FORFECATE	10	25
11 CU ZABRELE	26 INCOVOIATE	11	26

12 COMPUSE	27 CENTRIC	12	27
13 DREPTE	28 EXCENTRIC	13	28
14 CURBE	29 OBLIC	14	29
15 CONTRAVINTUIRI	30 PRETENSIONATE	15	30

RASPUNSUL Dvs. (Nr.GRUPA + Specificatii):

SPECIFICATI DESCRIEREA TIPULUI SECTIUNILOR (max. 10I4):

G R U P A 1		G R U P A 2	
1 CONSTANTA	16 PROFILE LAMIN.	1	16
2 VARIABILA	17 PERETI SUBTIRI	2	17
3 DUBLU	18 PERETI GROS	3	18
4 SIMETRICA	19 INCHISI	4	19
5 ARBITRARA	20 DESCHISI	5	20
6 COMPUSA	21 NEOMOGENA	6	21
7 DIN	22 ARMATA	7	22
8 DE FORMA	23 MULTI STRAT	8	23
9 DREPTUNGHIURI	24 T, dublu T etc	9	24
10 PATRATA	25 I, II etc	10	25
11 TRIUNGHIULARA	26 CU VUTE	11	26
12 CIRCULARA	27 DREPTE	12	27
13 INELARA	28 PARABOLICE	13	28
14 CHESONATA	29 IN TREPTE	14	29
15 CASETATA	30 TIP OBELISC	15	30

RASPUNSUL Dvs. (Nr.GRUPA + Specificatii):

SPECIFICATI DESCRIEREA TIPULUI LEGATURILOR (max. 10I4):

G R U P A 1		G R U P A 2	
1 IMBINARI	16 ARTICULATE	1	16
2 SUDATE, LIPITE	17 INCASTRATE	2	17
3 NITUIITE	18 ARBITRARE	3	18
4 BULONATE	19 ELASTICE	4	19
5 NODURI	20 PE MEDIU ELAST	5	20
6 IDEALE	21 CU DISCONTIN.	6	21
7 DE CADRU	22 IN DESCHIDERI	7	22
8 PUNCTIFORME	23 LIBERE	8	23
9 RIGIDE	24 BLOCATE	9	24
10 CU DIMENSIUNI	25 LA TRANSLATIE	10	25
11 FINITE	26 LA RASUCIRE	11	26
12 LEGATURI	27 LA ROTIRE	12	27
13 SIMPLE	28 LA CAPETEELE	13	28
14 REZEMARI	29 BARELOR	14	29
15 PENDULARE	30 LA REZEMARI	15	30

RASPUNSUL Dvs. (Nr.GRUPA + Specificatii):

SPECIFICATI DESCRIEREA TIPULUI MATERIALELOR (max. 10I4):

G R U P A 1		G R U P A 2	
1 BETON	16 PRECOMPRIMAT	1 LINIAR	16 PRANDTL
2 USOR	17 CU	2 NELINIAR	17 RAMBERG-OSGOOD
3 GREU	18 PROFILE	3 ELASTICA	18 VOIGT - KELVIN
4 MASE PLASTICE	19 CABLURI	4 VISCOS	19 BURGERS
5 AZBOCIMENT	20 TOROANE	5 PLASTICA	20 ZENER
6 COMPOZITE	21 INSERTIE	6 BILINIARA	21 CURBELE
7 STRATIFICATE	22 FIBRE, FIRE	7 TRILINIARA	22 CARACTERISTICE

8	NEOMOGENE	23	METAL	8	HISTERETICA	23	IZOCRONA
9	DIN	24	OTEL	9	GENERALA	24	SIGMA-EPSILON
10	LEMN	25	CONSTRUCTII	10	CU	25	SAU "P-DELTA"
11	LAMINAR	26	SUPERIOR	11	CONSOLIDARE /	26	CURGERE LENTA
12	INCLEIAT	27	ALUMINIU	12	DEGRADARE	27	EPSILON-TIMP
13	SIMPLU	28	STICLA	13	MODEL REOLOGIC	28	DE RELAXARE
14	DUBLU	29	CARBON	14	COMPLEX	29	SIGMA - TIMP
15	ARMAT (E)	30	CARAMIDA	15	HOOKE	30	FIIND DATE.

RASPUNSUL Dvs. (Nr.GRUPA + Specificatii):

SPECIFICATI DESCRIEREA TIPULUI ACTIUNILOR (max. 10I4):

G R U P A 1		G R U P A 2	
1	IPOTEZE	16	CRESCATOARE
2	STATIC	17	MOBILE
3	DINAMICE	18	NECONSERVATIVE
4	SEISMICE	19	DUPA COD P-100
5	CONVOAIE	20	CONCENTRATE
6	SPECIALE	21	DISTRIBUITE
7	FORTE	22	IN NODURI
8	MOMENTE	23	TENS. INITIALE
9	PRESIUNI	24	PE ELEMENTE
10	INERTIALE	25	VAR. TEMPERAT.
11	GRAVITATIONALE	26	IN AX
12	LATERALE	27	INTRE FETE EL.
13	ARBITRARE	28	CEDARI REAZEME
14	FIXE	29	RAFALE VINT
15	VARIABLE	30	EXPLOZII

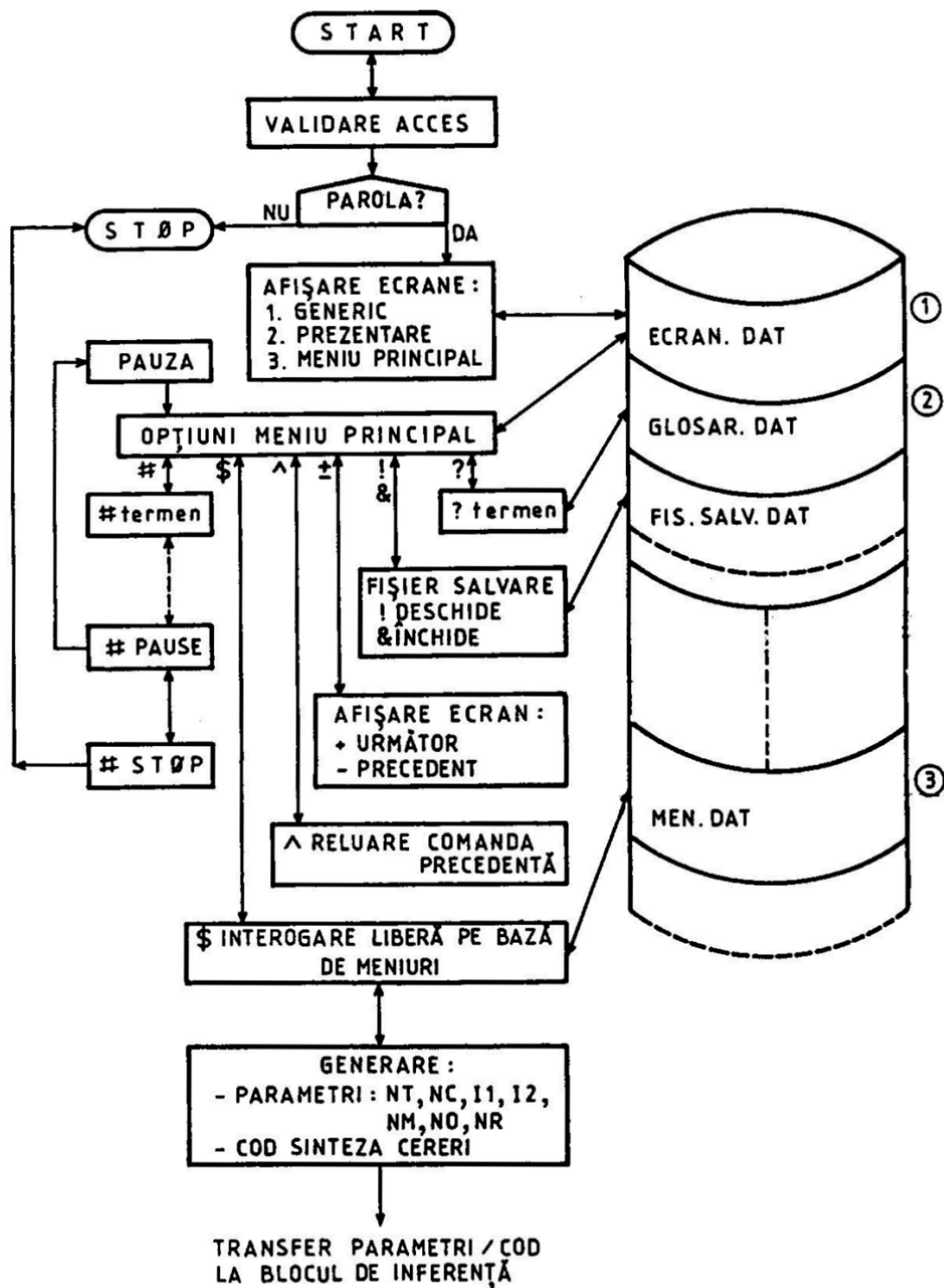
RASPUNSUL Dvs. (Nr.GRUPA + Specificatii):

SPECIFICATI DESCRIEREA TIPULUI DE ANALIZA (max. 1014):

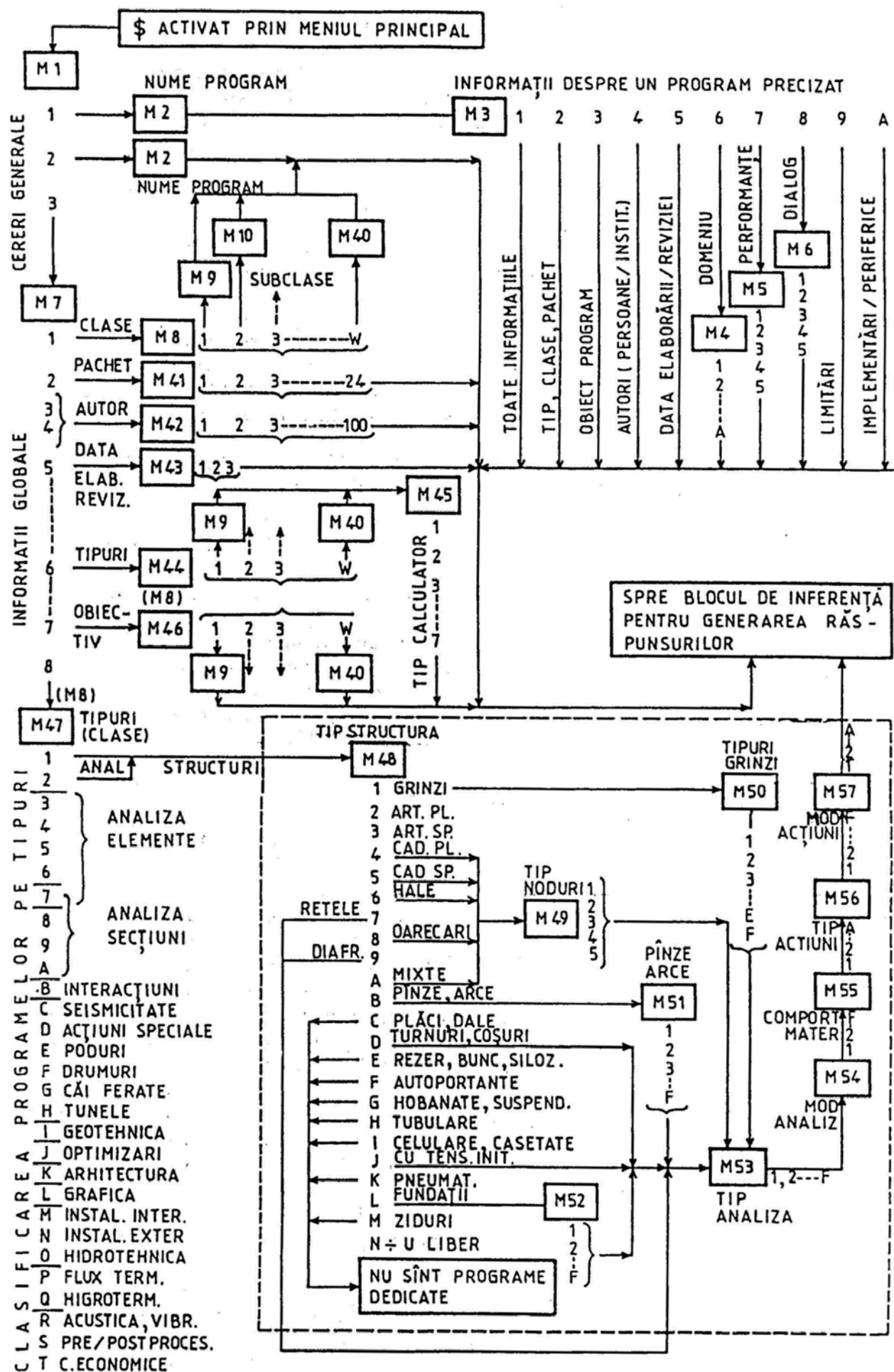
G R U P A 1			G R U P A 2		
1	LINII INFLUEN.	16 PLANA	1		16
2	LA STABILITATE	17 SPATIALA	2		17
3	STATICA	18 MODALA	3		18
4	SEISMICA	19 LA OBOSEALA	4		19
5	DINAMICA	20 HISTERETICA	5		20
6	BIOGRAFICA	21 LOCALA	6		21
7	LINIARA	22 GLOBALA	7		22
8	ELASTICA	23 TIME HISTORY	8		23
9	PLASTICA	24 PE BAZA DE	9		24
10	FIZIC	25 SPECTRE	10		25
11	GEOMETRIC	26 ACCELEROGRAMA	11		26
12	NELINIARA	27 NORMATIV P-100	12		27
13	POST ELASTICA	28 ELEMENT FINIT	13		28
14	DE ORDINUL 2	29 INTEGRARE	14		29
15	DE ORDINUL 3	30 PAS CU PAS	15		30

RASPUNSUL Dvs. (Nr.GRUPA + Specificatii):

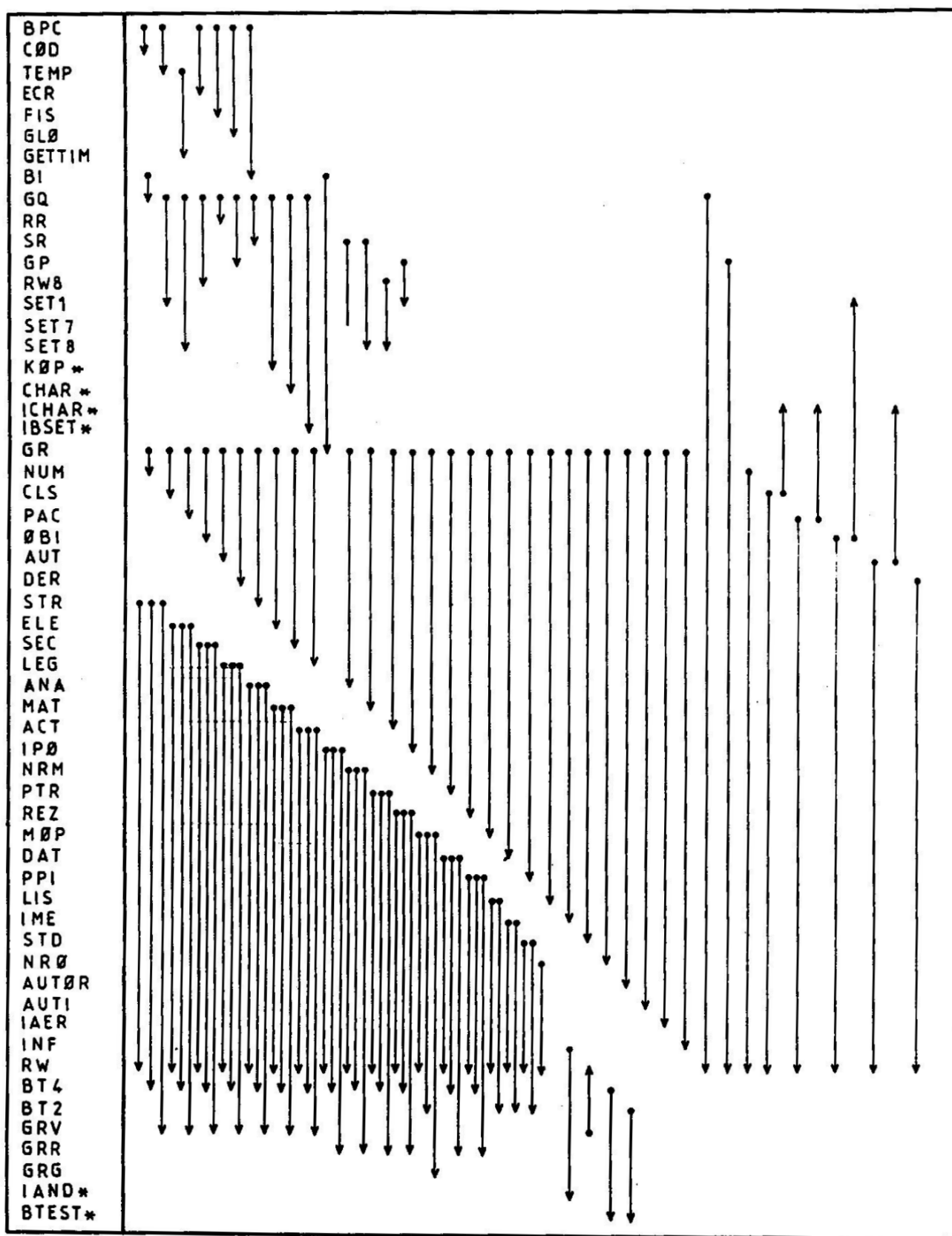
* URMATOARELE PROGRAME RASPUND PROBABIL OPTIMAL CERERILOR (OPTIUNILOR) Dvs.: *



5. Funcțiile și structura blocului de interfață a interpretorului de comenzi din sistemul expert BPC



6. Fluxul meniurilor de interogare



7. Fluxul apelurilor de subprograme în sistemul expert BPC

ANEXA 1 |

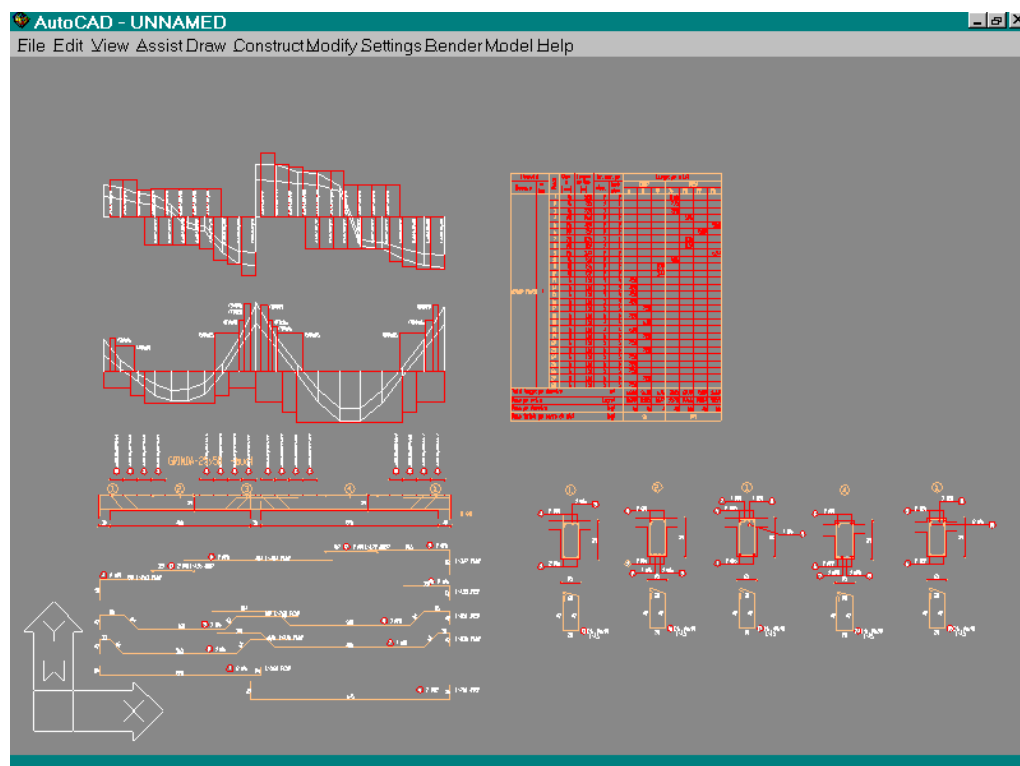
8. Formularul proiectat pentru completarea bazei de cunoștințe a sistemului expert BPC (conținând grila tip I.P.C.T. de clasificare a programelor).

INSTRUMENTE CONCEPTE ÎN CADRUL FIRMEI CONSORT CONSTRUCȚII S.R.L.

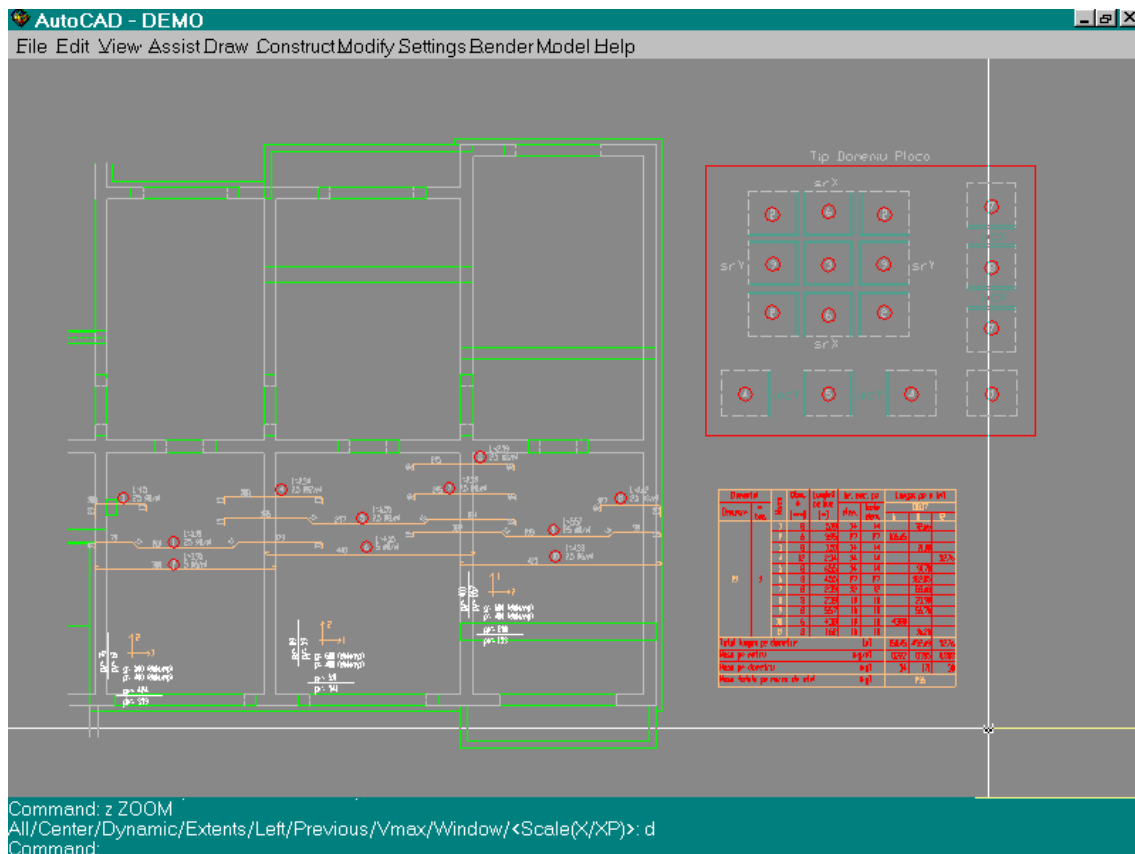
A. Capturi de ecran

1. Exemplu de utilizare a instrumentului ARMGR (modul AutoLISP, dezvoltat inițial pentru AutoCAD 12, destinat dimensionării grinzilor continue din beton armat cu secțiune dreptunghiulară, generând automat desenul de armare pentru execuție).

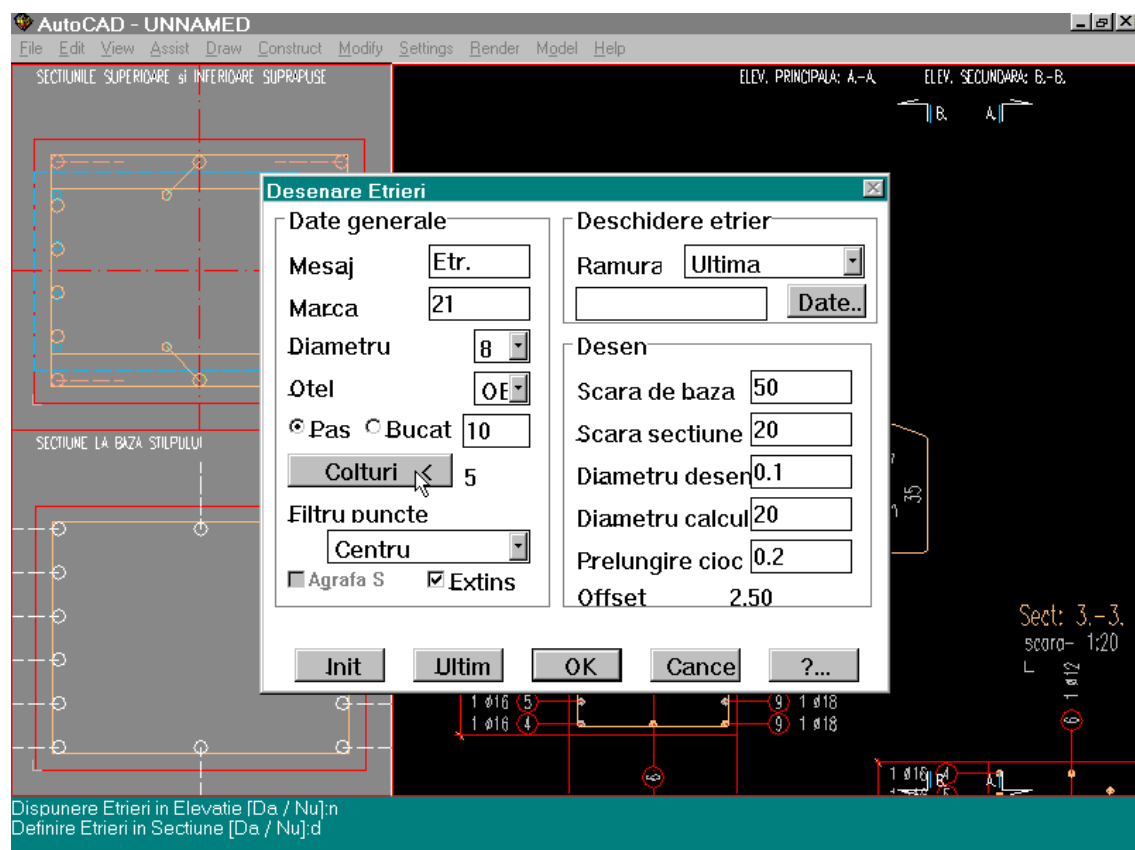
1.1. Selectarea parametrilor pentru dimensionarea și armarea unei grinzi continue.



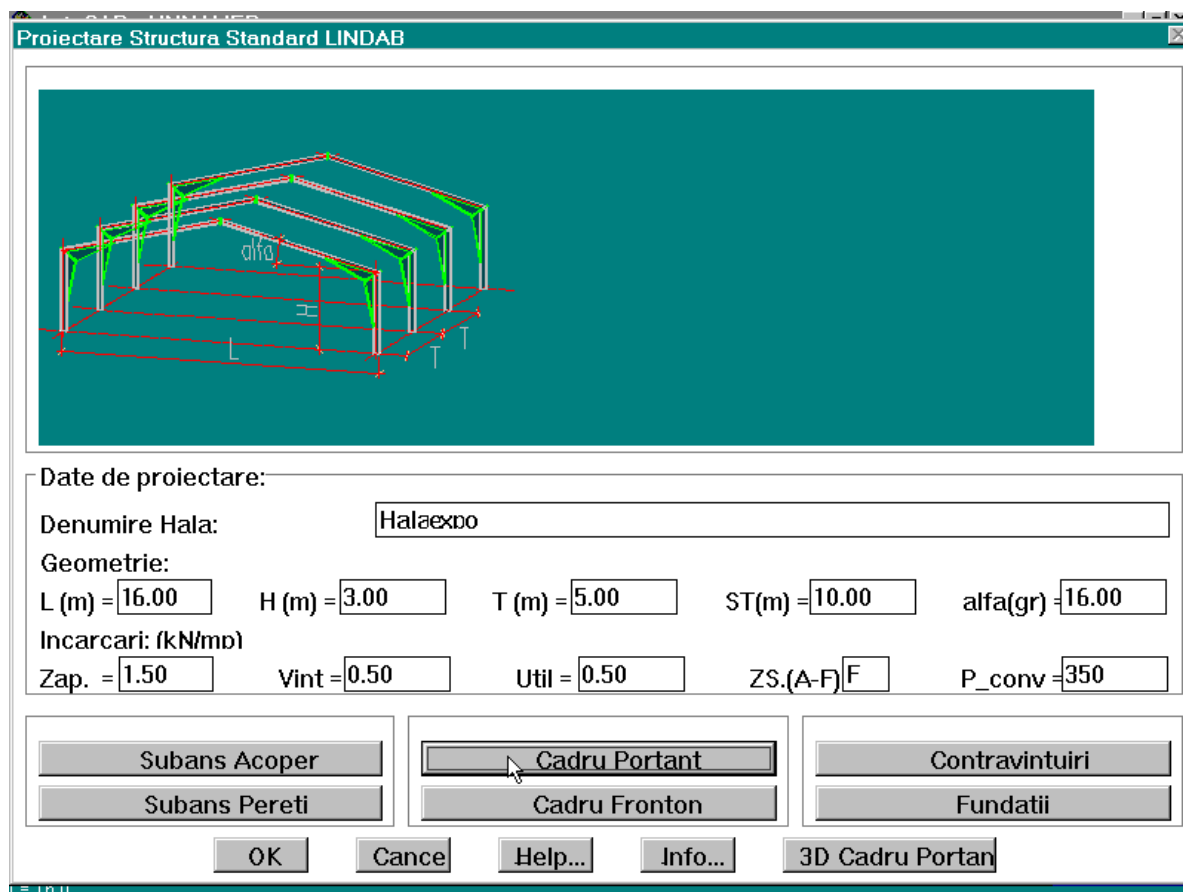
1.2. Desenul de armare (generat în urma rulării).



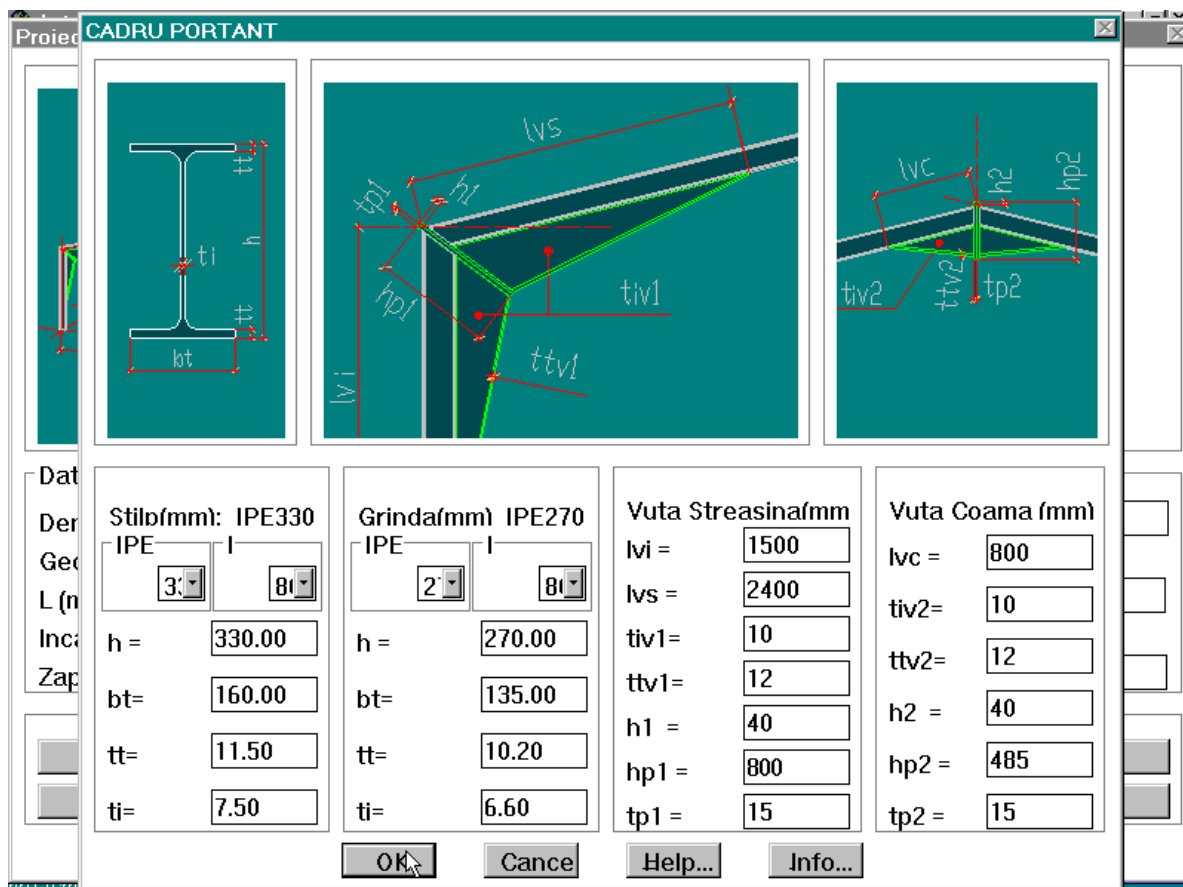
2. Ecran exemplificator din utilizarea modulului AutoLISP destinat dimensionării și armării planșelor din beton armat, sub AutoCAD 12.

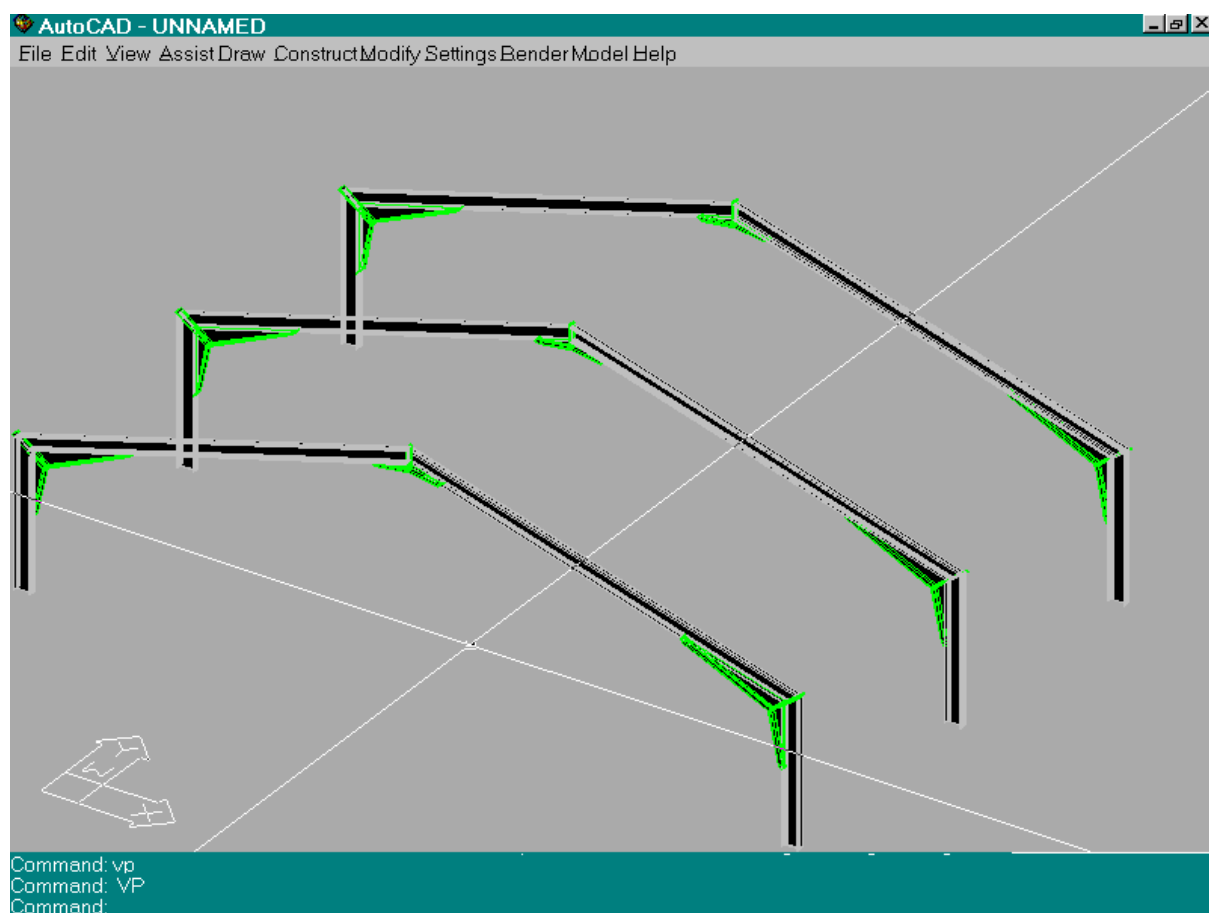


3. Ecran exemplificator din utilizarea modulului AutoLISP destinat dimensionării și armării stâlpilor din beton armat, cu secțiune dreptunghiulară (tot sub AutoCAD 12).

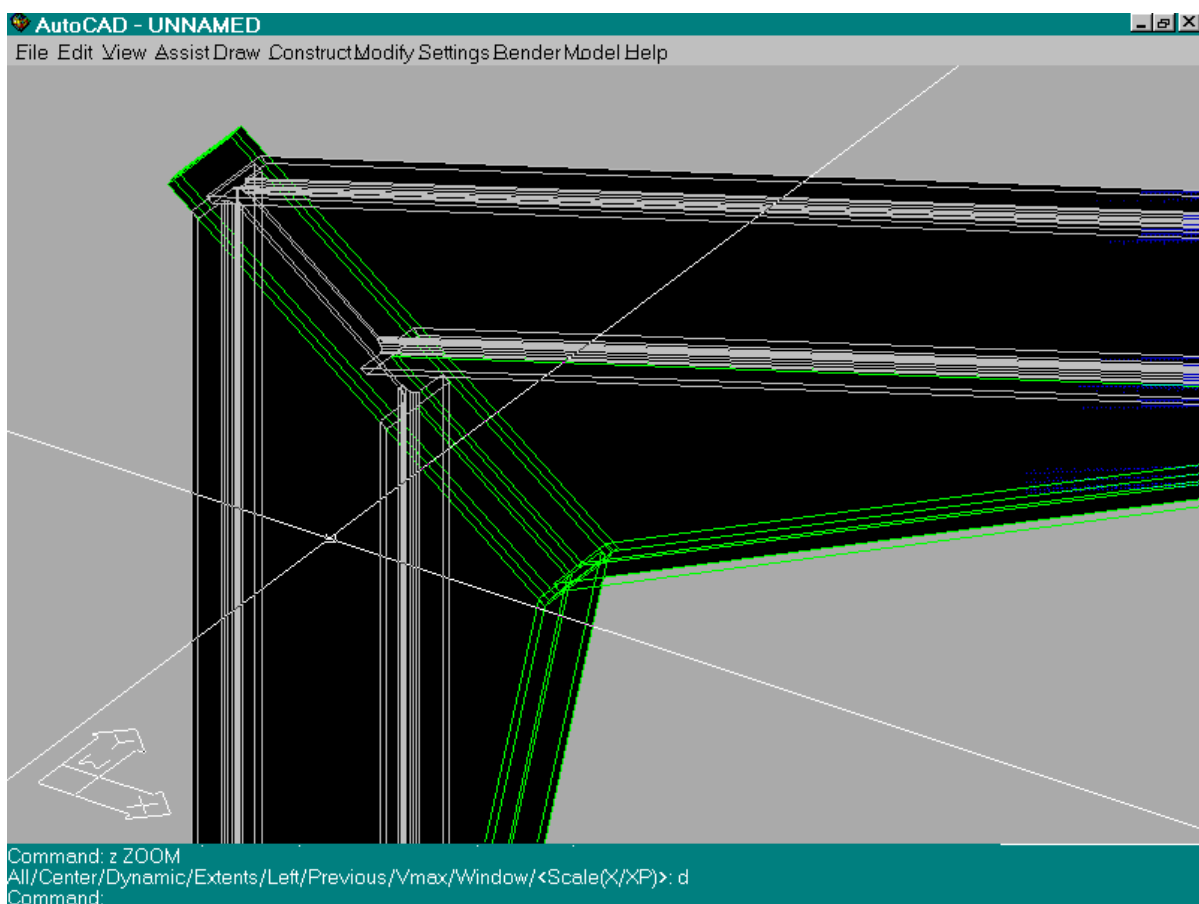


4. Capturi de ecran ilustrative, cu alegerea parametrilor din programul destinat generării virtuale 3D a halelor standard cu structură metalică, marca LINDAB. Prototipul a fost realizat în AutoLISP și (deși nu a fost complet finalizat) a fost implementat sub AutoCAD 12.





5. Vederi axonometrice: structura de rezistență a halei generate automat pe baza parametrilor aleși (imaginea de sus) și detaliu de nod pentru ilustrarea elementelor generate (imaginea de jos).



B. Fișiere sursă

În cele ce urmează, voi prezenta sursele realizate pentru ecranele de dialog (în DCL) și pentru aplicația propriu zisă (în AutoLISP) în varianta concepută pentru AutoCAD 12 a prototipului destinat generării halelor metalice standard (marca LINDAB). Așa cum am menționat, sistemul a fost implementat doar la nivel de prototip. Sursele prezentate reflectă stadiul dezvoltării atinse prin implicarea mea directă în realizarea acestui instrument.

Conținutul fișierului sursă LINDAB.DCL (pentru descierea ecranelor de dialog)

```
dcl_settings : default_dcl_settings { audit_level = 0; }
lindab : dialog {
/* latime error_tile 57 */
label = "Proiectare Structura Standard LINDAB";
// : text { label= "Model"; key = "i_text"; vertical_margin = none; }
: column {
: boxed_column {
: image {
key = "i_t";
width = 80; //95
height = 13; //20
fixed_width = true; //!
color = 0; /* 147 */
}
}
: boxed_column {
label = "Date de proiectare:";
: column {
: row {
: text { label= "Denumire Hala: "; }
: edit_box { label = ""; key = "den_hala_t"; edit_width = 60; edit_limit = 100; }
}
: text { label= "Geometrie: "; }
: row {
: edit_box { label = "L (m) ="; key = "deschid_t"; edit_width = 5; edit_limit = 100; fixed_width = true; }
: edit_box { label = "H (m) ="; key = "inalt_t"; edit_width = 5; edit_limit = 100; fixed_width = true; }
: edit_box { label = "T (m) ="; key = "travee_t"; edit_width = 5; edit_limit = 100; fixed_width = true; }
: edit_box { label = "ST(m) ="; key = "sum_travee_t"; edit_width = 5; edit_limit = 100; fixed_width = true; }
: edit_box { label = "alfa(gr) ="; key = "alfa_t"; edit_width = 5; edit_limit = 100; fixed_width = true; }
}
: text { label= "Incarcari: (kN/mp)"; }
: row {
: edit_box { label = "Zap. ="; key = "zapada_t"; edit_width = 5; edit_limit = 100; fixed_width = true; }
: edit_box { label = "Vint ="; key = "vint_t"; edit_width = 5; edit_limit = 100; fixed_width = true; }
: edit_box { label = "Util ="; key = "utiltehn_t"; edit_width = 5; edit_limit = 100; fixed_width = true; }
: edit_box { label = "ZS(A-F):"; key = "zona_seism_t"; edit_width = 2; edit_limit = 2; fixed_width = true; }
: edit_box { label = "P_conv. ="; key = "p_conv_t"; edit_width = 5; edit_limit = 100; fixed_width = true; }
}
}
}
```

```

    }
    : row {
    : boxed_column {
    : button { label = "Subans. Acoperis"; width = 25; key = "subans_acoper_t"; fixed_width = true; is_default = true; }
    : button { label = "Subans. Pereti"; width = 25; key = "subans_pereti_t"; fixed_width = true; is_default = true; }
    }
    : boxed_column {
    : button { label = "Cadru Portant"; width = 25; key = "cadru_portant_t"; fixed_width = true; is_default = true; }
    : button { label = "Cadru Fronton"; width = 25; key = "cadru_fronton_t"; fixed_width = true; is_default = true; }
    }
    : boxed_column {
    : button { label = "Contravantuirii"; width = 25; key = "contrav_t"; fixed_width = true; is_default = true; }
    : button { label = "Fundatii"; width = 25; key = "fundatii_t"; fixed_width = true; is_default = true; }
    }
    }
  } //row
  // : text { key = "error"; width = 63; is_error_tile = true; }
  : ok_cancel_help_info
  : column {
    : row {
    fixed_width = true;
    alignment = centered;
    ok_button;
    : spacer { width = 2; }
    cancel_button;
    : spacer { width = 2; }
    help_button;
    : spacer { width = 2; }
    info_button;
    : spacer { width = 2; }
    : button { label = "3D Cadru Portant <"; width = 20; key = "3d_cadru_p_t"; }
    }
  }
} //dialog
cadru_p : dialog {
label = "CADRU PORTANT";
: column {
: row {
: boxed_column { label = "Stalp";
: image {
key = "i_cp";
width = 15; //50
height = 13; //17
fixed_width = true; //!
color = 0; /* 147 */
}
}
: boxed_column { label = "Grinda";
: image {
key = "i_v1";
width = 40; //50
height = 13; //17
fixed_width = true; //!

```

```

        color = 0; /* 147 */
    }
}
: boxed_column { label = "Vute";
: image {
    key = "i_v2";
    width = 22; //50
    height = 13; //17
    fixed_width = true; //!
    color = 0; /* 147 */
}
}
} //row
: row {
    : boxed_column {
        : row {
            : text { label = "Stilp(mm): "; }
            : text { label = ""; key = "tip_st_t"; width = 5; }
        }
        : row {
            : column {
                label = "IPE";
                : popup_list {
                    key = "profs_ipe_t";
                    edit_width = 4;
                    list = " ";
                }
            }
            : column {
                label = "I";
                : popup_list {
                    key = "profs_i_t";
                    edit_width = 4;
                    list = " ";
                }
            }
        }
        : edit_box { label = "h ="; key = "hs_t"; edit_width = 6; edit_limit = 100; }
        : edit_box { label = "bt="; key = "bts_t"; edit_width = 6; edit_limit = 100; }
        : edit_box { label = "tt="; key = "tts_t"; edit_width = 6; edit_limit = 100; }
        : edit_box { label = "ti="; key = "tis_t"; edit_width = 6; edit_limit = 100; }
    }
    : boxed_column {
        : row {
            : text { label = "Grinda(mm): "; }
            : text { label = ""; key = "tip_gr_t"; width = 6; }
        }
        : row {
            : column {
                label = "IPE";
                : popup_list {
                    key = "profg_ipe_t";
                    edit_width = 4;

```

```

list = "";
}
}
: column {
label = "I";
: popup_list {
key = "profg_i_t";
edit_width = 4;
list = "";
}
}
}
: edit_box { label = "h ="; key = "hg_t"; edit_width = 6; edit_limit = 100; }
: edit_box { label = "bt="; key = "btg_t"; edit_width = 6; edit_limit = 100; }
: edit_box { label = "tt="; key = "ttg_t"; edit_width = 6; edit_limit = 100; }
: edit_box { label = "ti="; key = "tig_t"; edit_width = 6; edit_limit = 100; }
}
: boxed_column {
: text { label= "Vuta Streasina(mm): "; }
: edit_box { label = "lvi ="; key = "lvi_t"; edit_width = 6; edit_limit = 100; }
: edit_box { label = "lvs="; key = "lvs_t"; edit_width = 6; edit_limit = 100; }
: edit_box { label = "tiv1="; key = "tiv1_t"; edit_width = 6; edit_limit = 100; }
: edit_box { label = "ttv1="; key = "ttv1_t"; edit_width = 6; edit_limit = 100; }
: edit_box { label = "h1 ="; key = "h1_t"; edit_width = 6; edit_limit = 100; }
: edit_box { label = "hp1 ="; key = "hp1_t"; edit_width = 6; edit_limit = 100; }
: edit_box { label = "tp1 ="; key = "tp1_t"; edit_width = 6; edit_limit = 100; }
}
: boxed_column {
: text { label= "Vuta Coama (mm): "; }
: edit_box { label = "lvc="; key = "lvc_t"; edit_width = 6; edit_limit = 100; }
: edit_box { label = "tiv2="; key = "tiv2_t"; edit_width = 6; edit_limit = 100; }
: edit_box { label = "ttv2="; key = "ttv2_t"; edit_width = 6; edit_limit = 100; }
: edit_box { label = "h2 ="; key = "h2_t"; edit_width = 6; edit_limit = 100; }
: edit_box { label = "hp2 ="; key = "hp2_t"; edit_width = 6; edit_limit = 100; }
: edit_box { label = "tp2 ="; key = "tp2_t"; edit_width = 6; edit_limit = 100; }
}
} //row
} //col
: ok_cancel_help_info : column {
: row {
fixed_width = true;
alignment = centered;
ok_button;
: spacer { width = 2; }
cancel_button;
: spacer { width = 2; }
help_button;
: spacer { width = 2; }
info_button;
}
}
}

```

Conținutul fișierului sursă LINDAB.LSP (pentru generarea structurii de hală)

```

(defun *merr* (msg)
  (setq *error* m:err m:err nil)
  (princ))
(defun *merrmsg* (msg)
  (princ msg)
  (setq *error* m:err m:err nil)
  (princ))
(defun lindab(/ id_c)
  (setvar "osmode" 0)
  (new_dialog "lindab" (setq id_c (load_dialog "c:/ascad/lindab/lindab"))))
  (putimage_c "i_t" "c:/ascad/lindab/lindabo2" 0 0)
  (set_tile "den_hala_t" "Hala test")
  (set_tile "deschid_t" "12.00")
  (set_tile "inalt_t" "3.00")
  (set_tile "travee_t" "5.00")
  (set_tile "sum_travee_t" "50.00")
  (set_tile "alfa_t" "16.00")
  (set_tile "zapada_t" "1.50")
  (set_tile "vint_t" "0.50")
  (set_tile "utiltehn_t" "0.50")
  (set_tile "zona_seism_t" "F")
  (set_tile "p_conv_t" "350")
  (princ "\n      ")
  (setq den_hala_c (get_tile "den_hala_t"))
  (princ "\nDenumire Hala: ") (princ den_hala_c)
  (setq deschid_c (atof (get_tile "deschid_t")))
  (princ "\nL= ") (princ deschid_c)
  (setq inalt_c (atof (get_tile "inalt_t")))
  (princ "\nH= ") (princ inalt_c)
  (setq travee_c (atof (get_tile "travee_t")))
  (princ "\nT= ") (princ travee_c)
  (setq sum_travee_c (atof (get_tile "sum_travee_t")))
  (princ "\nST= ") (princ sum_travee_c)
  (setq alfa_c (atof (get_tile "alfa_t")))
  (princ "\nalfa= ") (princ alfa_c)
  (setq zapada_c (atof (get_tile "zapada_t")))
  (princ "\nzap= ") (princ zapada_c)
  (setq vint_c (atof (get_tile "vint_t")))
  (princ "\nvint= ") (princ vint_c)
  (setq utiltehn_c (atof (get_tile "utiltehn_t")))
  (princ "\nutiltehn= ") (princ utiltehn_c)
  (setq zona_seism_c (get_tile "zona_seism_t"))
  (princ "\nZSeism= ") (princ zona_seism_c)
  (setq p_conv_c (atof (get_tile "p_conv_t")))
  (princ "\np_conv= ") (princ p_conv_c)
  (action_tile "den_hala_t" "(den_hala)")
  (action_tile "deschid_t" "(deschid)")
  (action_tile "inalt_t" "(inalt)")
  (action_tile "travee_t" "(travee)")
  (action_tile "sum_travee_t" "(sum_travee)")
  (action_tile "alfa_t" "(alfa)")

```

```

(action_tile "zapada_t" "(zapada)")
(action_tile "vint_t" "(vint)")
(action_tile "utiltehn_t" "(utiltehn)")
(action_tile "zona_seism_t" "(zona_seism)")
(action_tile "p_conv_t" "(p_conv)")
(action_tile "subans_acoper_t" "(subans_acoper)")
(action_tile "subans_pereti_t" "(subans_pereti)")
(action_tile "cadru_portant_t" "(cadru_portant)")
(action_tile "cadru_fronton_t" "(cadru_fronton)")
(action_tile "contrav_t" "(contrav)")
(action_tile "fundatii_t" "(fundatii)")
(action_tile "accept" "(dlg_c)")
(start_dialog)
(unload_dialog id_c))
(defun dlg_c()
  (done_dialog) ;;OK!)
(defun den_hala()
  (setq den_hala_c (get_tile "den_hala_t"))
  (princ "\nDenumire hala: ") (princ den_hala_c))
(defun deschid()
  (setq deschid_c (atoi (get_tile "deschid_t")))
  (princ "\nL= ") (princ deschid_c))
(defun inalt()
  (setq inalt_c (atoi (get_tile "inalt_t")))
  (princ "\nH= ") (princ inalt_c))
(defun travee()
  (setq travee_c (atoi (get_tile "travee_t")))
  (princ "\nT= ") (princ travee_c))
(defun sum_travee()
  (setq sum_travee_c (atoi (get_tile "sum_travee_t")))
  (princ "\nST= ") (princ sum_travee_c))
(defun alfa()
  (setq alfa_c (atoi (get_tile "alfa_t")))
  (princ "\nalfa= ") (princ alfa_c))
(defun zapada()
  (setq zapada_c (atoi (get_tile "zapada_t")))
  (princ "\nzap= ") (princ zapada_c))
(defun vint()
  (setq vint_c (atoi (get_tile "vint_t")))
  (princ "\nvint= ") (princ vint_c))
(defun utiltehn()
  (setq utiltehn_c (atoi (get_tile "utiltehn_t")))
  (princ "\nutiltehn= ") (princ utiltehn_c))
(defun zona_seism()
  (setq zona_seism_c (get_tile "zona_seism_t"))
  (princ "\nzona_seism= ") (princ zona_seism_c))
(defun p_conv()
  (setq p_conv_c (atoi (get_tile "p_conv_t")))
  (princ "\np_conv= ") (princ p_conv_c))
(defun subans_acoper()
  (princ "\nprogram: subans_acoper"))
(defun subans_pereti()
  (princ "\nprogram: subans_pereti"))

```

```

(defun cadru_fronton()
  (princ"\nprogram: cadru_fronton"))
(defun contrav()
  (princ"\nprogram: contrav"))
(defun fundatii()
  (princ"\nprogram: fundatii"))
(defun tipstgr()
  (set_tile "tip_st_t" (nth 0 lista_st))
  (set_tile "tip_gr_t" (nth 0 lista_gr))
  (setq tip_st_c (get_tile "tip_st_t"))
  (setq tip_gr_c (get_tile "tip_gr_t"))
  (set_tile "hs_t" (rtos (* (nth 1 lista_st) 10.00) 2 2)) ;mm
  (set_tile "bts_t" (rtos (* (nth 2 lista_st) 10.00) 2 2)) ;mm
  (set_tile "tts_t" (rtos (* (nth 3 lista_st) 10.00) 2 2)) ;mm
  (set_tile "tis_t" (rtos (* (nth 4 lista_st) 10.00) 2 2)) ;mm
  (set_tile "hg_t" (rtos (* (nth 1 lista_gr) 10.00) 2 2)) ;mm
  (set_tile "btg_t" (rtos (* (nth 2 lista_gr) 10.00) 2 2)) ;mm
  (set_tile "ttg_t" (rtos (* (nth 3 lista_gr) 10.00) 2 2)) ;mm
  (set_tile "tig_t" (rtos (* (nth 4 lista_gr) 10.00) 2 2)) ;mm
  (setq hs_c (/ (atof (get_tile "hs_t")) 10.00)) ;cm
  (princ "\nhs (cm)= ") (princ hs_c)
  (setq bts_c (/ (atof (get_tile "bts_t")) 10.00))
  (princ "\nbts (cm)= ") (princ bts_c)
  (setq tts_c (/ (atof (get_tile "tts_t")) 10.00))
  (princ "\ntts (cm)= ") (princ tts_c)
  (setq tis_c (/ (atof (get_tile "tis_t")) 10.00))
  (princ "\ntis (cm)= ") (princ tis_c)
  (setq hg_c (/ (atof (get_tile "hg_t")) 10.00))
  (princ "\nhg (cm)= ") (princ hg_c)
  (setq btg_c (/ (atof (get_tile "btg_t")) 10.00))
  (princ "\nbtg (cm)= ") (princ btg_c)
  (setq ttg_c (/ (atof (get_tile "ttg_t")) 10.00))
  (princ "\nttg (cm)= ") (princ ttg_c)
  (setq tig_c (/ (atof (get_tile "tig_t")) 10.00))
  (princ "\ntig (cm)= ") (princ tig_c))
(defun cadru_portant()
  (setq lista_st '("IPE80" 8 4.6 0.52 0.38 0.5 7.6434 0.700 80.1376 8.48903 1.09233))
  (setq lista_gr '("IPE80" 8 4.6 0.52 0.38 0.5 7.6434 0.700 80.1376 8.48903 1.09233))
  (setq lprofs_ipe nil)
  (setq lprofs_i nil)
  (setq lprofg_ipe nil)
  (setq lprofg_i nil)
  (princ"\nprogram: cadru_portant")
  (new_dialog "cadru_p" (setq id_cp (load_dialog "c:/ascad/lindab/lindab")))
  (putimage_c "i_cp" "c:/ascad/lindab/sect_cp" 0 0)
  (putimage_c "i_v1" "c:/ascad/lindab/vuta1_cp" 0 0)
  (putimage_c "i_v2" "c:/ascad/lindab/vuta2_cp" 0 0)
  ;;consider IPE80
  (set_tile "tip_st_t" (nth 0 lista_st))
  (set_tile "tip_gr_t" (nth 0 lista_gr))
  (setq tip_st_c (get_tile "tip_st_t"))
  (setq tip_gr_c (get_tile "tip_gr_t"))
  ;initial >>>> IPE80..st..gr...

```

```

(set_tile "hs_t" "80") ;mm
(set_tile "bts_t" "46")
(set_tile "tts_t" "5.2")
(set_tile "tis_t" "3.8")
(set_tile "hg_t" "80")
(set_tile "btg_t" "46")
(set_tile "ttg_t" "5.2")
(set_tile "tig_t" "3.8")
(set_tile "lvi_t" "1500")
(set_tile "lvs_t" "2400")
(set_tile "h1_t" "50")
(set_tile "hp1_t" "840")
(set_tile "tp1_t" "15")
(set_tile "tiv1_t" "10")
(set_tile "ttv1_t" "12")
(set_tile "lvc_t" "800")
(set_tile "h2_t" "50")
(set_tile "hp2_t" "600")
(set_tile "tp2_t" "15")
(set_tile "tiv2_t" "10")
(set_tile "ttv2_t" "12")
(setq tip_st_c (get_tile "tip_st_t"))
(princ "\nTip_st: ") (princ tip_st_c)
(setq tip_gr_c (get_tile "tip_gr_t"))
(princ "\nTip_gr: ") (princ tip_gr_c)
(setq hs_c (/ (atof (get_tile "hs_t")) 10.0)) ;cm
(princ "\nhs (cm)= ") (princ hs_c)
(setq bts_c (/ (atof (get_tile "bts_t")) 10.0))
(princ "\nbts (cm)= ") (princ bts_c)
(setq tts_c (/ (atof (get_tile "tts_t")) 10.0))
(princ "\ntts (cm)= ") (princ tts_c)
(setq tis_c (/ (atof (get_tile "tis_t")) 10.0))
(princ "\ntis (cm)= ") (princ tis_c)
(setq hg_c (/ (atof (get_tile "hg_t")) 10.0))
(princ "\nhg (cm)= ") (princ hg_c)
(setq btg_c (/ (atof (get_tile "btg_t")) 10.0))
(princ "\nbtg (cm)= ") (princ btg_c)
(setq ttg_c (/ (atof (get_tile "ttg_t")) 10.0))
(princ "\nttg (cm)= ") (princ ttg_c)
(setq tig_c (/ (atof (get_tile "tig_t")) 10.0))
(princ "\ntig (cm)= ") (princ tig_c)
(setq lvi_c (/ (atof (get_tile "lvi_t")) 10.0)) ;cm
(princ "\nlvi (cm)= ") (princ lvi_c)
(setq lvs_c (/ (atof (get_tile "lvs_t")) 10.0)) ;cm
(princ "\nlvs (cm)= ") (princ lvs_c)
(setq h1_c (/ (atof (get_tile "h1_t")) 10.0)) ;cm
(princ "\nh1 (cm)= ") (princ h1_c)
(setq hp1_c (/ (atof (get_tile "hp1_t")) 10.0)) ;cm
(princ "\nhp1 (cm)= ") (princ hp1_c)
(setq tp1_c (/ (atof (get_tile "tp1_t")) 10.0)) ;cm
(princ "\ntp1 (cm)= ") (princ tp1_c)
(setq tiv1_c (/ (atof (get_tile "tiv1_t")) 10.0)) ;cm
(princ "\ntiv1 (cm)= ") (princ tiv1_c)

```

```

(setq ttv1_c (/ (atof (get_tile "ttv1_t")) 10.0)) ;cm
(princ "\nttv1 (cm)= " (princ ttv1_c)
(setq lvc_c (/ (atof (get_tile "lvc_t")) 10.0)) ;cm
(princ "\nlvc (cm)= " (princ lvc_c)
(setq h2_c (/ (atof (get_tile "h2_t")) 10.0)) ;cm
(princ "\nh2 (cm)= " (princ h2_c)
(setq hp2_c (/ (atof (get_tile "hp2_t")) 10.0)) ;cm
(princ "\nhp2 (cm)= " (princ hp2_c)
(setq tp2_c (/ (atof (get_tile "tp2_t")) 10.0)) ;cm
(princ "\ntp2 (cm)= " (princ tp2_c)
(setq tiv2_c (/ (atof (get_tile "tiv2_t")) 10.0)) ;cm
(princ "\ntiv2 (cm)= " (princ tiv2_c)
(setq ttv2_c (/ (atof (get_tile "ttv2_t")) 10.0)) ;cm
(princ "\nttv2 (cm)= " (princ ttv2_c)
(setq lipe '(
;Nume_ h b tt i r A lz ly lx Wz
;Nume d b t i r A lz ly lx Wz
; cmcm cm cm cm cm cm**2 cm**4 cm**4 cm**4 cm**3
("IPE80" 8 4.6 0.52 0.38 0.5 7.6434 0.700 80.1376 8.48903 1.09233)
("IPE100" 10 5.5 0.57 0.41 0.7 10.3232 1.100 171.012 15.9187 1.57140)
("IPE120" 12 6.4 0.63 0.44 0.7 13.2102 1.710 317.753 27.6682 2.20600)
("IPE140" 14 7.3 0.69 0.47 0.7 16.4260 2.540 541.224 44.9178 2.98500)
("IPE160" 16 8.2 0.74 0.50 0.9 20.0913 3.530 869.293 68.3145 3.85300)
("IPE180" 18 9.1 0.80 0.53 0.9 23.9473 4.900 1316.96 100.850 4.94960)
("IPE200" 20 10.0 0.85 0.56 1.2 28.4841 6.460 1943.17 142.368 6.13600)
("IPE220" 22 11.0 0.92 0.59 1.2 33.3705 8.860 2771.84 204.886 7.77600)
("IPE240" 24 12.0 0.98 0.62 1.5 39.1162 11.60 3891.63 283.634 9.54900)
("IPE270" 27 13.5 1.02 0.66 1.5 45.9450 14.93 5789.78 419.869 11.8000)
("IPE300" 30 15.0 1.07 0.71 1.5 53.8120 19.47 8356.11 603.779 14.6700)
("IPE330" 33 16.0 1.15 0.75 1.8 62.6062 25.70 11766.9 788.143 18.0000)
("IPE360" 36 17.0 1.27 0.80 1.8 72.7292 36.20 16265.6 1043.45 22.9500)
("IPE400" 40 18.0 1.35 0.86 2.1 84.4636 46.80 23128.4 1317.82 27.9400)
("IPE450" 45 19.0 1.46 0.94 2.1 98.8208 63.80 33742.9 1675.86 35.2600)
("IPE500" 50 20.0 1.60 1.02 2.1 115.522 89.00 48198.5 2141.69 44.8300)
("IPE550" 55 21.0 1.72 1.11 2.4 134.416 118.4 67116.5 2667.58 55.5400)
("IPE600" 60 22.0 1.90 1.20 2.4 155.984 166.2 92083.4 3387.34 70.5600)
))
;;; stilp ipe
(start_list "profs_ipe_t")
(mapcar 'add_list (mapcar '(lambda (x) (substr (car x) 4)) lipe))
(end_list)
(action_tile "profs_ipe_t"
"(progn
(setq lprofs_ipe (nth (atoi $value) lipe))
(setq lista_st lprofs_ipe)
(tipstgr) )"
)
;;; grinda ipe
(start_list "profg_ipe_t")
(mapcar 'add_list (mapcar '(lambda (x) (substr (car x) 4)) lipe))
(end_list)
(action_tile "profg_ipe_t"
"(progn

```

```

(setq lprofgr_ipe (nth (atoi $value) lipe))
(setq lista_gr lprofgr_ipe)
(tipstgr))"
)
(setq li '(
;Nume h b t d=R r A ly lx
; cmcm cm cm cm cm**2 cm**4 cm**4
("180" 8 4.20 0.59 0.39 0.23 7.580 6.2900 77.8000)
("1100" 10 5.00 0.68 0.45 0.27 10.60 12.200 171.000)
("1120" 12 5.80 0.77 0.51 0.31 14.20 21.500 328.000)
("1140" 14 6.60 0.86 0.57 0.34 18.30 36.200 573.000)
("1160" 16 7.40 0.95 0.63 0.38 22.80 54.700 935.000)
("1180" 18 8.20 1.04 0.69 0.41 27.90 81.300 1450.00)
("1200" 20 9.00 1.13 0.75 0.45 33.50 117.00 2140.00)
("1240" 24 10.6 1.31 0.87 0.52 46.10 221.00 4250.00)
("1300" 30 12.5 1.62 1.08 0.65 69.10 451.00 9800.00)
("1400" 40 15.5 2.16 1.44 0.86 118.0 1160.0 29210.0)
))
;;;stilp i
(start_list "profs_i_t")
(mapcar 'add_list (mapcar '(lambda (x) (substr (car x) 2)) li))
(end_list)
(action_tile "profs_i_t"
"(progn
(setq lprofs_i (nth (atoi $value) li))
(setq lista_st lprofs_i)
(tipstgr))"
)
;;;grinda i
(start_list "profg_i_t")
(mapcar 'add_list (mapcar '(lambda (x) (substr (car x) 2)) li))
(end_list)
(action_tile "profg_i_t"
"(progn
(setq lprofg_i (nth (atoi $value) li))
(setq lista_gr lprofg_i)
(tipstgr))"
)
)
(action_tile "hs_t" "(hs)")
(action_tile "bts_t" "(bts)")
(action_tile "tts_t" "(tts)")
(action_tile "tis_t" "(tis)");st
(action_tile "hg_t" "(hg)")
(action_tile "btg_t" "(btg)")
(action_tile "ttg_t" "(ttg)")
(action_tile "tig_t" "(tig)");gr
(action_tile "lvi_t" "(lvi)")
(action_tile "lvs_t" "(lvs)")
(action_tile "h1_t" "(h1)")
(action_tile "hp1_t" "(hp1)")
(action_tile "tp1_t" "(tp1)")
(action_tile "tiv1_t" "(tiv1)")
(action_tile "ttv1_t" "(ttv1)")

```

```

(action_tile "lvc_t" "(lvc)")
(action_tile "h2_t" "(h2)")
(action_tile "hp2_t" "(hp2)")
(action_tile "tp2_t" "(tp2)")
(action_tile "tiv2_t" "(tiv2)")
(action_tile "ttv2_t" "(ttv2)")
(start_dialog)
; (unload_dialog id_cp)
(princ)
)

;cm "hs" "st"/"gr"
(defun fuser(ttt tipu)
(cond
((/= (eval (read (strcat ttt "_c")))) (/ (atof (get_tile (strcat ttt "_t"))) 10.00))
(set_tile (strcat "tip_" tipu "_t") "USER")
(set (read (strcat "tip_" tipu "_c")) (get_tile (strcat "tip_" tipu "_t")))
(set (read (strcat ttt "_c")) (/ (atof (get_tile (strcat ttt "_t"))) 10.0))
(princ (strcat "\n" ttt " (cm)="))
(princ (eval (read (strcat ttt "_c")))))
(t (princ "\n-valoare neschimbata"))))
(defun hs() (fuser "hs" "st"))
(defun bts() (fuser "bts" "st"))
(defun tts() (fuser "tts" "st"))
(defun tis() (fuser "tis" "st"))
(defun hg() (fuser "hg" "gr"))
(defun btg() (fuser "btg" "gr"))
(defun ttg() (fuser "ttg" "gr"))
(defun tig() (fuser "tig" "gr"))
(defun lvi() (setq lvi_c (/ (atof (get_tile "lvi_t")) 10.0)) (princ "\nlvi (cm)= ") (princ lvi_c))
(defun lvs() (setq lvs_c (/ (atof (get_tile "lvs_t")) 10.0)) (princ "\nlvs (cm)= ") (princ lvs_c))
(defun h1() (setq h1_c (/ (atof (get_tile "h1_t")) 10.0)) (princ "\nh1 (cm)= ") (princ h1_c))
(defun hp1() (setq hp1_c (/ (atof (get_tile "hp1_t")) 10.0)) (princ "\nhp1 (cm)= ") (princ hp1_c))
(defun tp1() (setq tp1_c (/ (atof (get_tile "tp1_t")) 10.0)) (princ "\ntp1 (cm)= ") (princ tp1_c))
(defun tiv1() (setq tiv1_c (/ (atof (get_tile "tiv1_t")) 10.0)) (princ "\ntiv1 (cm)= ") (princ tiv1_c))
(defun ttv1() (setq ttv1_c (/ (atof (get_tile "ttv1_t")) 10.0)) (princ "\nttv1 (cm)= ") (princ ttv1_c))
(defun lvc() (setq lvc_c (/ (atof (get_tile "lvc_t")) 10.0)) (princ "\nlvc (cm)= ") (princ lvc_c))
(defun h2() (setq h2_c (/ (atof (get_tile "h2_t")) 10.0)) (princ "\nh2 (cm)= ") (princ h2_c))
(defun hp2() (setq hp2_c (/ (atof (get_tile "hp2_t")) 10.0)) (princ "\nhp2 (cm)= ") (princ hp2_c))
(defun tp2() (setq tp2_c (/ (atof (get_tile "tp2_t")) 10.0)) (princ "\ntp2 (cm)= ") (princ tp2_c))
(defun tiv2() (setq tiv2_c (/ (atof (get_tile "tiv2_t")) 10.0)) (princ "\ntiv2 (cm)= ") (princ tiv2_c))
(defun ttv2() (setq ttv2_c (/ (atof (get_tile "ttv2_t")) 10.0)) (princ "\nttv2 (cm)= ") (princ ttv2_c))
(defun icon_c (img)
(putimage_c "i_t" img 0 0))
(defun putimage_c (image slide dx dy)
(start_image image)
(slide_image dx dy (fix (* (- (dimy_tile image) (* 2 dx)) 1.35)); 1.35
(- (dimy_tile image) (* 2 dy)) slide)
(end_image))
;;;desenare 3d cadru portant
(defun desen_cp (/)
(setvar "osmode" 0)
(setvar "orthomode" 0)
(setvar "BLIPMODE" 0)

```

```

(setvar "CMDECHO" 0)
(setvar "TEXTEVAL" 1)
(setvar "DIMZIN" 1)
(command "LAYER" "n" "stilp_cp,grinda_cp,vuta_cp,placa_pres_cp,scris_cp,cote_cp,hasuri_cp"
  "c" "red" "cote_cp,hasuri_cp"
  "c" "3" "placa_pres_cp,vuta_cp"
  "c" "254" "scris_cp,stilp_cp,grinda_cp" "")
(command "units" "2" "6" "1" "4" "0" "N")
(command "UCSICON" "ON")
(command "UCSICON" "OR")
(command "UCS" "world")
(command "elev" "0" "0")
(command "LAYER" "S" "stilp_cp" "")
(setq o1 (list 0 0))
(xload "ame")
(solw dens 5)
(setq inalt_c (* inalt_c 100.00))
(setq deschid_c (* deschid_c 100.00))
;;;sect transv stilp
(cond
  ((= tip_st_c "USER")
    (setq hi (- hs_c (* tts_c 2.00)))
    (setq dt1 (/ (- bts_c tis_c) 2.00))
    (command "pline"
      (setq p_st_sect (polar o1 (/ pi 2.00) (/ bts_c 2.00)))
      (setq ps (polar p_st_sect (/ pi -2.00) bts_c))
      (setq ps (polar ps 0 tts_c))
      (setq ps (polar ps (/ pi 2.00) dt1))
      (setq ps (polar ps 0 hi))
      (setq ps (polar ps (/ pi -2.00) dt1))
      (setq ps (polar ps 0 tts_c))
      (setq ps (polar ps (/ pi 2.00) bts_c))
      (setq ps (polar ps pi tts_c))
      (setq ps (polar ps (/ pi -2.00) dt1))
      (setq ps (polar ps pi hi))
      (setq ps (polar ps (/ pi 2.00) dt1))
      "c" );
    (setq ps nil))
  ((= (substr tip_st_c 1 3) "IPE")
    (pipe lista_st))
  (and
    (= (substr tip_st_c 1 1) "I")
    (/= (substr tip_st_c 2 1) "P"))
    (profi lista_st)))
(command "change" "l" "" "p" "e" "0" "t" inalt_c "") ;in W
(command "ucs" "x" "90")
(setq o2 (polar o1 (/ pi 2.00) inalt_c)) ;cm
;;o3
(setq alfa_c (/ (* pi alfa_c) 180)) ;;rad
(setq beta_c (+ (/ pi 4.00) (/ alfa_c 2.00))) ;;rad
(setq p1 (polar o1 0 tts_c)) ;cm
(setq p3 (polar o2 (/ pi -2.00) (/ tp1_c (sin beta_c))))
(setq p2 (polar p3 (* beta_c -1.00) (/ tts_c (sin beta_c))))

```

```

(command "ucs" "or" p3)
(command "ucs" "z" (/ (* (- beta_c alfa_c) -180) pi))
(command "ucs" "x" "-90")
(setq es (entlast))
(solcut es "3" '(0 0) '(100 100) '(200 0) '(0 0 -500))
(setq es (ssname (ssget "L") 0))
;;;pina aici stilp
(command "ucs" "w")
(command "ucs" "x" "90")
(setq o2 (polar o1 (/ pi 2.00) inalt_c)) ;cm
(command "ucs" "or" o2)
(command "ucs" "z" (/ (* alfa_c 180) pi))
(command "ucs" "z" "-90")
(command "ucs" "x" "-90")
(command "LAYER" "S" "grinda_cp" "")
(setq hi (- hg_c (* ttg_c 2.00)))
(setq dt1 (/ (- btg_c tig_c) 2.00))
(setq o2 (list 0 0))
(cond
  ((= tip_gr_c "USER")
    (command "pline"
      (setq p_st_sect (polar o2 (/ pi 2.00) (/ btg_c 2.00)))
      (setq ps (polar p_st_sect (/ pi -2.00) btg_c))
      (setq ps (polar ps 0 ttg_c))
      (setq ps (polar ps (/ pi 2.00) dt1))
      (setq ps (polar ps 0 hi))
      (setq ps (polar ps (/ pi -2.00) dt1))
      (setq ps (polar ps 0 ttg_c))
      (setq ps (polar ps (/ pi 2.00) btg_c))
      (setq ps (polar ps pi ttg_c))
      (setq ps (polar ps (/ pi -2.00) dt1))
      (setq ps (polar ps pi hi))
      (setq ps (polar ps (/ pi 2.00) dt1))
      "c" )
    (setq ps nil))
  ((= (substr tip_gr_c 1 3) "IPE")
    (pipe lista_gr))
  ((and
    (= (substr tip_gr_c 1 1) "I")
    (/= (substr tip_gr_c 2 1) "P"))
    (profi lista_gr)))
  (command "change" "l" "" "p" "e" "0" "t"
    (/ deschid_c (* 2.00 (cos alfa_c))) "" ) ;in W
  (command "ucs" "w")
  (command "ucs" "x" "90")
  (setq o2 (polar o1 (/ pi 2.00) inalt_c)) ;cm
  (setq p12 (polar o2 alfa_c (/ tp1_c (sin beta_c))))
  (command "ucs" "or" p12)
  (command "ucs" "z" (/ (* (- beta_c alfa_c) -180) pi))
  (command "ucs" "x" "-90")
  ; (solidify (entlast))
  (setq eg (entlast))
  (solcut eg "3" '(0 0) '(100 100) '(200 0) '(0 0 500))

```

```

;;;pina aici grinda partea inf
(command "ucs" "w")
(command "ucs" "x" "90")
(setq o2 (polar o1 (/ pi 2.00) inalt_c)) ;cm
(setq o3 (polar o2 alfa_c (/ deschid_c (* 2.00 (cos alfa_c))))))
(setq p14 (polar o3 (+ pi alfa_c) (/ tp2_c (cos alfa_c))))
(command "ucs" "or" p14)
(command "ucs" "y" "90")
(setq eg2 (ssname (ssget "L") 0))
(solcut eg2 "3" '(0 0) '(100 100) '(200 0) '(0 0 -500))
(setq eg2 (ssname (ssget "L") 0))
;;;pina aici grinda partea sup
(command "ucs" "w")
(command "ucs" "x" "90")
(setq o2 (polar o1 (/ pi 2.00) inalt_c)) ;cm
(command "LAYER" "S" "placa_pres_cp" "")
(command "pline"
o2
(polar o2 (+ (- pi beta_c) alfa_c) h1_c)
(polar
  (polar o2 (+ (- pi beta_c) alfa_c) h1_c)
  (+ (- pi beta_c) alfa_c (/ pi 2))
  tp1_c)
(polar
  (polar
    (polar o2 (+ (- pi beta_c) alfa_c) h1_c)
    (+ (- pi beta_c) alfa_c (/ pi 2))
    tp1_c) (- alfa_c beta_c) hp1_c)
(polar
  (polar
    (polar
      (polar o2 (+ (- pi beta_c) alfa_c) h1_c)
      (+ (- pi beta_c) alfa_c (/ pi 2))
      tp1_c) (- alfa_c beta_c) hp1_c) (+ (- (/ pi 2) beta_c) alfa_c) tp1_c)
"c" )
(command "change" "l" "" "p" "e" (/ bts_c -2.00) "t" bts_c "") ;in W
(setq epp (entlast))
(solidify epp)
(setq epp (ssname (ssget "L") 0))
(command "mirror" epp "" o2 (polar o2 (- alfa_c beta_c) 300) "n")
(setq epp2 (ssname (ssget "L") 0))
;;;pina aici placa_pres str
(command "ucs" "w")
(command "ucs" "x" "90")
(command "LAYER" "S" "vuta_cp" "")
(setq o2 (polar o1 (/ pi 2.00) inalt_c)) ;cm
(setq p11
(polar
  (polar
    (polar o2 (+ (- pi beta_c) alfa_c) h1_c)
    (+ (- pi beta_c) alfa_c (/ pi 2.00000))
    tp1_c) (- alfa_c beta_c) hp1_c)
(setq p6

```

```

(inters
  p11 (polar p11 (+ (/ pi 2.00) beta_c) 100)
  (polar o1 0 hs_c)
  (polar o2 0 hs_c)
  nil))
(setq p15 (polar p6 (/ pi -2) (- (cadr p6) (- inalt_c lvi_c))))
(setq p152
  (polar
    (polar p15 (/ pi 2) 5.00)
    0 5.00))
(setq p151
  (inters
    p6 p15
    (polar p15 (/ pi -4) ttv1_c)
    (polar p152 (/ pi -4) ttv1_c)
    nil))
(setq h12
  (sqrt
    (- (* (distance p152 p11) (distance p152 p11))
      (* ttv1_c ttv1_c))))
(setq u1 (atan (/ h12 ttv1_c)))
(setq ui (- u1 (angle p152 p11)))
(setq p111 (polar p11 (- pi ui) ttv1_c))
(setq p110
  (inters
    p152 p111
    p6 p11
    nil))
(setq p150
  (inters
    p151 (polar p151 (/ pi 4) 100)
    p11 (polar p11 (angle p110 p152) 100)
    nil))
(command "pline" p11 p110 p152 p150 "c")
(command "change" "l" "" "p" "e" (/ bts_c -2.00) "t" bts_c "") ;in W
(setq etvi (entlast))
(solidify etvi)
(setq etvi (ssname (ssget "L") 0))
;;;pina aici talpa vuta inf str
(command "pline" p15 p152 p150 p151 "c")
(command "change" "l" "" "p" "e" (/ bts_c -2.00) "t" bts_c "") ;in W
(setq ertvi (entlast))
(solidify ertvi)
(setq ertvi (ssname (ssget "L") 0))
;;;pina aici rigidizare talpa vuta inf str
(command "pline" p6 p110 p152 p15 "c")
(command "change" "l" "" "p" "e" (/ tiv1_c -2.00) "t" tiv1_c "") ;in W
(setq evi (entlast))
(solidify evi)
(setq evi (ssname (ssget "L") 0))
;;;pina aici inima vuta inf str
(setq p18 (polar p11 beta_c (* tp1_c 2)))
(setq p19 (polar (polar o2 alfa_c lvs_c) (- alfa_c (/ pi 2)) hg_c))

```

```

(setq p20
(inters
  p18
  (polar p18 (angle p11 p6) 200)
  p19
  (polar p19 (+ pi alfa_c) 200)
  nil ))
(setq p21
(polar
  (polar p19 (+ pi alfa_c) 5.0000)
  (+ pi alfa_c (/ pi 2.0000))
  5.00))
(setq p22
(inters
  p20 p19
  (polar p21 (+ (angle p19 p21) (/ pi 2.00)) ttv1_c)
  (polar p19 (+ (angle p19 p21) (/ pi 2.00)) ttv1_c)
  nil ))
(setq h12
(sqrt
  (- (* (distance p18 p21) (distance p18 p21))
    (* ttv1_c ttv1_c))))
(setq u1 (atan (/ h12 ttv1_c)))
(setq p180 (polar p18 (+ (angle p18 p21) u1) ttv1_c))
(setq p210 (polar p21 (+ (angle p21 p18) u1) ttv1_c))
(setq p23
(inters
  p21 p180
  p18 p20
  nil ))
(setq p24
(inters
  p18 p210
  p22 (polar p22 (angle p19 p21) 100)
  nil ))
(command "pline" p18 p23 p21 p24 "c")
(command "change" "l" "" "p" "e" (/ btg_c -2.00) "t" btg_c "") ;in W
(setq etvs (entlast))
(solidify etvs)
(setq etvs (ssname (ssget "L") 0))
;;;pina aici talpa vuta sup str
(command "pline" p19 p21 p24 p22 "c")
(command "change" "l" "" "p" "e" (/ btg_c -2.00) "t" btg_c "") ;in W
(setq ertvs (entlast))
(solidify ertvs)
(setq ertvs (ssname (ssget "L") 0))
;;;pina aici rigidizare talpa vuta sup str
(command "pline" p19 p21 p23 p20 "c")
(command "change" "l" "" "p" "e" (/ tiv1_c -2.00) "t" tiv1_c "") ;in W
(setq evs (entlast))
(solidify evs)
(setq evs (ssname (ssget "L") 0))
;;;pina aici inima vuta sup str

```

```

(command "LAYER" "S" "placa_pres_cp" "")
(command "pline"
o3
(setq pc (polar o3 (/ pi 2.0000) h2_c))
(setq pc (polar pc pi tp2_c))
(setq p25 (polar pc (/ pi -2.0000) hp2_c))
(setq pc (polar p25 0 tp2_c))
"c" )
(command "change" "l" "" "p" "e" (/ btg_c -2.00) "t" btg_c "") ;in W
(setq eppc (entlast))
(solidify eppc)
(setq eppc (ssname (ssget "L" ) 0))
;;;pina aici placa_pres coama
(command "LAYER" "S" "vuta_cp" "")
(setq p26
(polar
(polar o3 (+ pi alfa_c) lvc_c)
(+ pi alfa_c (/ pi 2)) hg_c))
(setq p27
(polar
(polar p26 alfa_c 5.0000)
(- alfa_c (/ pi 2)) 5.0000))
(setq p28
(inters
p26 (polar p26 alfa_c 100)
(polar p26 (+ (angle p27 p26) (/ pi 2)) ttv2_c)
(polar p27 (+ (angle p27 p26) (/ pi 2)) ttv2_c)
nil ))
(setq p29
(inters
p26 (polar p26 alfa_c 100)
p14 p25
nil ))
(setq h75
(sqrt (- (* (distance p25 p27) (distance p25 p27))
(* ttv2_c ttv2_c))))
(setq u1 (atan (/ h75 ttv2_c)))
(setq p250 (polar p25 (- (angle p25 p27) u1) ttv2_c))
(setq p270 (polar p27 (- (angle p27 p25) u1) ttv2_c))
(setq p30
(inters
p27 p250
p14 p25
nil ))
(setq p31
(inters
p270 p25
p28 (polar p28 (angle p26 p27) 100)
nil ))
(command "pline" p25 p30 p27 p31 "c")
(command "change" "l" "" "p" "e" (/ btg_c -2.00) "t" btg_c "") ;in W
(setq etvc (entlast))
(solidify etvc)

```

```

(setq ertvc (ssname (ssget "L") 0))
;;;pina aici talpa vuta coama
(command "pline" p26 p28 p31 p27 "c")
(command "change" "l" "" "p" "e" (/ btg_c -2.00) "t" btg_c "") ;in W
(setq ertvc (entlast))
(solidify ertvc)
(setq ertvc (ssname (ssget "L") 0))
;;;pina aici rigidizare talpa vuta coama
(command "pline" p29 p30 p27 p26 "c")
(command "change" "l" "" "p" "e" (/ tiv2_c -2.00) "t" tiv2_c "") ;in W
(setq evc (entlast))
(solidify evc)
(setq evc (ssname (ssget "L") 0))
;;;pina aici inima vuta coama
(command "_mirror" evc ertvc ertvc eppc evs ertvs etvs evi ertvi etvi epp epp2 eg2 es "" o3 "@100<-90" "n")
(command "ucs" "w")
(setq nr_cp (fix (+ (/ sum_travee_c travee_c 1))))
(command "array" "all" "" "r" nr_cp "1" (* travee_c 100))
(princ)
)
(defun c:lindab()
(lindab)
(desen_cp)
;;;
)
(defun pipe (lprof / d b t i r punct1)
(setq d (nth 1 lprof))
(setq b (nth 2 lprof))
(setq ti (nth 4 lprof))
(setq t (nth 3 lprof))
(setq r (nth 5 lprof))
(setq punct1 '(0 0 0))
(command "_pline" punct1
  (setq punct1 (polar punct1 pi (/ b 2)))
  (setq punct1 (polar punct1 (/ pi 2) t))
  (setq punct1 (polar punct1 0 (- (/ (- b ti) 2) r)))
  "a" (setq punct1 (polar punct1 (/ pi 4) (* r (sqrt 2))))
  "l" (setq punct1 (polar punct1 (/ pi 2) (- d (* 2 (+ t r)))))
  "a" (setq punct1 (polar punct1 (/ (* pi 3) 4) (* r (sqrt 2))))
  "l" (setq punct1 (polar punct1 pi (- (/ (- b ti) 2) r)))
  (setq punct1 (polar punct1 (/ pi 2) t))
  (setq punct1 (polar punct1 0 b))
  (setq punct1 (polar punct1 (/ pi -2) t))
  (setq punct1 (polar punct1 pi (- (/ (- b ti) 2) r)))
  "a" (setq punct1 (polar punct1 (/ (* pi 3) -4) (* r (sqrt 2))))
  "l" (setq punct1 (polar punct1 (/ pi -2) (- d (* 2 (+ t r)))))
  "a" (setq punct1 (polar punct1 (/ pi -4) (* r (sqrt 2))))
  "l" (setq punct1 (polar punct1 0 (- (/ (- b ti) 2) r)))
  (setq punct1 (polar punct1 (/ pi -2) t))
  "c" )
(command "_rotate" "l" "" "0,0,0" "-90")
(princ))
(defun profi (lprof); / h b d t p rmic rmare punct j x ymic ymare punctg xp1 yp1 lista lprof)

```

```

; (setq lis (assoc (strcat "l" (itoa inalt)) li))
(setq h (nth 1 lprof))
(setq b (nth 2 lprof))
(setq t (nth 3 lprof))
(setq d (nth 4 lprof))
(setq rmic (nth 5 lprof))
(setq rmare (nth 4 lprof))
(setq p 14)
(setq punct '(0 0 0))
(setq j "bc"); (getstring "\nPozitia:")
(cond
  ((= (strcase j) "BL")
    (setq punctg (list (+ (car punct) (* 0.5 b)) (+ (cadr punct) (* 0.5 h)) 0.0)))
  ((= (strcase j) "BC")
    (setq punctg (list (car punct) (+ (cadr punct) (* 0.5 h)) 0.0)))
  ((= (strcase j) "BR")
    (setq punctg (list (- (car punct) (* 0.5 b)) (+ (cadr punct) (* 0.5 h)) 0.0)))
  ((= (strcase j) "ML")
    (setq punctg (list (+ (car punct) (* 0.5 b)) (cadr punct) 0.0)))
  ((= (strcase j) "MC")
    (setq punctg (list (car punct) (cadr punct) 0.0)))
  ((= (strcase j) "MR")
    (setq punctg (list (- (car punct) (* 0.5 b)) (cadr punct) 0.0)))
  ((= (strcase j) "TL")
    (setq punctg (list (+ (car punct) (* 0.5 b)) (- (cadr punct) (* 0.5 h)) 0.0)))
  ((= (strcase j) "TC")
    (setq punctg (list (car punct) (- (cadr punct) (* 0.5 h)) 0.0)))
  ((= (strcase j) "TR")
    (setq punctg (list (- (car punct) (* 0.5 b)) (- (cadr punct) (* 0.5 h)) 0.0)))
  (t (setq punctg (list (car punct) (+ (cadr punct) (* 0.5 h)) 0.0))))
(setq
  ;; colt stinga-jos
  x (* 0.0025 p b)
  ymic (* p rmic (sqrt (/ 1.0 (+ 10000.0 (* p p)))))
  ymare (* p rmare (sqrt (/ 1.0 (+ 10000.0 (* p p)))))
  xp1 (- (car punctg) (* 0.5 b))
  yp1 (- (cadr punctg) (* 0.5 h))
  lista (cons
    (list
      xp1
      yp1
      0.0) ; punct 1
    lista)
  lista (cons
    (list
      xp1
      (+ yp1 t (- x) (* (- rmic) (- (sqrt (+ 1.0 (* 0.01 0.01 p p))) (* 0.01 p)))
      0.0) ; punct 2
    lista)
  lista (cons
    (list
      (+ xp1 rmic (- ymic))
      (+ yp1 t (- x) (* 0.01 p (- rmic ymic)))

```

```

0.0);punct 3
lista)
lista (cons
  (list
    (+ xp1 (* 0.5 (- b d)) (- rmare) ymare)
    (+ yp1 (* 0.0025 p (- b (* 2 d))) t (- (* 0.01 p (- rmare ymare))))
    0.0);punct 4
  lista)
lista (cons
  (list
    (+ xp1 (* 0.5 (- b d)))
    (+ yp1 t (* 0.0025 p (- b (* 2 d))) (* rmare (- (sqrt (+ 1.0 (* 0.01 0.01 p p))) (* 0.01 p))))
    0.0);punct 5
  lista)
;;:colt stinga-sus
x (* 0.0025 p b)
ymic (* p rmic (sqrt (/ 1.0 (+ 10000.0 (* p p))))
ymare (* p rmare (sqrt (/ 1.0 (+ 10000.0 (* p p))))
xp1 (- (car punctg) (* 0.5 b))
yp1 (+ (cadr punctg) (* 0.5 h))
lista (cons
  (list
    (+ xp1 (* 0.5 (- b d)))
    (- yp1 t (* 0.0025 p (- b (* 2 d))) (* rmare (- (sqrt (+ 1.0 (* 0.01 0.01 p p))) (* 0.01 p))))
    0.0);punct 5
  lista)
lista (cons
  (list
    (+ xp1 (* 0.5 (- b d)) (- rmare) ymare)
    (- yp1 (* 0.0025 p (- b (* 2 d))) t (- (* 0.01 p (- rmare ymare))))
    0.0);punct 4
  lista)
lista (cons
  (list
    (+ xp1 rmic (- ymic))
    (- yp1 t (- x) (* 0.01 p (- rmic ymic)))
    0.0);punct 3
  lista)
lista (cons
  (list
    xp1
    (- yp1 t (- x) (* (- rmic) (- (sqrt (+ 1.0 (* 0.01 0.01 p p))) (* 0.01 p))))
    0.0);punct 2
  lista)
lista (cons
  (list
    xp1
    yp1
    0.0);punct 1
  lista)
;;:colt dreapta-sus
x (* 0.0025 p b)
ymic (* p rmic (sqrt (/ 1.0 (+ 10000.0 (* p p))))

```

```

ymare (* p rmare (sqrt (/ 1.0 (+ 10000.0 (* p p))))))
xp1 (+ (car punctg) (* 0.5 b))
yp1 (+ (cadr punctg) (* 0.5 h))
lista (cons
  (list
    xp1
    yp1
    0.0) ;punct 1
  lista)
lista (cons
  (list
    xp1
    (- yp1 t (- x) (* (- rmic) (- (sqrt (+ 1.0 (* 0.01 0.01 p p))) (* 0.01 p))))
    0.0) ;punct 2
  lista)
lista (cons
  (list
    (- xp1 rmic (- ymic))
    (- yp1 t (- x) (* 0.01 p (- rmic ymic)))
    0.0) ;punct 3
  lista)
lista (cons
  (list
    (- xp1 (* 0.5 (- b d)) (- rmare) ymare)
    (- yp1 (* 0.0025 p (- b (* 2 d))) t (- (* 0.01 p (- rmare ymare))))
    0.0) ;punct 4
  lista)
lista (cons
  (list
    (- xp1 (* 0.5 (- b d)))
    (- yp1 t (* 0.0025 p (- b (* 2 d))) (* rmare (- (sqrt (+ 1.0 (* 0.01 0.01 p p))) (* 0.01 p))))
    0.0) ;punct 5
  lista)
;;;colt dreapta-jos
x (* 0.0025 p b)
ymic (* p rmic (sqrt (/ 1.0 (+ 10000.0 (* p p))))))
ymare (* p rmare (sqrt (/ 1.0 (+ 10000.0 (* p p))))))
xp1 (+ (car punctg) (* 0.5 b))
yp1 (- (cadr punctg) (* 0.5 h))
lista (cons
  (list
    (- xp1 (* 0.5 (- b d)))
    (+ yp1 t (* 0.0025 p (- b (* 2 d))) (* rmare (- (sqrt (+ 1.0 (* 0.01 0.01 p p))) (* 0.01 p))))
    0.0) ;punct 5
  lista)
lista (cons
  (list
    (- xp1 (* 0.5 (- b d)) (- rmare) ymare)
    (+ yp1 (* 0.0025 p (- b (* 2 d))) t (- (* 0.01 p (- rmare ymare))))
    0.0) ;punct 4
  lista)
lista (cons
  (list

```

```

(- xp1 rmic (- ymic))
(+ yp1 t (- x) (* 0.01 p (- rmic ymic)))
0.0);punct 3
lista)
lista (cons
(list
xp1
(+ yp1 t (- x) (* (- rmic) (- (sqrt (+ 1.0 (* 0.01 0.01 p p))) (* 0.01 p))))
0.0);punct 2
lista)
lista (cons
(list
xp1
yp1
0.0);punct 1
lista))
(reverse lista)
;(setvar "cmdecho" 0)
(command "_pline"
(nth 0 lista) (nth 1 lista) "a" (nth 2 lista) "l" (nth 3 lista)
"a" (nth 4 lista) "l" (nth 5 lista) "a" (nth 6 lista) "l" (nth 7 lista)
"a" (nth 8 lista) "l" (nth 9 lista) (nth 10 lista) (nth 11 lista)
"a" (nth 12 lista) "l" (nth 13 lista) "a" (nth 14 lista) "l" (nth 15 lista)
"a" (nth 16 lista) "l" (nth 17 lista) "a" (nth 18 lista) "l" (nth 19 lista)
"c" )
(command ". _rotate" "l" "" "0,0,0" "-90")
(princ)
)
; Copyright (C) 1997 by Gobesz F.Zsongor & Teglas Attila (CONSORT Constructii s.r.l., Cluj-Napoca)

```

SISTEMUL AIC INFO

A. Exemple de fișiere sursă

1. Sursa paginii de prezentare, generată în HTML (ilustrat în figura 27, capitolul 4):

```
<html>
<head> <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
      <meta name="Author" content="Zsongor Gobesz">
      <meta name="KeyWords" content="architecture, arhitectura, building, construction, constructii, engineering, genie civil, inginerie,
romania, roumanie, CAD, CAM, CAE, software, free, open, download">
      <meta name="Classification" content="Business, Information provider, Software development, Electronic commerce, Romania, Edu-
cational, Architecture, Engineering, Construction">
      <meta name="Description" content="AIC Info is the first Romanian A/E/C resource, providing professionals and public with specific
information">
      <meta name="GENERATOR" content="Mozilla/4.61 [en] (Win95; I) [Netscape]">
      <title>AIC Info</title>
<style TYPE="text/css">
<!-- hide
strong { color: #ffffff; background: #006699 }
td {font-family: Arial,Helvetica,Sans-serif; }
a { color: #0000cc; }
#a1 { background: #e0e0e0;
color: #0000ff;
text-decoration: none;
}
#a2 { color: #0000ff;
text-decoration: none;
}
// stop -->
</style>
<script LANGUAGE="JavaScript">
<!-- hide
function butoane() {
rb = Math.random();
bt = Math.round(rb*50);
if (bt == 0) bt = 1;
document.write('<center><a href="http://www.aic.info.ro/cgi-bin/redirect.pl?adr='+bt+' target="_top"></a></center>');
}
function butp() {
rbp = Math.random();
btp = Math.round(rbp*5);
if (btp == 0) btp = 1;
document.write('<center><a href="http://www.aic.info.ro/cgi-bin/redirectp.pl?adr='+btp+' target="_top"></a></center>');
}
// stop hiding -->
</script>
</head>
```

```

<body text="#000066" bgcolor="#FFFFFF" link="#0000FF" vlink="#0000FF" alink="#FF0000" background="http://www.aic.info.ro/bg2.gif">
&nbsp;
<br>&nbsp;
<table BORDER=0 CELLSPACING=0 CELLPADDING=2 WIDTH="100%" >
<tr> <td ALIGN=CENTER VALIGN=TOP>
<center><table BORDER=0 CELLSPACING=0 WIDTH="100%" BGCOLOR="#CCCCCC" >
<tr> <td>
<table BORDER=0 CELLSPACING=0 CELLPADDING=2 WIDTH="100%" BGCOLOR="#FFFFFF" >
<tr> <td BGCOLOR="#FFCC33"><b><font size=-1>Romanian:</font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=01000000">Produse, tehnologii</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://aic.info.ro/cgi-bin/aic/user_gen?c=00000">Promo!</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=08000000">Proiectanti, constructori</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=02000000">Proiecte, detalii</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=03000000">Preturi</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/core.pl?f=cdrom.htm">CD Rom</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/core.pl?f=info.htm">Despre AIC Info</a> / <a href="http://www.aic.info.ro/
cgi-bin/aic/core.pl?f=cinfo.htm">Contact info</a></font></b></td>
</tr> </table> </td> </tr> </table> <p>&nbsp;</p>
<td ALIGN=CENTER VALIGN=TOP>
<table BORDER=0 CELLSPACING=0 CELLPADDING=2 WIDTH="100%" >
<tr> <td> <center><a href="http://www.aic.info.ro/"><img SRC="http://www.aic.info.ro/te.gif" VSPACE=2 BORDER=0 height=41 width=165</
a> <br><font size=-1>"The first Romanian A/E/C Internet resource"</font></center>
</td>
<td ALIGN=CENTER VALIGN=TOP><!-- BEGIN CERAMLINKS ADVERTISING CODE --><script LANGUAGE="JavaScript">
<!--
if ((navigator.appName.indexOf ("Netscape") == -1)
|| (navigator.appVersion.indexOf ("2.0") == -1))
{
var clickurl="http://www.ceramlinks.com/cerambanners/engine/members/link120.asp?515P";
var imageurl="http://www.ceramlinks.com/cerambanners/engine/members/get120.asp?515P";
var randomNumber = new String (Math.random());
imageurl=imageurl+randomNumber.substring (2, 6)+"V1M";
clickurl=clickurl+randomNumber.substring (2, 6)+"V1M";
document.write ('<A HREF="'+clickurl+'" TARGET="_top">');
document.write ('<IMG SRC="'+imageurl+'" width=120 height=60 border=0></A>');
}
//--> </script>
<noscript><a href="http://www.ceramlinks.com/cerambanners/engine/members/link120.asp?515P1V1M" TARGET="_top"><img SRC="http:/
/www.ceramlinks.com/cerambanners/engine/members/get120.asp?515P1V1M" height=60 width=120></a></noscript><!-- END
CERAMLINKS ADVERTISING CODE --></td>
</tr>
<tr> <td COLSPAN="2" HEIGHT="1" BGCOLOR="#CCCCCC">
<table BORDER=0 CELLSPACING=0 CELLPADDING=0 >
<tr> <td></td>
</tr> </table> </td> </tr>
<tr> <td><form name="searchform" action="http://www.aic.info.ro/cgi-bin/aic/src.pl" method="get"><input type="text" name="src" size="12"
maxlength=50><input type="submit" value="Search"><b><font size=-1>&nbsp;<a href="http://www.aic.info.ro/cgi-bin/aic/

```

```

core.pl?f=searchhelp.htm">Help</a></font></b></td>
<td align=right><img SRC="http://www.aic.info.ro/disk.gif" height=20 width=16><b><font size=-1>
<a href="http://www.aic.info.ro/cgi-bin/aic/core.pl?f=addto.htm">Add your site</a>&nbsp;<img SRC="http://www.aic.info.ro/disk.gif" height=20
width=16>&nbsp;</font></b></div>
</td> </tr>
<tr <td COLSPAN="2" HEIGHT="1" BGCOLOR="#CCCCCC">
<table BORDER=0 CELLSPACING=0 CELLPADDING=0 >
<tr> <td></td>
</tr> </table> </td> </tr>
<tr <td COLSPAN="2"><script type="text/javascript">
<!-- hide
butoane();
// -->
</script>
</td> </tr> </table> </td>
<td ALIGN=CENTER VALIGN=TOP>
<table BORDER=0 CELLSPACING=0 WIDTH="100%" BGCOLOR="#CCCCCC" >
<tr> <td>
<table BORDER=0 CELLSPACING=0 CELLPADDING=2 WIDTH="100%" BGCOLOR="#FFFFFF" >
<tr> <td BGCOLOR="#FFCC33"><b><font size=-1>General resources:</font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=22000000">Import/Export</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=04000000">Software</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=05000000">Links</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=06000000">Books</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=07000000">E- Journals</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/HyperNews/get/general.html">Forum</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/core.pl?f=indexen.htm">About</a> / <a href="http://www.aic.info.ro/cgi-
bin/aic/core.pl?f=cinfoen.htm">Contact</a></font></b></td>
</tr> </table> </td> </tr> </table> </td> </tr> </table>
<table BORDER=0 CELLSPACING=0 WIDTH="100%" BGCOLOR="#CCCCCC" >
<tr BGCOLOR="#CCCCCC"> <td>
<table BORDER=0 CELLSPACING=0 CELLPADDING=2 WIDTH="100%" BGCOLOR="#FFCC33" >
<tr> <td><font face="Arial,Helvetica"><font size=-1>You are here: <a href="http://www.aic.info.ro/index.html">AIC Info Home</a></font></
font></td>
</tr> </table>
<table BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH="100%" BGCOLOR="#FFFFFF" >
<tr> <td VALIGN=TOP WIDTH="130">
<table BORDER=0 CELLSPACING=0 CELLPADDING=3 WIDTH="100%" HEIGHT="100%" BGCOLOR="#FFFFFF" >
<tr> <td ALIGN=CENTER VALIGN=TOP>
<script type="text/javascript">
<!-- hide
butp();
// -->
</script>
<p><a href="http://home.netscape.com/download" target="_top"><img SRC="http://www.aic.info.ro/now.gif" BORDER=0 height=31 width=89></
a> <p><a href="http://www.microsoft.com/ie/download" target="_top" border="0"><img SRC="http://www.aic.info.ro/ie.gif" BORDER=0

```

```

height=31 width=88></a></td>
</tr> </table>
<br>&nbsp;<p>&nbsp;</p></td>
<td VALIGN=TOP WIDTH="1" HEIGHT="100%" BGCOLOR="#CCCCCC"></td>
<td VALIGN=TOP>
<table BORDER=0 CELLSPACING=0 CELLPADDING=3 WIDTH="100%" HEIGHT="100%" BGCOLOR="#FFFFFF" >
<tr> <td WIDTH="670">
<table CELLPADDING=5 WIDTH="100%" >
<tr> <td VALIGN=TOP>
<center><b><font face="Arial,Helvetica"><font size=+1>The special link of this month ...</font></font></b></center>
<p><a href="http://www.pbs.org/wgbh/nova/bridge/"><img SRC="but/19991101.jpg" HSPACE=20 BORDER=0 align=LEFT></a><a
href="http://www.pbs.org/wgbh/nova/bridge/"><img SRC="but/19991102.gif" HSPACE=20 BORDER=0 align=RIGHT></a> How would
you span a freeway? A canyon? A river? Or an ocean waterway? <a href="http://www.pbs.org/wgbh/nova/bridge/">Learn</a> about the
four major types of bridges and then test your knowledge by matching the right bridge to the right location.
</td> </tr> </table> </td> </tr> </table> </td> </tr> </table> </td>
</tr> </table>
<center> <p><script type="text/javascript">
<!-- hide
butoane();
// -->
</script> </center>
<p><br> <table BORDER=0 CELLSPACING=0 WIDTH="100%" BGCOLOR="#CCCCCC" >
<tr> <td>
<table BORDER=0 CELLSPACING=0 CELLPADDING=3 WIDTH="100%" BGCOLOR="#FFFFFF" >
<tr BGCOLOR="#FFCC33"> <td>
<div align=right><a href="#top" id="a2"><img SRC="http://www.aic.info.ro/th_up.gif" BORDER=0 height=19 width=25></a><b><font
face="Arial,Helvetica"><font size=-1><a href="#top" id="a2">Top</a></font></font></b></div>
</td> </tr>
<tr> <td> <center><b><font face="Arial,Helvetica"><font color="#000066"><font size=-2>Copyright &copy; 1998-1999 EXAROM Consulting
srl Cluj-Napoca - Romania. All rights reserved<a href="http://aic.info.ro/listfbpsc.htm" id="a1">.</a></font></font></font></b>
<br><b><font face="Arial,Helvetica"><font color="#000066"><font size=-2>All the documents contained under aic.info.ro Internet domain are
informative and not contractual.</font></font></font></b></center>
</td> </tr> </table>
</td> </tr> </table>
<center></form></center>
</body>
</html>

```

2. Sursa HTML a paginii principale din BPSC (ilustrată parțial în figura 26, capitolul 4), generată automat în urma unei selecții executate printr-un *gen.pl* (program Perl):

```
<html>
<head>
  <meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
  <meta name="KeyWords" content="architecture, arhitectura, building, construction, constructii, engineering, genie civil, inginerie,
romania, roumanie, CAD, CAM, CAE, software, free, open, download">
  <meta name="Classification" content="Business, Information provider, Software development, Electronic commerce, Romania, Edu-
cational, Architecture, Engineering, Construction">
  <meta name="Description" content="AIC Info is the first Romanian A/E/C resource, providing professionals and public with specific
information">
  <meta name="GENERATOR" content="Mozilla/4.61 [en] (Win95; I) [Netscape]">
  <title>AIC Info</title>
<style TYPE="text/css">
<!-- hide
strong { color: #ffffff; background: #006699 }
td {font-family: Arial,Helvetica,Sans-serif; }
a { color: #0000cc; }
#a1 { background: #e0e0e0;
color: #0000ff;
text-decoration: none;
}
#a2 { color: #0000ff;
text-decoration: none;
}
// stop -->
</style>
<script LANGUAGE="JavaScript">
<!-- hide
function butoane() {
rb = Math.random();
bt = Math.round(rb*50);
if (bt == 0) bt = 1;
document.write('<center><a href="http://www.aic.info.ro/cgi-bin/redirect.pl?adr='+bt+'"" target="_top"></a></center>');
}
function butp() {
rbp = Math.random();
btp = Math.round(rbp*5);
if (btp == 0) btp = 1;
document.write('<center><a href="http://www.aic.info.ro/cgi-bin/redirectp.pl?adr='+btp+'"" target="_top"></a></center>');
}
// stop hiding -->
</script>
</head>
<body text="#000066" bgcolor="#FFFFFF" link="#0000FF" vlink="#0000FF" alink="#FF0000" background="http://www.aic.info.ro/bg2.gif">
<table BORDER=0 CELLSPACING=0 CELLPADDING=2 WIDTH="100%" > <tr> <td ALIGN=Center VALIGN=TOP>
<center><table BORDER=0 CELLSPACING=0 WIDTH="100%" BGCOLOR="#CCCCCC" > <tr> <td>
<table BORDER=0 CELLSPACING=0 CELLPADDING=2 WIDTH="100%" BGCOLOR="#FFFFFF" >
<tr> <td BGCOLOR="#FFCC33"><b><font size=-1>Romanian:</font></b></td>
```

```

</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=01000000">Produse, tehnologii</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://aic.info.ro/cgi-bin/aic/user_gen?c=00000">Promo!</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=08000000">Proiectanti, constructori</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=02000000">Proiecte, detalii</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=03000000">Preturi</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/core.pl?f=cdrom.htm">CD Rom</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/core.pl?f=info.htm">Despre AIC Info</a> / <a href="http://www.aic.info.ro/cgi-bin/aic/core.pl?f=cinfo.htm">Contact info</a></font></b></td>
</tr> </table> </td> </tr>
</table></center>
<p>&nbsp;</td>
<td ALIGN=Center VALIGN=TOP>
<table BORDER=0 CELLSPACING=0 CELLPADDING=2 WIDTH="100%" >
<tr> <td>
<center><a href="http://www.aic.info.ro/"><img SRC="http://www.aic.info.ro/te.gif" VSPACE=2 height=41 width=165 border=0></a>
<br><font size=-1>"The first Romanian A/E/C Internet resource"</font></center>
</td>
<td ALIGN=Center VALIGN=TOP><!-- BEGIN CERAMLINKS ADVERTISING CODE --><script LANGUAGE="JavaScript">
<!--
if ((navigator.appName.indexOf ("Netscape") == -1)
    || (navigator.appVersion.indexOf ("2.0") == -1))
{
    var clickurl="http://www.ceramlinks.com/cerambanners/engine/members/link120.asp?515P";
    var imageurl="http://www.ceramlinks.com/cerambanners/engine/members/get120.asp?515P";
    var randomNumber = new String (Math.random());
    imageurl=imageurl+randomNumber.substring (2, 6)+"V1M";
    clickurl=clickurl+randomNumber.substring (2, 6)+"V1M";
    document.write ('<A HREF="'+clickurl+'" TARGET="_top">');
    document.write ('<IMG SRC="'+imageurl+'" width=120 height=60 border=0></A>');
}
//--> </script>
<noscript><a href="http://www.ceramlinks.com/cerambanners/engine/members/link120.asp?515P1V1M" TARGET="_top"><img SRC="http://www.ceramlinks.com/cerambanners/engine/members/get120.asp?515P1V1M" height=60 width=120></a></noscript><!-- END CERAMLINKS ADVERTISING CODE --></td>
</tr>
<tr> <td COLSPAN="2" HEIGHT="1" BGCOLOR="#CCCCCC">
<table BORDER=0 CELLSPACING=0 CELLPADDING=0 >
<tr> <td></td>
</tr> </table> </td> </tr>
<tr> <td><form name="searchform" action="http://www.aic.info.ro/cgi-bin/aic/src.pl" method="get"><input type="text" name="src" size="12"
maxlength=50> <input type="submit" value="Search"><b> <font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/core.pl?f=searchhelp.htm">Help</a></font></b></td>
<td>
<div align=right><img SRC="http://www.aic.info.ro/disk.gif" height=20 width=16><b><font size=-1>&nbsp;<a href="http://www.aic.info.ro/cgi-bin/aic/core.pl?f=addto.htm">Add your site</a></font></b>&nbsp;<img SRC="http://www.aic.info.ro/disk.gif" height=20 width=16>&nbsp;</div>
</td> </tr>

```

```

<tr> <td COLSPAN="2" HEIGHT="1" BGCOLOR="#CCCCCC">
<table BORDER=0 CELLSPACING=0 CELLPADDING=0 >
<tr> <td></td>
</tr> </table> </td>
</tr>
<tr> <td COLSPAN="2"><script type="text/javascript">
<!-- hide
butoane();
// -->
</script>
</td>
</tr> </table> </td>
<td ALIGN=CENTER VALIGN=TOP>
<table BORDER=0 CELLSPACING=0 WIDTH="100%" BGCOLOR="#CCCCCC" >
<tr> <td>
<table BORDER=0 CELLSPACING=0 CELLPADDING=2 WIDTH="100%" BGCOLOR="#FFFFFF" >
<tr> <td BGCOLOR="#FFCC33"><b><font size=-1>General resources:</font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=22000000">Import/Export</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=04000000">Software</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=05000000">Links</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=06000000">Books</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/gen.pl?c=07000000">E- Journals</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/HyperNews/get/general.html">Forum</a></font></b></td>
</tr>
<tr> <td><b><font size=-1><a href="http://www.aic.info.ro/cgi-bin/aic/core.pl?f=indexen.htm">About</a> / <a href="http://www.aic.info.ro/cgi-
bin/aic/core.pl?f=cinfoen.htm">Contact</a></font></b></td>
</tr> </table> </td> </tr> </table> </td> </tr> </table>
<table BORDER=0 CELLSPACING=0 WIDTH="100%" BGCOLOR="#CCCCCC" >
<tr BGCOLOR="#CCCCCC"> <td>
<table BORDER=0 CELLSPACING=0 CELLPADDING=2 WIDTH="100%" BGCOLOR="#FFCC33" >
<tr> <td>
<div align=right>
<font face="Arial,Helvetica"><font size=-1>
<!-- sfarsit corp initial -->
</center><p><br>You are here: Produse/tehnologii
< <a href=http://www.aic.info.ro/index.html>AIC Info Home</a>&nbsp;<!-- inceput corp mijloc -->
</font></font></div>
</td>
</tr> </table>
<table BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH="100%" BGCOLOR="#FFFFFF" >
<tr> <td VALIGN=TOP width="130">
<table BORDER=0 CELLSPACING=0 CELLPADDING=3 WIDTH="100%" HEIGHT="100%" BGCOLOR="#FFFFFF" >
<tr> <td ALIGN=CENTER VALIGN=TOP>
<script type="text/javascript">
<!-- hide
butp(); // -->
</script>

```

```

<p><a href="http://home.netscape.com/download" target="_top"><img SRC="http://www.aic.info.ro/now.gif" height=31 width=89 border=0></a>
<p><a href="http://www.microsoft.com/ie/download" target="_top" border=0><img SRC="http://www.aic.info.ro/ie.gif" height=31 width=88 border=0></a></td>
</tr> </table>
<br>&nbsp;
<p>&nbsp;</td>
<td VALIGN=TOP WIDTH="1" HEIGHT="100%" BGCOLOR="#CCCCCC"></td>
<td VALIGN=TOP></form>
<table BORDER=0 CELLSPACING=0 CELLPADDING=3 WIDTH="100%" HEIGHT="100%" BGCOLOR="#FFFFFF" >
<tr> <td width=670>
<!-- sfarsit corp mijloc -->
<table width="100%" height=380 cellpadding=5><tr><td valign=top><font face="Arial,Helvetica" size=-1><b><a href="http://aic.info.ro/cgi-bin/aic/gen.pl?c=01010000" id="a1">Servicii </a></b><p><b><a href="http://aic.info.ro/cgi-bin/aic/gen.pl?c=01020000" id="a1">Unelte, echipamente si utilaje </a></b><p><b><a href="http://aic.info.ro/cgi-bin/aic/gen.pl?c=01030000" id="a1">Materii prime si materiale </a></b><p><b><a href="http://aic.info.ro/cgi-bin/aic/gen.pl?c=01040000" id="a1">Elemente, subansamble, stadii fizice </a></b><p><b><a href="http://aic.info.ro/cgi-bin/aic/gen.pl?c=01050000" id="a1">Rețele, echipamente si dotari interioare </a></b><p><b><a href="http://aic.info.ro/cgi-bin/aic/gen.pl?c=01060000" id="a1">Rețele, echipamente si dotari exterioare </a></b><p><b><a href="http://aic.info.ro/cgi-bin/aic/gen.pl?c=01080000" id="a1">Constructii integrale </a></b><p><b><a href="http://aic.info.ro/cgi-bin/aic/gen.pl?c=01090000" id="a1">Intretinere, reparatii </a></b><p><b><a href="http://aic.info.ro/cgi-bin/aic/gen.pl?c=01110000" id="a1">Demolare, recuperare, reciclare </a></b><p><b><a href="http://aic.info.ro/cgi-bin/aic/gen.pl?c=01150000" id="a1">Protectii active si pasive contra incendiilor. Managementul protectiei contra incendiilor.</a></b><p><center><table border=0><tr></table></center></td></tr></table><!-- inceput corp final --></center>
</td> </tr> </table> </td> </tr> </table> </td> </tr> </table>
<font size=-2></font>
<center>
<p><script type="text/javascript">
<!-- hide
butoane();
// -->
</script>
</center>
<br><font size=-2></font>&nbsp;
<table BORDER=0 CELLSPACING=0 WIDTH="100%" BGCOLOR="#CCCCCC" >
<tr> <td>
<table BORDER=0 CELLSPACING=0 CELLPADDING=3 WIDTH="100%" BGCOLOR="#FFFFFF" >
<tr BGCOLOR="#FFCC33"> <td>
<div align=right><a href="#top" id="a2"><img SRC="http://www.aic.info.ro/th_up.gif" BORDER=0 height=19 width=25></a><b><font face="Arial,Helvetica"><font size=-1>
<a href="#top" id="a2">Top</a></font></font></b></div>
</td> </tr>
<tr> <td>
<center><b><font face="Arial,Helvetica"><font color="#000066"><font size=-2>Copyright &copy; 1998-1999 EXAROM Consulting srl Cluj-Napoca - Romania. All rights reserved<a href="http://aic.info.ro/listfbpsc.htm" id="a1">.</a></font></font></font></b>
<br><b><font face="Arial,Helvetica"><font color="#000066"><font size=-2>All the documents contained under aic.info.ro Internet domain are informative and not contractual.</font></font></font></b></center>
</td> </tr> </table> </td> </tr> </table>
</body>
</html>

```

3. Sursa în JavaScript pentru afișarea aleatoare a *banner*-elor publicitare (cu funcția *butoane* apelată din paginile anterioare):

```
var i, k, r, r1, mesf = " ", bt1, bt2, bt3, cc, xs = 0;
adr = new Array();
adr[1]="/bpl.htm\n\n";
adr[2]="/bpsc.htm\n\n";
adr[3]="/act/index.htm\n\n";
adr[4]="/styrofoam/index.htm\n\n";
adr[5]="/klausen/index.htm\n\n";
adr[6]="/macon_deva/index.htm\n\n";
adr[7]="/politub/index.htm\n\n";
adr[8]="/incerc_cj/index.htm\n\n";
adr[9]="/rigips/index.htm\n\n";
adr[10]="/romexim/index.htm\n\n";
adr[11]="/tomsa/index.htm\n\n";
adr[12]="/zero_aluminium/index.htm\n\n";
adr[13]="/schiedel/index.htm\n\n";
adr[14]="/termoplast/index.htm\n\n";
adr[15]="/aco/index.htm\n\n";
adr[16]="/destpa/index.htm\n\n";
adr[17]="/tanoprest/index.htm\n\n";
adr[18]="/deceuninck/index.htm\n\n";
adr[19]="/aquapol/index.htm\n\n";
adr[20]="/vanolux/index.htm\n\n";
adr[21]="/rigips/index.htm\n\n";
adr[22]="/romexim/index.htm\n\n";
adr[23]="/vt/index.htm\n\n";
adr[24]="/amdoro/index.htm\n\n";
adr[25]="/schiedel/index.htm\n\n";
adr[26]="/termoplast/index.htm\n\n";
adr[27]="/soft.htm\n\n";
adr[28]="/vt/index.htm\n\n";
adr[29]="http://www.aic.info.ro/HyperNews/get/general.htm\n\n";
adr[30]="/cautare.htm\n\n";
function butoane() {
  rb = Math.random();
  bt = Math.round(rb*30);
  if (bt==0) bt=1;
  document.write('<center><a href="'+adr[bt]+'"'></a>');
}
```

B. Privire asupra bazei de cunoștințe

1. Pentru a ilustra volumul disponibil de cunoștințe (la ora actuală) prin sistemul AIC Info, în cele ce urmează prezint partea esențială din conținutul bazei de date, prin care se gestionează acesta. Având în vedere structura înregistrărilor și lungimea câmpurilor (chiar dacă am omis descrierile existente ale resurselor, cuvintele cheie și zonele secundare), sunt nevoit să prezint listingul rezultat împărțit pe câte două pagini alăturate:

01	02	02	RO	http://webgmb.gmb.ro/briobit.htm	
01	03	02	RO	http://concrete-int.hypermart.net/index.html	
01	03	03	RO	http://www.romwin.ccir.ro/CONSTRUCTEXPO/atrium/indexr.htm	
01	05	02	RO	http://www.climalux.ro	
01	06	02	RO	http://webgmb.gmb.ro/briobit.htm	
01	15	01	RO	http://webgmb.gmb.ro/briobit.htm	
02	01	00	RO	http://www.netsoft.ro/cjmures/certifi.html	
02	02	00	RO	http://www.aic.info.ro/bds/schiedel/index.html	
02	03	00	RO	http://www.aic.info.ro/bds/cinderella/index.html	
02	04	00	RO	http://www.aic.info.ro/bds/bdc/index.html	
05	01	00	EN	http://www.spec-net.com.au/	
05	03	00	EN	http://www.vtt.fi/cic/links/	
05	05	00	EN	http://www.sweets.com	
05	06	00	EN	http://www.buildingonline.com/	
05	08	00	EN	http://www.build.com.au/	
05	09	00	EN	http://www.pascalarquitectos.com/	
01	06	03	RO	http://www.aic.info.ro/destpa/index.html	
06	05	00	EN	http://catalog.wiley.com/title.cgi?isbn=0471127426	
06	06	00	EN	http://www.wiley.com	
06	07	00	EN	http://www.wkap.nl/book.htm/0-7923-5283-1	
06	08	00	EN	http://www.wiley.com	
06	09	00	EN	http://catalog.wiley.com/title.cgi?isbn=0471129186	
06	10	00	EN	http://catalog.wiley.com/title.cgi?isbn=0471984450	
08	01	00	RO	http://www.building.ro/inter.html	
08	02	00	RO	http://www.romwin.ccir.ro/CONSTRUCTEXPO/atrium/indexr.htm	
08	03	00	RO	http://www.vipnet.ro/camiconi/Despre.htm	
08	04	00	RO	http://www.mic.ro/clients/conimpuls/romania1.html	
08	05	00	RO	http://www.alsicor.ro/index.htm	
08	06	00	RO	http://www.romwin.ccir.ro/CONSTRUCTEXPO/romproiect/indexr.htm	
08	07	00	RO	http://www.eltop.ro/elco/index.html	
08	08	00	RO	http://www.romwin.ccir.ro/CONSTRUCTEXPO/citc/indexr.htm	
08	09	00	RO	http://www.topnet.ro/bittbau/	
08	10	00	RO	http://www.cianet.ro/rvc/	
08	11	00	RO	http://business.fortunecity.com/lerner/704/pagina1.htm	
07	01	00	EN	http://www.aic.info.ro/ej/298SIPS.html	
07	02	00	EN	http://www.aic.info.ro/ej/enrg2.htm	
07	03	00	RO	http://www.aic.info.ro/ej/v6n1_02a.html	
07	04	00	RO	http://www.aic.info.ro/ej/v7n103a.html	
07	05	00	EN	http://www.aic.info.ro/ej/v7n104a.html	
04	04	01	EN	http://www.data-maxx.com/	
04	01	01	EN	http://www.turbocad.com	
04	03	01	EN	http://www.prokon.com	

BRIO BIT SRL Constanta	
Concrete SA Bucuresti	
ATRIUM Design srl	
CLIMALUX Center Bucuresti	
BRIO BIT SRL Constanta	
BRIO BIT SRL Constanta	
Administratie publica - Jud. Mures	
Schiedel sisteme de cosuri de fum srl Brasov	
Birou de proiectare Dico & Tiganas srl Cluj-Napoca	
Birou de proiectare Dico & Tiganas srl Cluj-Napoca	
Spec-Net	
Construction IT at VTT	
Sweet_s	
BuildingOnLine	
Buildata	
Pascal Arquitectos	
D.S. & P. GmbH Germania	
Guide to Stability Design Criteria for Metal Structures, 5th Ed. Edited by: Theodore V. Galambos	
Soil Mechanics in Engineering Practice, 3rd Edition By Karl Terzaghi & al.	
Vrancea Earthquakes: Tectonics, Hazard and Risk Mitigation, edited by F. Wenzel, D. Lungu, O. Novak	
Slope Stability and Stabilization Methods By Lee W. Abramson & al.	
Matrix Structural Analysis, 2nd Ed. by William McGuire & al.	
Airport Builders by Marcus Binney	
INTERCONSTRUCT Romania Constanta	
ATRIUM DESIGN srl	
CAMICONI - S.R.L. Bucuresti	
CONIMPLUS SA Bacau	
C.C. ALSICOR SA Bucuresti	
ROMPROIECT SA Bucuresti	
Electroconstructia SA Tg. Jiu	
C.I.T.C. SA Bucuresti	
Bitt-Bau srl Miercurea Ciuc	
ROMVIOCONS srl CLuj-Napoca	
KIRSCHBAUM srl Sibiu	
SIPs Face the Skeptics, by Polly Sprenger	
Sick Houses: Using Diagnostic Tools To Improve Indoor Air Quality, by Scott Finley	
CLADIREA LUI MICHAEL GRAVES DIN PORTLAND — O analiza de Meredith L. Clausen si Kim Christiansen	
Theory Visualized: Technologically Enhanced Classroom Presentation, by Peter Miniutti and Ruth Klue	
ARCHITECTURE AND TECHN-: The Impossible Project of Tendenza, by Belgin Turan, Middle East Technical	
Data-Maxx	
TurboCAD	
PROKON	

04	02	01	EN	http://www.vistra.de/default-e.htm	
04	05	01	EN	http://www.gnu.org	
04	05	02	EN	http://www.opensource.org/	
04	04	02	RO	http://x-dev.hypermart.net/jos0.htm	
04	04	03	RO	http://www.mmc.ro/devize.htm	
04	04	05	RO	http://www.eunet.ro/SIS/	
04	04	04	RO	http://www.eunet.ro/SIS/	
04	01	03	EN	http://www.softcad.com	
04	05	03	EN	http://www.povray.org	
04	05	04	EN	http://www.linuxberg.com/	
04	05	05	EN	http://www.cdrom.com/pub/infzip/WiZ.html	
01	01	01	RO	http://www.aic.info.ro/incerc_cj/index.html	
01	01	02	RO	http://www.aic.info.ro/politub/index.html	
01	02	01	RO	http://www.aic.info.ro/vanolux/index.html	
01	02	03	RO	http://www.aic.info.ro/amdoro/index.html	
01	03	04	RO	http://www.aic.info.ro/saniterm/index.html	
01	03	05	RO	http://www.aic.info.ro/aquapol/index.html	
01	03	06	RO	http://www.aic.info.ro/styrofoam/index.html	
01	03	08	RO	http://www.aic.info.ro/romexim/index.html	
01	03	09	EN	http://www.deceuninck.com	
01	03	10	RO	http://www.aic.info.ro/tanoprest/index.html	
01	03	01	RO	http://www.aic.info.ro/vanolux/index.html	
01	03	12	RO	http://www.aic.info.ro/macon_deva/index.html	
01	03	11	RO	http://www.aic.info.ro/tomsa/index.html	
01	03	13	RO	http://www.aic.info.ro/aco/index.html	
01	04	01	RO	http://www.aic.info.ro/rigips/index.html	
01	04	02	EN	http://www.aic.info.ro/act/index.html	
01	04	04	EN	http://www.deceuninck.com	
01	04	05	RO	http://www.aic.info.ro/spacedecks/index.html	
01	04	06	RO	http://www.aic.info.ro/zero_aluminium/index.html	
01	09	01	RO	http://www.aic.info.ro/aquapol/index.html	
01	04	07	RO	http://www.aic.info.ro/tomsa/index.html	
01	09	02	RO	http://www.aic.info.ro/tomsa/index.html	
01	05	01	RO	http://www.aic.info.ro/macon_deva/index.html	
01	09	03	RO	http://www.aic.info.ro/romexim/index.html	
01	05	03	RO	http://www.aic.info.ro/schiedel/index.html	
01	06	01	RO	http://www.aic.info.ro/aquapol/index.html	
01	11	01	RO	http://www.deceuninck.com	
01	15	02	RO	http://www.unify.ro	
05	11	00	EN	http://www.aecmedia.net	
05	07	00	EN	http://www.greatbuildings.com/	
04	01	02	EN	http://www.softcad.com	
50	02	00	RO	http://www.softplan.com	
04	01	04	EN	http://www.softplan.com	
50	01	00	RO	http://www.softcad.com	
50	03	00	RO	http://www.softcad.com	
04	01	07	EN	http://www.povray.org	
50	04	00	RO	http://www.povray.org	
04	01	06	EN	http://www.comp.lancs.ac.uk/computing/users/andy/ac3d.html	
04	01	05	EN	http://www.artifice.com	
01	04	09	RO	http://www.condem.inet.ro	
08	12	00	RO	http://www.alca.ro	
08	13	00	RO	http://www.verena.net/axis/	

VISTRA COMPUTER SYSTEMS	
GNU Project	
opensource.org	
Spectrum Proiect Iasi	
M&M Computers Ploiesti	
Societatea de Inginerie Sisteme Bucuresti	
Societatea de Inginerie Sisteme Bucuresti	
ArchiTECH.PC	
Persistence of Vision Raytracer	
Linuxberg	
WiZ	
INCERC - Filiala Cluj	
POLITUB srl Bistrita	
VANOLUX Impex srl Satu-Mare	
Departament Constructii AMDORO srl Ulmeni	
SANITERM srl Oradea	
AQUAPOL Romania SA Oradea	
DOW Chemicals Romania Bucuresti	
ROMEXIM P&M Bucuresti	
DECEUNINCK Romania srl Bucuresti	
TANOPREST srl Oradea	
VANOLUX Impex srl Satu-Mare	
MACON SA Deva	
TOMSA Import Export srl Cluj-Napoca	
ACO Impex srl Cluj-Napoca	
RIGIPS Romania srl Bucuresti	
AMERICAN CONSTRUCTION TECHNOLOGIES srl Bucuresti	
DECEUNINCK Romania srl Bucuresti	
EXAROM Consulting srl Cluj-Napoca	
ZERO Aluminium srl Bucuresti	
AQUAPOL Romania SA Oradea	
TOMSA Import Export srl Cluj-Napoca	
TOMSA Import Export srl Cluj-Napoca	
MACON SA Deva	
ROMEXIM P&M srl Bucuresti	
Schiedel sisteme de cosuri de fum srl Brasov	
AQUAPOL Romania SA Oradea	
DECEUNINCK Romania srl Bucuresti	
Unify srl Cernavoda	
AEC Media	
The Great Buildings Collection	
SoftCAD International	
SoftPlan	
SoftPlan	
SoftCAD 3D	
ArchiTECH PC	
Persistence of Vision Raytracer	
Persistence of Vision	
AC3D — by Andy Colbourne	
DesignWorkshop	
CONDEM SA Bucuresti	
ALCA SA Bucuresti	
AXIS Grup SA Sibiu	

04	01	08	RO	http://www.maptrade.org/	
08	15	00	RO	http://www.ricons.ro	
01	03	14	RO	http://topnet.topnet.ro/hamerock/	
08	16	00	RO	http://www.sintec.ro/rustic/	
01	02	04	RO	http://topnet.topnet.ro/hamerock/	
04	05	06	EN	http://www.photomodeler.com/Lite/	
05	02	00	RO	http://www.building.ro	
07	06	00	EN	http://www.eeri.org/EQ_Basics/LIQ/LIQUEFAC.html	
07	07	00	EN	http://onlineethics.org/cases/iino.html	
01	01	03	RO	http://www.logicnet.ro	
04	01	09	EN	http://www.totalsoft.ro	
04	04	06	EN	http://www.totalsoft.ro	
01	03	15	RO	http://www.aic.info.ro/baseimpex/index.html	
03	01	00	RO	http://www.aic.info.ro/cgi-saniterm/main.pl	
01	05	04	RO	http://Ctr.alexim.ro/	
08	17	00	RO	http://www.deuroconsult.ro/burcea/	
01	02	05	RO	http://www.rolink.ro/euro94/	
01	02	06	RO	http://www.midalgroup.ro/	
08	18	00	RO	http://www.pb.ro	
01	05	05	RO	http://www.recep.ro/Clienti/Simalexim/Home.html	
01	05	06	EN	http://www.recep.ro/Clienti/emmerik/index.htm	
08	19	00	RO	http://www.recep.ro/Clienti/proinstal/prog302.htm	
01	02	07	00	http://www.axagrup.dynip.com/	
01	02	08	RO	http://www.sinta.ro/RMD/fr1075.htm	
08	20	00	RO	http://home.dntis.ro/~avatti/index.html	
08	21	00	RO	http://home.dntis.ro/~servcon/index.html	
01	04	11	RO	http://www.decebal.ro/frisomat/	
01	03	16	RO	http://www.decebal.ro/rolac/	
01	06	04	RO	http://www.netsoft.ro/purator/	
01	03	17	RO	http://www.netsoft.ro/radson/	
01	02	09	RO	http://www.sieta.iiruc.ro/	
01	03	18	EN	http://www.tepro.ro	
01	03	19	RO	http://www.intfor.ro	
01	04	12	RO	http://www.building.ro/gg/WELCOME.htm	
08	14	00	RO	http://www.netsoft.ro/izotim/index.html	
01	06	05	RO	http://www.aic.info.ro/maccaferri	
01	04	13	RO	http://www.canad.ro/proconsul/	
01	01	04	RO	http://www.sherut.israel.net/f_profrom.htm	
01	04	14	RO	http://contral.quattro.ro/	
01	04	15	RO	http://www.romwin.ccir.ro/CONSTRUCTEXPO/magic/indexr.htm	
01	04	16	RO	http://www.tehnoem.ro/index.htm	
01	05	07	RO	http://www.infocities.ro/Expo/NUMEEXPO/Prezentari/clima.htm	
04	01	10	EN	http://domino2.kappa.ro/interactiv/RCAD.nsf	
04	03	02	EN	http://www.engr.usask.ca/%7Emacphed/finite/fe_resources/fe_resources.html	
07	08	00	EN	http://www.wkap.nl/sampletoc.htm?0921-030X+17+1+1998	
03	02	00	RO	http://www.aic.info.ro/termoplast/prel.htm	
03	03	00	RO	http://www.aic.info.ro/amdoro/wienese/index.html	
08	22	00	RO	http://members.wbs.net/homepages/6/1/s/61s.html	
04	02	02	EN	http://domino2.kappa.ro/interactiv/RCAD.nsf/All/Home	
05	04	00	EN	http://www.aecinfo.com	
04	03	03	EN	http://www.grsoft-ltd.com	
01	03	20	RO	http://www.reokorr.hu/rom/beton.htm	
01	03	21	RO	http://www.ronline.ro/Mikim/Pag1.htm	

International Map Trade Association	
RICON_S	
HAMEROCK SA Miercurea Ciuc	
Rustic SRL Baia-Mare	
HAMEROCK SA Miercurea Ciuc	
Photomodeler Lite	
Building Romania	
Liquefaction and what can be done about it	
TWO CASES OF ENGINEERING ERRORS THAT LED TO ACCIDENTS IN JAPAN	
LOGIC Telecom SA	
TOTALSOFT Bucuresti	
TOTALSOFT Bucuresti	
BASE Impex Cluj-Napoca	
Saniterm srl Oradea	
CT Romania Services Bucuresti	
Burse Constructii srl Bucuresti	
Euro'94 comimpex srl	
MIDAL Group	
Proiect Bucuresti	
SIMAL EXIM Deva	
van Emmerik	
PROINSTAL G 302 Bistrita	
Axa Group Braila	
COMELF SA Bistrita	
AVATTI Iasi	
SERV CON ROTPUNKT srl Iasi	
FRISOMAT SA Piatra Neamt	
KOBBER-ROLAC Turturesti Neamt	
PURATOR ECOTEHNIC SRL Targu Mures	
G&M International Instalatii SRL Targu Mures	
Sieta SA Cluj-Napoca	
TEPRO SA Iasi	
INTFOR SA Galati	
G&G Romania	
IZOTIM SA Targu Mures	
TUBOTRADE International Bucuresti	
PRO-CONSUL Bucuresti	
S. M. SHAPIRO ENGINEERS SRL Bucuresti	
CONTRAL srl Focsani	
MAGIC SIGN srl Bucuresti	
TehnoEM srl Bucuresti	
SIC-CLIMAVENT srl Arad	
RCAD SOFTWARE Cluj-Napoca	
INTERNET FINITE ELEMENT RESOURCES	
Prevention and Mitigation of Natural Hazards	
TERMOPLAST Cluj-Napoca	
DEP. CONSTRUCTII AMDORO Ulmeni	
61S Bucuresti	
RCAD SOFTWARE Cluj-Napoca (Romanian)	
AEC Info	
GrSoft AEC Smart Software Solutions	
REOKORR pardoseli colorate de protectie contra prafului	
MIKIM srl Brasov (CASA 2000)	

01	08	01	RO	http://www.aic.info.ro/leda	
05	10	00	EN	http://www.dbm.com.au/Cinfo/Default.HTM	
08	23	00	RO	http://www.kappa.ro/clients/baduc/index.html	
01	06	06	RO	http://www.energo.ro	
01	08	02	RO	http://www.building.ro/ATAGroup/index.html	
08	24	00	RO	http://www.romwin.ccir.ro/CONSTRUCTEXPO/consimex/indexr.htm	
08	25	00	RO	http://www.romwin.ccir.ro/CONSTRUCTEXPO/citc/indexr.htm	
07	09	00	EN	http://www.archrecord.com/Navigation/Frameset.asp?Loadpage=green_architect	
07	10	00	EN	http://www.archrecord.com/Navigation/Frameset.asp?Loadpage=digital_architect	
05	12	00	EN	http://www.publicworks.com/	
01	03	23	RO	http://vest.digiro.net/produse/index.htm	
01	02	11	RO	http://www.umat.ro/	
01	02	12	RO	http://www.intermas.netcompsj.ro/	
01	05	08	RO	http://www.eurosic.ro/	
01	03	24	RO	http://www.topedge.ro/erpia/	
01	02	13	RO	http://www.topedge.ro/erpia/	
04	05	07	EN	http://www.cadalog.com/	
05	13	00	EN	http://www.atlantis-pro.com/	
05	14	00	FR	http://www.virtualhome.be/	
04	02	03	EN	http://www2.publicworks.com/content/downloads/Downloads.asp	
01	03	25	RO	http://marflex.sintec.ro/clienti/marcarex/despre_marcarex.html	
01	03	26	RO	http://www.nextra.ro/trazmar/	
01	05	09	RO	http://www.aic.info.ro/absrl/	
05	15	00	EN	http://www.transdanube.com	
04	01	11	RO	http://www.nemetschek.ro	
01	02	14	RO	http://www.electrimtech.com/romanian/index.html	
01	02	15	RO	http://www.hcstrade.ro/	
04	01	12	EN	http://ftp.arl.mil/brlcad/	
04	01	13	EN	http://www.tm.wb.utwente.nl/~roddeman/tn_release/tnhome.html	
04	01	14	EN	http://www.gv.kotnet.org/~kdf/3dom/	
05	16	00	EN	http://www.omegaconcept.com/	
01	02	16	RO	http://www.tehnofrig.ro/	
08	26	00	RO	http://www.parcons.ro/	
01	02	17	00	http://www.dante.ro/	
01	02	18	RO	http://www.swep.ro/	
01	04	17	RO	http://www.aic.info.ro/absrl/	
01	02	19	RO	http://www.aromet.buzaunet.ro/	
22	01	00	EN	http://www.aic.info.ro/absrl/	
22	02	00	EN	http://www.sinta.ro/RMD/fe1004.htm	
22	04	00	EN	http://topnet.topnet.ro/hamerock/	
22	05	00	EN	http://www.tehnofrig.ro/	
01	06	07	RO	http://www.asio.cz/romania/index.htm	
22	06	00	EN	http://www.hitrom.necomm.ro/	
01	04	03	RO	http://www.aic.info.ro/termoplast/index.html	
01	04	10	RO	http://www.arexim.ro/pvc/	
05	17	00	EN	http://cca.qc.ca/installations/	
07	11	00	EN	http://virtualia.ong.ro/	
07	12	00	EN	http://www.uiah.fi/projects/metodi/111.htm	
01	02	10	RO	http://www.aic.info.ro/epinvest/index.html	
01	04	08	RO	http://www.meganet.ro/dagweb/default.htm	
04	01	15	RO	http://www.bentley.ro	
04	01	16	EN	http://www.graphisoft.com/	
06	01	00	EN	http://www.efnspon.com/	

|LEDA S.A. Satu-Mare
 |Construction InfoNet
 |BADUC SA Bucuresti
 |ENERGOPETROL SA Campina
 |ATA Group Brasov
 |CONSIMEX Trading - Bucuresti
 |C.I.T.C. SA - Bucuresti
 |The Green Architect
 |Digital Architect
 |Public Works Online
 |Prefabricate Vest SA Bucuresti
 |Uzinele Mecanice Timisoara S.A.
 |Intermasters IMPEX srl Simleul Silvaniei
 |EUROSIC S.A.
 |E.R.P.I.A. - S.A. Craiova
 |E.R.P.I.A. - S.A. Craiova
 |CADalog
 |Atlantis Multimedia
 |Virtual Home
 |Public Works — downloads page
 |MARCAREX Viseul de Sus - Maramures
 |TRAZMAR SA Miercurea-Ciuc
 |A & B Satulung (Maramures)
 |TRANSDANUBE
 |NEMETSCHEK
 |ELECTRIM TECH INC.
 |HCS Trade Bucuresti
 |BRL-CAD«
 |TOCHNOG
 |3dom
 |Omega Concept
 |TEHNOFRIG SA Cluj-Napoca
 |PARCONS Timisoara
 |DANTE Systems Bucuresti
 |PRATCO srl Bucuresti
 |A & B Satulung (Maramures)
 |AROMET SA Buzau
 |A&B Ltd. Satulung Romania
 |ARMATURA SA Cluj-Napoca / Romania
 |HAMEROCK SA Miercurea Ciuc / Romania
 |TEHNOFRIG SA Cluj-Napoca / Romania
 |ASIO ROMANIA srl Cluj-Napoca
 |HITROM SA Vaslui / Romania
 |TERMOPLAST srl Cluj-Napoca
 |AREXIM SA Bucuresti
 |CCA 5 Architects
 |VIRTUALIA
 |ARTEOLOGY
 |EPINVEST Cluj-Napoca
 |Dag Tech S.R.L. Bucuresti
 |Bentley
 |ArchCAD
 |E & FN SPON

06	02	00	EN	http://www.wiley.com/	
06	13	00	EN	http://www.wkap.nl/book.htm/0-7923-8491-1	
06	03	00	EN	http://www.wkap.nl/	
06	11	00	EN	http://www.routledge.com/cgi-bin/plweb/search.cgi?dbname+routledge&title+Routledge	
05	18	00	EN	http://www.pbs.org/wgbh/nova/bridge/	
01	03	27	EN	http://www.ductil.ro/	
01	03	28	RO	http://www.elpreco.ro/	
01	02	20	RO	http://www.eufex.ro/	
01	05	10	RO	http://jadex.icnet.ro/	
22	09	00	EN	http://www.luxten.ro/	
01	05	11	RO	http://www.sauter.ro/	
01	05	12	RO	http://www.prescom.ro/	
22	10	00	EN	http://www.feromet.ro/cupreng.htm	
01	03	29	RO	http://www.sintezacolor.ro/	
01	05	13	RO	http://www.swissglass.ro/	
01	01	05	EN	http://www.seiscom.ro/	
01	03	30	RO	http://www.policolor.ro/	
22	11	00	EN	http://www.griro.ro/	
01	02	21	RO	http://www.yorkromania.ro	
22	12	00	EN	http://www.sirma.ro/	
08	27	00	EN	http://www.prominvest.ro/	
04	01	17	RO	http://members.xoom.com/infocad	
01	01	06	RO	http://romanianindustry.com/	
22	13	00	EN	http://www.mic.ro/clients/robind/	
01	05	14	EN	http://www.bravo.ro	
01	03	31	RO	http://www.aic.info.ro/terion/	
01	04	18	RO	http://www.aic.info.ro/terion/	
01	08	03	RO	http://www.aic.info.ro/saniterm/	
01	05	15	RO	http://www.aic.info.ro/saniterm/	
01	05	06	RO	http://www.abindustries.ro/	
22	14	00	EN	http://www.abc2000.ro/	
01	03	32	RO	http://www.aic.info.ro/conex7/	
01	04	19	RO	http://www.aic.info.ro/conex7/	
01	04	20	RO	http://www.novoferm.ro/	
05	19	00	EN	http://www.digdez.com/wood/	
04	05	10	EN	http://download.elsaco.com	
05	20	00	EN	http://www.faqs.org/	
01	03	32	RO	http://www.unimetall.ro/	
22	15	00	EN	http://www.elba.ro/	
01	05	17	RO	http://www.elba.ro/	
01	06	08	RO	http://www.elba.ro/	
22	16	00	EN	http://www.redriverlumber.com/	
22	03	00	EN	http://www.sinta.ro/RMD/fe1008.htm	
06	04	00	EN	http://www.blackwell.co.uk/	
22	07	00	EN	http://www.building.com/tehis/members/construction/concrete/secohouse/	
22	08	00	EN	http://www.ductil.ro/	
01	05	16	RO	http://www.wirsbo.ro/	

John Wiley & Sons	
Intelligent Building Systems, by Albert Ting-pat So and Wai Lok Chan	
KLUWER	
Architecture and Revolution by Neil Leach	
Build a Bridge	
DUCTIL SA Buzau	
ELPRECO Craiova	
EUFEX Targoviste	
JADEX Trading Craiova	
LUXTEN SA Bucharest / Romania	
SAUTER Bucuresti	
PRES COM Brasov	
FEROMET M srl Piatra Neamt / Romania	
SINTEZA COALOR Oradea	
SWISS GLASS Bucuresti	
SEIS COM Bucuresti	
POLICOLOR SA Bucuresti	
GRIRO Bucharest / Romania	
YORK Romania Bucuresti	
Industria Sarmei Campia Turzii / Romania	
PROMINVEST Bucuresti	
Proiecte Software "Online"	
Romanian Industry	
ROBINETE INDUSTRIALE SA / Bacau / Romania	
BRAVO Grup Bucuresti	
TERION srl Brasov	
TERION srl Brasov	
SANITERM srl Oradea	
SANITERM srl Oradea	
AB Industries Bucuresti	
ABC Construct 2000 / Bucharest / Romania	
CONEX7 srl Cluj-Napoca	
CONEX7 srl Cluj-Napoca	
NOVOFERM Romania srl Satu-Mare	
The Woodworkers WebRing	
DOWNLOAD ELSACO	
Internet FAQs Archive	
UNIMETALL srl Targu Mures	
ELBA SA Timisoara / Romania	
ELBA SA Timisoara	
ELBA SA Timisoara	
Red River Lumber Company / U.S.A.	
CEMACON SA Zalau / Romania	
Blackwell Science	
SECO HOUSE / Romania	
DUCTIL SA Buzau / Romania	
Agis International - Bucuresti	

C. Completare la aplicațiile decizionale implementate prin AIC Info

1. Sursa HTML a paginii principale de prezentare din subsistemul decizional realizat pentru Rigips România (ilustrată în figura 32 din capitolul 4):

```
<HTML><HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
<TITLE>Rigips: estimarea costurilor</TITLE>
</HEAD>
<FRAMESET ROWS="17%,*" name="main">
<FRAME NAME="top" src="/cs/topcalc.htm" NORESIZE FRAMEBORDER="NO" SCROLLING="NO" marginheight="3" marginwidth="3"></
FRAME> <FRAMESET cols="70%,*" name="main1">
<FRAME NAME="bottom" src="/cs/calchelp1.htm" FRAMEBORDER="NO" SCROLLING="AUTO"></FRAME><FRAMESET rows="40%,*"
name="main2">
<FRAME NAME="rightup" src="/cs/rightup.htm" FRAMEBORDER="NO" SCROLLING="AUTO"></FRAME>
<FRAME NAME="rightdown" src="/cs/calchelp.htm" FRAMEBORDER="NO" SCROLLING="AUTO"></FRAME>
<NOFRAMES></NOFRAMES>
</FRAMESET></FRAMESET></FRAMESET>
<body onUnload=alert("no!")>
</body></HTML>
```

2. Sursa HTML prin care se generează dinamic partea superioară a paginii curente:

```
<HTML><HEAD>
<TITLE></TITLE>
<style> td { font-family: arial, helvetica, sans-serif; font-size: smaller}
strong { background: #99ffff}
</style></head>
<body text="#000066"><font face="Arial,Helvetica,Sans-serif" size=-2><b>
<script src="/fc.js">
</script>
<script> var win0 = parent.frames[0];
var doc0 = win0.document; var win1 = parent.frames[1]; var doc1 = win1.document; var win3 = parent.frames[3]; var doc3 = win1.document;
var nrfisa = 0;
document.write("<form><table><tr><td>"); for ( i = 1; i <= nrfise; i++) {
    document.write("</td></tr><tr><td><input type='radio' name='r' onblur='reset()' onclick='javascript: nrfisa=" + i + ","; value=" + i + "'></
td><td><font face='Arial,Helvetica' size=-2><b>" + numefisa[i]);
} document.write("</td></tr></table></form>"); document.close()
</script>
</body></html>
```

3. Sursa HTML a cadrului din mijloc, prin care se afișează tabelul cu "rețeta" aleasă și calculația aferentă:

```
<HTML><HEAD>
<META HTTP-EQUIV="Content-Type" CONTENT="text/html; charset=iso-8859-1">
<META NAME="GENERATOR" CONTENT="Mozilla/4.05 [en] (Win95; I) [Netscape]">
<TITLE>Topcalc</TITLE>
<STYLE> td { font-family: arial, helvetica, sans-serif; }
strong { background: yellow}
```

```

</STYLE>
<SCRIPT LANGUAGE="JavaScript">
<!-- hide
    function s2() {
window.open(b,'Info','toolbar=no,location=no,directories=no,status=yes,scrollbars=yes,resizable=yes,width=300,height=300')
    }
// stop hiding -->
</SCRIPT></HEAD>
<BODY TEXT="#000066" BGCOLOR="#EEEEEE">
<SCRIPT src="rigd.js">
</SCRIPT>
<SCRIPT> function fy() {
if (parent.frames[2].nrfisa == 0) parent.frames[3].location = "../plafoane/p21-1.jpg"
}
function fg() {
parent.frames[3].location = "../p"+parent.frames[2].nrfisa+".htm";
}
</SCRIPT>
<CENTER><FORM name="fop"></CENTER>
<CENTER><TABLE BORDER=0 CELLSPACING=0 CELLPADDING=0 WIDTH="100%" >
<TR><TD VALIGN=TOP align="center"></TD>
<TD VALIGN=TOP align="center"><A HREF="javascript: b='calchelp.htm'; s2()"><IMG SRC="l8.jpg" ALT="Instrucțiuni de folosire"
BORDER=0 HEIGHT=15 WIDTH=50></A></TD>
<TD VALIGN=TOP ROWSPAN="2" align="center"><IMG SRC="Bpblogo.jpg" HEIGHT=85 WIDTH=133></TD>
<TD VALIGN=TOP><B><FONT SIZE=-1>Cantitate:</FONT></B></TD>
<TD VALIGN=TOP><B><FONT SIZE=-1>Rata de schimb de referinta:</FONT></B></TD>
<TD VALIGN=TOP ROWSPAN="2" BGCOLOR="#FFFFFF" align="center">
<CENTER><INPUT type="button" value="Go! " onClick="fg()"></CENTER></TD></TR>
<TR><TD COLSPAN="2"><B><FONT SIZE=+2>Calculator de costuri&nbsp;</FONT></B></TD>
<TD VALIGN=TOP><INPUT name="cant" size=3 value=0><FONT SIZE=-1>mp</FONT></TD>
<TD VALIGN=TOP><FONT SIZE=-1>1 ATS =&nbsp;<INPUT name="rsr" size=5 value=0>lei&nbsp;</FONT></TD>
</TR></TABLE></CENTER>
<CENTER></FORM><SCRIPT> this.document.close()</SCRIPT></CENTER>
</BODY></HTML>

```

4. Definirea "rețetelor" posibile (în JavaScript):

```

var numefisa = new Array();
var codfisa = new Array();
var detaliu = new Array();
var nr_lin = new Array();
var nrfise = 16;
function fisa(nrf, nfl, cf, det, nrlin) {
    numefisa[nrf] = nfl;
    codfisa[nrf] = cf;
    detaliu[nrf] = det;
    nr_lin[nrf] = nrlin;
}
fisa( 1,"Tencuiala fara structura","3.10.01","",4);
fisa( 2,"Tencuiala cu structura - prindere pe rama","3.20.04","",8);
fisa( 3,"Tencuiala cu structura - prindere cu bride","3.20.05a","",10);
fisa( 4,"Pereti despartitori CW 50/75 - simpli, cu structura metalica","",9);

```

```
fisa( 5,"Pereti despartitori CW 75/100 - simpli, cu structura metalica",,,,,9);
fisa( 6,"Pereti despartitori pentru bai CW 75/100 - dubli, cu structura metalica",,,,,10);
fisa( 7,"Pereti despartitori CW 100/175 - tripli, cu structura metalica cu rezistenta la foc 180 min.",,,,,11);
fisa( 8,"Placare pod — rezistenta la foc 30 min.",,,,,5);
fisa( 9,"Tencuiala fara structura cu placi termoizolante PLN 10+30",,,,,4);
fisa( 10,"Tencuiala fara structura cu placi termoizolante PMS 10+40",,,,,4);
fisa( 11,"Tencuiala fara structura cu placi termoizolante PLN1 10+60",,,,,4);
fisa( 12,"Plafon fix cu structura metalica",,,,,11);
fisa( 13,"Plafon suspendat cu structura metalica",,,,,12);
fisa( 14,"Plafon curb cu structura metalica",,,,,11);
fisa( 15,"Plafon demontabil tip DECOGIPS - SPANIA cu structura semivizibila",,,,,11);
fisa( 16,"Plafon demontabil tip DECOGIPS - SPANIA cu structura ascunsa",,,,,11);
```

5. Aplicația propriu-zisă (scrisă în JavaScript și prezentată cu comentarii) pentru calcularea valorilor necesare alcătuirii tabelului din cadrul principal, în funcție de componentele incluse în mulțimea de selecție rezultată din filtrarea cu ajutorul rețetelor:

```
// ————— Generator/calculator tabele
function fy() {
  if (parent.frames[2].nrfisa != 0) {
    var total_ftva = 0;
    var win1 = parent.frames[1];
    var doc1 = win1.document;
    doc1.open()
    var win2 = parent.frames[2];
    var doc2 = win2.document;
    var nrf = win2.nrfisa;
    var nlin = win2.nr_lin[nrf];
    var win0 = parent.frames[0];
    var doc0 = win0.document;
    cant = doc0.forms[0].elements[1].value;
    rsr = doc0.forms[0].elements[2].value;

    // ————— cap tabel
    doc1.write("<html><head></head><body bgcolor='#ffffee' text='#000066'
onunload='parent.frames[1].document.location.href=calchelp1.htm'><FONT FACE='Arial,Helvetica,Sans-serif' size=-1><B>RIGIPS
ROMANIA S.R.L.</B><BR>Tel: +40 1 3305614, +40 1 33057-23(24), Fax: +40 1 33056-08(11)<BR>Banca Societe Generale cont
30293<CENTER><B>");
    doc1.write("<TABLE BORDER CELLSAPACING=0 CELLPADDING=3 width='100%'><TR><TD COLSPAN=2><FONT
FACE='Arial,Helvetica,Sans-serif' size=-1><B>FISA DE CALCUL</B><BR>(consumuri normate)</TD><TD COLSPAN=4><FONT
FACE='Arial,Helvetica,Sans-serif' size=-1><b>");
    doc1.write(win2.numefisa[nrf] + "<br>Cod:" + win2.codfisa[nrf] + "<br>Cantitate:" + cant + " mp;<br>Rata de schimb: " + rsr + " lei/ATS</
TD><TD COLSPAN=5><FONT FACE='Arial,Helvetica,Sans-serif' size=-1><form>Cost total fara TVA:<BR>&nbsp;<input name ='vallei'
size=5 value=0.00&nbsp;<lei/mp<BR>&nbsp;<input name ='valats' size=5 value=0.00&nbsp;<ATS/mp</form></TD></tr>");
    doc1.write("<TR><TD><FONT FACE='Arial,Helvetica,Sans-serif' size=-1>Elementul de montaj</TD><TD><FONT
FACE='Arial,Helvetica,Sans-serif' size=-1>Unitate de masura / Unitate de livrare</TD><TD><FONT FACE='Arial,Helvetica,Sans-serif'
size=-1>Cantitate / Unitate de livrare</TD><TD><FONT FACE='Arial,Helvetica,Sans-serif' size=-1>Consum specific<BR>[ UM / mp]</
TD><TD><FONT FACE='Arial,Helvetica,Sans-serif' size=-1>Consum normal</TD><TD><FONT FACE='Arial,Helvetica,Sans-serif'
size=-1>Cantitate necesara [unitati de livrare]</TD><TD><FONT FACE='Arial,Helvetica,Sans-serif' size=-1>Cantitate livrabila [unitati de
livrare]</TD><TD><FONT FACE='Arial,Helvetica,Sans-serif' size=-1>Pret unitar (ATS)</TD><TD><FONT FACE='Arial,Helvetica,Sans-
serif' size=-1>Valoare fara TVA (lei)</TD></tr>");
    // ————— corp tabel
    var i=1,j,f1,f2,f3,f4,f5;
```

```

while ( nlin > 0 ) {
    j = idx[j];
    f1 = csp[j]*cant;
    f2 = csp[j]*cant/win0.car[j];
    f3 = Math.floor(f2) + Math.floor(100*(f2-Math.floor(f2))/100);
    f4 = Math.floor(f3);
    if ( (f4 - f3) != 0 ) f4++;
    f5 = Math.round(win0.put[j]*f4*rsr);
    total_ftva = total_ftva + f5;

    doc1.write('<tr><td><FONT FACE="Arial,Helvetica,Sans-serif" size=-1><a href=http://localhost/exarom/rigips/
'+win0.lnk[j]+'>'+win0.den[j]+'</a></td><td><FONT FACE="Arial,Helvetica,Sans-serif" size=-1>'+win0.ums[j]+'</td><td><FONT
FACE="Arial,Helvetica,Sans-serif" size=-1>'+win0.car[j]);
    doc1.write('</td><td><FONT FACE="Arial,Helvetica,Sans-serif" size=-1>'+csp[j]+'</td><td><FONT FACE="Arial,Helvetica,Sans-serif"
size=-1>'+f1+'</td><td><FONT FACE="Arial,Helvetica,Sans-serif" size=-1>'+f3);
    doc1.write('</td><td><FONT FACE="Arial,Helvetica,Sans-serif" size=-1>'+f4+'</td><td><FONT FACE="Arial,Helvetica,Sans-serif" size=-
1>'+win0.put[j]+'</td><td><FONT FACE="Arial,Helvetica,Sans-serif" size=-1>'+f5+'</td></tr>');
    nlin--;
    i++;
}
doc1.write("<TR><TD COLSPAN=8><FONT FACE='Arial,Helvetica,Sans-serif' size=-1><DIV ALIGN=right><B>TOTAL FARA TVA:</
B></DIV></TD>");
doc1.write("<TD><FONT FACE='Arial,Helvetica,Sans-serif' size=-1><b>");
doc1.write(total_ftva + "</b></TD></TR>");
doc1.write("<TR><TD COLSPAN=8><DIV ALIGN=right><FONT FACE='Arial,Helvetica,Sans-serif' size=-1><B>TOTAL CU TVA
(22%):</B></DIV></TD><TD><FONT FACE='Arial,Helvetica,Sans-serif' size=-1><b>" + Math.round(total_ftva*1.22) + "</b></TD></
TR></TABLE>");
doc1.forms[0].elements[0].value = Math.round(total_ftva / cant);
var f6 = total_ftva / cant / rsr;
doc1.forms[0].elements[1].value = Math.floor(f6) + Math.floor(100*(f6-Math.floor(f6))/100);
// ----- terminare generare tabel
// doc0.close();
doc1.close();
//doc2.close();
}
}

```

6. Setul de cunoștințe folosit de subsistem, modelat prin JavaScript:

```

den[1] = "Placa RB 12.5 mm"; ums[1] = "m"; car[1] = 2.75; put[1] = 26.84;
den[2] = "Profil UW 30"; ums[2] = "m"; car[2] = 3; put[2] = 5.179;
den[3] = "Placa RIFLEX 6 mm"; ums[3] = "m"; car[3] = 2.4; put[3] = 61.565;
den[4] = "Surub 212/3.5-25 mm"; ums[4] = "buc/cutie"; car[4] = 1000; put[4] = 67.888;
den[5] = "Surub 421/3.5-9.5 mm"; ums[5] = "buc/cutie"; car[5] = 1000; put[5] = 198.554;
den[6] = "Banda garnitura b=30 mm"; ums[6] = "m/rola"; car[6] = 30; put[6] = 36.532;
den[7] = "Surub cu diblu 6 x 42 mm"; ums[7] = "buc/cutie"; car[7] = 100; put[7] = 57.828;
den[8] = "Pasta imbinare SUPER"; ums[8] = "kg/sac"; car[8] = 5; put[8] = 23.011;
den[9] = "Banda imbinare (din fibre de sticla)"; ums[9] = "m/rola"; car[9] = 25; put[9] = 7.399;
den[10] = "Bride 3*6"; ums[10] = "buc/cutie"; car[10] = 100; put[10] = 1.721;
den[11] = "Ghips adeziv"; ums[11] = "kg/sac"; car[11] = 40; put[11] = 116.367;
den[12] = "Profil UW 50"; ums[12] = "m"; car[12] = 4; put[12] = 6.844;
den[13] = "Profil CW 50"; ums[13] = "m"; car[13] = 2.6; put[13] = 8.342;
den[14] = "Banda garnitura b=50 mm"; ums[14] = "m/rola"; car[14] = 30; put[14] = 63.543;

```

den[15] = "Placa PLN 10+30 mm" ums[15] = "m" car[15] = 2.6 put[15] = 106.340
 den[16] = "Ghips adeziv MAP" ums[16] = "kg/sac" car[16] = 25 put[16] = 114.071
 den[17] = "Placa PLN 10+60 mm" ums[17] = "m" car[17] = 2.6 put[17] = 192.444
 den[18] = "Placa PLN1 10+30 mm" ums[18] = "m" car[18] = 2.6 put[18] = 136.747
 den[19] = "Placa PLN1 10+60 mm" ums[19] = "m" car[19] = 2.6 put[19] = 209.795
 den[20] = "Placa RF 15 mm" ums[20] = "m" car[20] = 2.75 put[20] = 36.474
 den[21] = "Placa PMS 10+40 mm" ums[21] = "m" car[21] = 2.6 put[21] = 76.768
 den[22] = "Placa PMS 10+60 mm" ums[22] = "m" car[22] = 2.6 put[22] = 96.729
 den[23] = "Profil UW 75" ums[23] = "m" car[23] = 4 put[23] = 8.224
 den[24] = "Profil CW 75" ums[24] = "m" car[24] = 2.75 put[24] = 9.660
 den[25] = "Banda garnitura b=70 mm" ums[25] = "m/rola" car[25] = 30 put[25] = 86.854
 den[26] = "Vata minerala" ums[26] = "mp/sul" car[26] = 15.6 put[26] = 23.014
 den[27] = "Placa RBI 12.5 mm" ums[27] = "m" car[27] = 2.75 put[27] = 35.522
 den[28] = "Surub 212/3.5-35 mm" ums[28] = "buc/cutie" car[28] = 26 put[28] = 81.347
 den[29] = "Placa RF 12.5 mm" ums[29] = "m" car[29] = 3 put[29] = 30.453
 den[30] = "Profil UW 100" ums[30] = "m" car[30] = 4 put[30] = 9.152
 den[31] = "Profil CW 100" ums[31] = "m" car[31] = 3.0 put[31] = 10.952
 den[32] = "Banda garnitura b=95 mm" ums[32] = "m/rola" car[32] = 30 put[32] = 126.868
 den[33] = "Surub 212/3.5-55 mm" ums[33] = "buc/cutie" car[33] = 1000 put[33] = 71.835
 den[34] = "Profil plafon CD 27*60*27" ums[34] = "m" car[34] = 4 put[34] = 6.573
 den[35] = "Imbinare liniara profile" ums[35] = "buc/cutie" car[35] = 100 put[35] = 1.453
 den[36] = "Bride 3*6" ums[36] = "buc/cutie" car[36] = 100 put[36] = 1.721
 den[37] = "Profil UW 30" ums[37] = "m" car[37] = 3 put[37] = 5.179
 den[38] = "Tija de suspensie l=500 mm" ums[38] = "buc/legat." car[38] = 100 put[38] = 1.114
 den[39] = "Piesa de suspendare" ums[39] = "buc/cutie" car[39] = 100 put[39] = 2.214
 den[40] = "Piesa de ancorare" ums[40] = "buc/cutie" car[40] = 250 put[40] = 0.705
 den[41] = "Diblu DN 6" ums[41] = "buc/cutie" car[41] = 100 put[41] = 1.497
 den[42] = "Profil pentru tavan curb" ums[42] = "m" car[42] = 4 put[42] = 54.460
 den[43] = "Tija de suspensie NONIUS L = 240 mm" ums[43] = "buc/legat" car[43] = 100 put[43] = 3.907
 den[44] = "Piesa de suspendare NONIUS L = 190 mm" ums[44] = "buc/cutie" car[44] = 100 put[44] = 3.117
 den[45] = "Siguranta NONIUS" ums[45] = "buc/cutie" car[45] = 250 put[45] = 0.719
 den[46] = "Pasta de imbinare VARIO" ums[46] = "kg/sac" car[46] = 5 put[46] = 49.219
 den[47] = "Piesa de incrucisare" ums[47] = "buc/cutie" car[47] = 100 put[47] = 2.031
 den[48] = "Placa 0.60 x 0.60" ums[48] = "mp/cutie" car[48] = 2.16 put[48] = 60.809
 den[49] = "Profil principal 24 mm" ums[49] = "m" car[49] = 3.6 put[49] = 11.973
 den[50] = "Profil secundar" ums[50] = "m" car[50] = 1.2 put[50] = 11.973
 den[51] = "Profil de compartimentare" ums[51] = "m" car[51] = 0.6 put[51] = 11.973
 den[52] = "Profil perimetral" ums[52] = "m" car[52] = 3.0 put[52] = 9.578
 den[53] = "Diblu filetat" ums[53] = "buc/cutie" car[53] = 100 put[53] = 1.916
 den[54] = "Tija filetata" ums[54] = "buc/cutie" car[54] = 100 put[54] = 5.986
 den[55] = "Piesa de suspendare" ums[55] = "buc" car[55] = 1 put[55] = 3.592
 den[56] = "Saiba" ums[56] = "buc" car[56] = 1 put[56] = 0.479
 den[57] = "Piulita" ums[57] = "buc" car[57] = 1 put[57] = 0.239
 den[58] = "Placa 0.60 x 0.60" ums[58] = "mp/cutie" car[58] = 1.44 put[58] = 97.356
 den[59] = "Profil principal OCULTO" ums[59] = "m" car[59] = 4.0 put[59] = 13.884
 den[60] = "Profil de compartimentare OCULTO" ums[60] = "m" car[60] = 0.6 put[60] = 4.292
 den[61] = "Profil perimetral" ums[61] = "m" car[61] = 3.0 put[61] = 10.098
 den[62] = "Diblu filetat" ums[62] = "buc/cutie" car[62] = 100 put[62] = 2.020
 den[63] = "Tija filetata" ums[63] = "buc/cutie" car[63] = 100 put[63] = 6.311
 den[64] = "Piesa de suspendare OCULTO" ums[64] = "buc" car[64] = 1 put[64] = 4.039
 den[65] = "Piesa de imbinare principala OCULTO" ums[65] = "buc/cutie" car[65] = 100 put[65] = 2.020
 den[66] = "Cuie beton" ums[66] = "buc" car[66] = 1 put[66] = 0.252

REFERINȚE SUPLIMENTARE

A. RESURSE ACTUALE, ACCESIBILE PRIN INTERNET

❶ Inteligență artificială:

CLIPS (software, documentatii etc)	http://www.ghg.net/clips/CLIPS.html
CMU Artificial Intelligence Repository	http://www.cs.cmu.edu/Groups/AI/html/repository.html
Free On-Line Dictionary Of Computing (FOLDOC)	http://wombat.doc.ic.ac.uk/foldoc/index.html
Java Expert System Shell (Jess)	http://herzberg.ca.sandia.gov/jess/
Knowledge Based Engineering (si G2)	http://www.kbe.co.za/
Laboratorul sistemelor bazate pe cunostinte	http://www-ksl.stanford.edu/
Prime Time Freeware for AI (si PC AI)	http://www.primenet.com/pcai/index1.html
Sisteme expert si sisteme bazate pe cunostinte	http://ai.iit.nrc.ca/subjects/Expert.html

❷ Resurse generale dedicate construcțiilor:

AIC Info (sistemul autohton prezentat in teza)	http://www.aic.info.ro
Biblioteket Bygg (Civil Engineering Library)	http://www1.ldc.lu.se/lthvbibl/civ-www.htm
Bricsnet (fostul AEC INFO din Canada)	http://www2.bricsnet.com/
Civil Engineering Virtual Library	http://www.ce.gatech.edu/WWW-CE/
Construction IT Resources: Home Page	http://itc.fgg.uni-lj.si/ICARIS/cr/
Cyburbia - Architecture Images and Exhibits	http://www.arch.buffalo.edu/cgi-bin/pairc/arch_img
Great Buildings Online (arhitectura si modelare)	http://www.greatbuildings.com/
GuideMe.com Civil Engineering Resources on the Internet	http://www.GuideMe.com/CivilEngineering.htm
Meridian Marketing Group	http://www.meridian-marketing.com
Publicitate dedicata constructiilor (AEC Media)	http://www.aecmedia.com/
Publicitate dedicata constructiilor (Ceramlinks)	http://www.ceramlinks.com/cerambanners/us/default.asp
World of Concrete	http://www.worldofconcrete.com/HomePage/WOCHomePage/
WWW Virtual Library: Architectural Engineering	http://energy.arce.ukans.edu/wwwvl/wwwwarce.htm

B. Exemple de sisteme expert comerciale din domeniul construcțiilor

1. CÂTEVA SISTEME EXPERT DIN DOMENIUL CONSTRUCȚIILOR

Denumire:	Destinație:	Software:
SACON	pregătirea datelor pentru MEF	EMYCIN
CDA	inginerie structurală	PROLOG
WISER	proiectarea clădirilor înalte	LISP
BDES	ingineria vântului	PASCAL
DAPS	evaluarea degradării	EXSYS
TIME	evaluare costuri și a durate	PROLOG2
CONE	inginerie geotehnică	LISP, OPS5
KNOW-HOW	evaluarea și gestionarea riscului	---
MASON	evaluarea duratei de zidire	OPS5
PLASHTRAN	asistă utilizatorii MSC/NASTRAN	INTERLISP

2. CÂTEVA SISTEME EXPERT DESTINATE ARHITECTURII

Denumire:	Destinație:	Software:
ALL-RISE	Proiectarea clădirilor	SRL
BERT	Țeserea zidărilor	AUTOLISP
CONSTR.SCHED. ANALYSIS	Planificarea lucrărilor	DBASE III
CPO-ES	Organizare și planificare	DECIDING- FACTOR
DESTINY	Proiectare ambientală	SRL
HI-COST	Estimări de costuri	KADBASE INGRES
HI-RISE	Proiectarea preliminară a clădirilor înalte	LISP
LOW-RISE	Proiectarea clădirilor joase industriale	C, LISP
SOJA	Planificare pe durate scurte	LISP

3. CÂTEVA SISTEME EXPERT PENTRU DRUMURI, CĂI FERATE ȘI PODURI

Denumire:	Destinație:	Software:
HERCULES	Generarea planurilor pentru controlul traficului	CLISP
FOREST ROAD DESIGN	Proiectarea drumurilor forestiere	LISP INSIGHT2+
CHINA	Proiectarea barierelor de sunet	FORTTRAN LISP
EXPERT-UFOS	Proiectarea rețelelor de transport	M1
EXSRM	Anticiparea revărsării râurilor	FORTTRAN LISP
PETRUS	Planificarea traficului subteran	---
PLATFORM	Planificarea proiectelor și a lucrărilor	LISP
PRESERVER	Întreținerea drumurilor	OPS5
RETWAIL	Asistă proiectarea sprijinirilor	PROLOG, C
RODEOS	Proiectarea curbelor	BASIC
ROSE	Întreținere preventivă a drumurilor	FORTTRAN EXSYS
TRALZ	Proiectarea semnalizatoarelor	OPS5
WELDING ADVISOR	Selectarea metodei de sudare	LOTUS 3
WELD DEFECT ADVISOR	Detectarea defectelor de sudură	EXSYS
SPEX	Proiectarea componentelor structurale	FORTTRAN
SCEPTRE	Analiza, proiectarea și reabilitarea pavajelor	EXSYS

Sursele programelor de calculator incluse în anexele din acest document constituie proprietatea intelectuală a autorilor. Reproducerea integrală sau parțială a conținutului acestui document este interzisă fără acordul scris al autorului.

