

ESYS - SISTEM INTERACTIV DE ACHIZIȚIE A CUNOȘTINȚELOR DE LA MAI MULȚI EXPERTI, BAZAT PE TEORIA CONSTRUCȚIILOR PERSONALE ȘI TEHNICI DE ÎNVĂȚARE INDUCTIVĂ

ec. Doina Țilivea

Institutul de Cercetări în Informatică

Rezumat: Lucrarea prezintă sistemul grafic de achiziție a cunoștințelor ESYS, bazat în tehnicile de interviu a experților pe teoria construcțiilor personale cu posibilitatea combinării cunoștințelor de la mai mulți experți despre același subiect. De asemenea, sistemul oferă o interfață pentru extragerea selectivă a unor informații aflate în baze de date clasice care sunt prelucrate (ca și datele provenite direct de la experți) în vederea extragerii de noi cunoștințe sub formă de reguli, arbori de decizii și frame-uri prin folosirea unor tehnici de învățare inductivă.

Cuvinte cheie: achiziția cunoștințelor, "elicitarea" cunoștințelor, teoria construcțiilor personale, grilă repertoar, construcție, atribut, analiza grupurilor de date, reguli de producție, concepte, arbori de decizie.

1. Introducere

Cunoștințele dintr-un domeniu trebuie achiziționate și reprezentate într-un format util, testate, combinate și adaptate pentru a se constitui într-o bază de cunoștințe, asociată unui sistem expert dintr-un domeniu. Se va conveni asupra termenului de "elicitare" ca fiind *modalitatea de a extrage sau de a obține o cunoștință dintr-un domeniu, indiferent de metoda*.

În cercetarea și practica achiziției de cunoștințe s-au separat două direcții majore [9]:

1. tehnici de transfer a cunoștințelor din surse existente cum ar fi experții umani, cărți, documentație etc ("expertise transfer");

2. derivarea de noi cunoștințe prin inducție empirică din exemple sau învățare automată ("machine learning").

Majoritatea sistemelor bazate pe cunoștințe practice, achiziționează cunoștințele prin transfer, și nu prin învățarea din experiență, și au fost dezvoltate o serie de instrumente care asigură un astfel de transfer.

Există o relație fundamentală între cele două paradigme prin faptul că, experiența existentă la un expert uman a fost cândva obținută prin inducție empirică. De asemenea, există o continuitate între cele două paradigme prin aceea că, experiența existentă poate fi parțială, eronată și poate avea diverse forme, deci ea nu poate înlocui complet inducția empirică, dar poate servi la direcționarea și la accelerarea procesului de

achiziție. În primul caz, experții umani în probleme de decizie trebuie să cunoască un set complet, minimal de reguli de decizie, deci un set de cazuri critice descrise în termenii unui set minimal de atribute relevante pentru deciziile corecte; în celălalt caz, expertul trebuie să cunoască o sursă de cazuri descrise în termenii unor atribute relevante pentru decizie și cu o probabilitate destul de mare pentru a obține o decizie corectă.

În concluzie, achiziția de cunoștințe se desfășoară astfel:

- experții dintr-un domeniu sunt intervieuați utilizând instrumente de transfer a cunoștințelor care să nu presupună o formalizare într-un anumit limbaj și care să-i ajute să-și structureze cunoștințele din domeniu;
- acolo unde există baze de date clasice, care conțin deja descrierile de cazuri care prezintă interes pentru rezolvarea problemei, aceste cazuri ar trebui extrase selectiv din aceste baze de date, deoarece există garanția ca aceste surse de informații să fi fost validate prin utilizare de către experții din domeniu, care au construit baza;
- cunoștințele rezultate prin abstractizarea informațiilor rezultate din interviu sau acces la o bază de date sunt prezentate expertului pentru a fi validate, cu posibilitatea corectării sau a respingerii acelor care sunt inconsistente;
- utilizarea cunoștințelor în sisteme de prelucrare a cunoștințelor poate aduce noi revizuri în baza de cunoștințe.

Scopul acestei lucrări a fost realizarea unui sistem de achiziție a cunoștințelor de la mai mulți experți din același domeniu, bazat pe teoria construcțiilor personale [14]. Sistemul asigură achiziția și prelucrarea cunoștințelor în scopul abstractizării lor și al obținerii de noi forme de reprezentare a cunoștințelor cum sunt regulile, arbori de decizie și ierarhiile de concepte.

Baza de cunoștințe astfel rezultată poate fi folosită în două moduri:

- fie intern, fiind posibilă generarea unui sistem expert pentru probleme de clasificare/diagnosticare în domenii în care soluțiile sunt destul de puține pentru a putea fi comod enumerate;
- fie încărcată în contextul unui sistem de prelucrare de cunoștințe mai general, care asigură rafinarea acestor cunoștințe și rezolvarea problemei, atunci când problema depășește

posibilitățile ESYS și necesită și alt fel de cunoștințe.

ESYS poate rezolva complet numai probleme de tip **analiză**, alte probleme de tip sinteză cum ar fi configurarea, planificarea, proiectarea, pentru care soluțiile sunt construite din părți componente, sunt dificil de descris integral folosind tehnica grilelor. Pare dificil de utilizat această tehnică pentru a elicită cunoștințe cauzale, procedurale sau strategice.

Modalitatea de elicitare este grafică și condusă de utilizator, dând posibilitatea descrierii de cunoștințe destul de diverse într-un format accesibil și asistat în permanență de sistem. Scopul major al acestui instrument de achiziție a cunoștințelor a fost asigurarea accesului direct al expertului din domeniu la specificarea cunoștințelor din domeniu, cu intervenția minimală a "inginerului de cunoștințe".

2. Prezentarea generală a sistemului de elicitare ESYS

Achiziția grafică de cunoștințe creează posibilitatea ca experții din domeniu să fie capabili să-și structureze și ei înșiși să-și formalizeze cunoștințele, cu o asistență cât mai limitată a inginerului de cunoștințe. Practica, deși redusă, a elaborării de sisteme expert a arătat că nu este suficient să se furnizeze instrumente puternice de rezolvare a problemelor și că este la fel de important să se pună la punct instrumentele de achiziție a cunoștințelor. Unul din motivele importante pentru care au fost dezvoltate puține sisteme expert a fost acela că expertul din domeniu trebuie să apeleze la inginerul de cunoștințe pentru a-i structura și formaliza experiența și trebuie să facă aceasta ori de câte ori sistemul expert se dezvoltă.

Sistemul ESYS poate funcționa în două ipostaze depinzând de sursa informațiilor de descriere a unui subiect dintr-un domeniu:

1. informațiile sunt obținute **direct de la expert** prin tehnici de interviu speciale, care produc așa-zisele "construcții personale", în această ipostază fiind posibilă mixarea cunoștințelor de la mai mulți experți;

2. informațiile sunt extrase dintr-o bază de date prin intermediul unei componente care necesită prezența expertului pentru rafinarea lor în scopul reprezentării adecvate în baza de cunoștințe, generată de sistem.

Sistemul ESYS are patru componente majore:

1. **interfața cu expertul** pentru elicitarea cunoștințelor care se poate utiliza în una din ipostazele descrise mai sus;

2. **analiza informațiilor** introduse și vizualizarea sugestivă a rezultatelor acestei analize, cu posibilitatea revenirii asupra unor informații introduse în prealabil;

3. **derivarea de noi cunoștințe** din analiza informațiilor introduse: reguli, arbori de decizie, descrieri de concepte sub formă de frame-uri, ierarhii de clasificare într-o bază de cunoștințe;

utilizarea bazei de cunoștințe, fie direct sub ESYS sub forma unor sisteme expert de diagnosticare/clasificare, fie prin intermediul unor sisteme mai generale de prelucrare a cunoștințelor.

2.1 Interfața cu expertul

Sistemul ESYS permite descrierea **multi-expert**, pe baza construcțiilor personale a unor exemple din domeniu sau extragerea lor dintr-o bază de date.

Vom folosi, atunci când vorbim de inducția de reguli pentru exemple, denumirea de soluții deoarece de fapt expertului i se cere să descrie soluții din domeniul respectiv, care, în mod mai general, pot fi denumite și exemple de rezolvare ale problemei. Vom păstra și denumirea de exemple deoarece ea este adecvată atunci când descriem chiar exemple de obiecte dintr-un domeniu pe care vrem să le clasificăm. În prezentare se vor folosi următorii termeni:

- subiectul și scopul elicitării;
- exemple ale subiectului;
- atribute ale acestor exemple exprimate inițial sub forma unor construcții bipolare în care apar două caracteristici opuse ale atributului. Într-o astfel de construcție, polul pozitiv se va asocia pe scala de notare cu nota maximă, iar polul negativ nota minim;
- relațiile dintre exemple și atribute exprimate cu ajutorul unor note pe o scală de notare aleasă de expert și memorate într-o tabelă sau grilă (grid);
- relațiile dintre exemple și atribute exprimate cu ajutorul valorilor acestor atribute într-o grilă eterogenă.

Metodele pentru a obține aceste informații furnizate de sistemul ESYS sunt diferite după caz:

- expertul are clar delimitate exemplele și atributele lor și le introduce pe rând fără să se gândească la legătura dintre ele, urmând ca apoi să noteze doar relațiile dintre ele (prin enumerare);

- expertul are clar delimitate exemplele din domeniu la care se va referi, dar atributele nu îi sunt clare; va folosi, în acest caz, elicitarea triadică;
- expertul nu are clar delimitate, nici exemplele, nici atributele și va folosi, în acest caz, elicitarea diadică, iar de la un anumit nivel elicitarea triadică.

Modalitățile prezentate mai sus pot fi combinate după necesități.

Pentru a descrie relațiile dintre exemple și atribute, expertul are posibilitatea să-și dimensioneze scala de notare și să noteze pentru fiecare construcție asociată atributului exemplele din domeniu (figura 6). Expertul poate reveni asupra informațiilor introduse, având toate posibilitățile de modificare, ștergere, adăugare, atât pentru exemple și atribute, cât și pentru notele pe care le-a introdus.

Sarcina primului expert care este interviuat în legătură cu un anumit domeniu, este cea mai dificilă, deoarece el porneste de la 0, în timp ce ceilalți experți au posibilitatea să lucreze pe exemplele și atributele introduse de experții care au elicitat anterior, cu posibilitatea, evident, de a adăuga, modifica, șterge exemple și/sau construcții.

În final, expertul este invitat să încerce să rafineze construcțiile introduse (figura 11). Astfel, dacă luăm de exemplu construcția "preț scăzut - preț ridicat" referitoare la subiectul "mașină", expertul poate preciza un nume de atribut asociat construcției, în cazul de față "preț". De asemenea, dacă scala de notare are mai mult de două gradații și expertul a indicat la notare note diferite de polii construcției, expertul are posibilitatea să asocieze o valoare cazului luat în considerare. De exemplu, dacă scala asociată "prețului" are trei gradații, nota 2 poate avea semnificația de "preț mediu". Această completare a descrierii unei construcții este o modalitate de a obține dintr-o construcție un atribut care are atașate toate valorile posibile pe care le poate lua atributul. De fapt, când spun toate valorile posibile, mă refer la descrierea tuturor cazurilor posibile în care se poate regăsi valoarea unui atribut. De exemplu, revenind la exemplul cu prețul un preț mic ar putea însemna un preț < 10.000, un preț mediu ar putea însemna un preț cuprins în (10.000,40.000], iar un preț mare ar însemna un preț > 40.000.

Această rafinare a construcțiilor este esențială pentru a obține reguli semnificative, lizibile; altfel, regulile vor conține referințe la construcții și la notele acordate fiecărei construcții ceea ce nu este prea sugestiv.

Procesul de rafinare este folosit și în procesul de extragere a exemplurilor dintr-o bază de date, unde se cere utilizatorului să descrie domeniul de definiție al fiecărui atribut și să împartă acest

domeniu în subdomenii corespunzătoare cazurilor care vor fi luate în considerare la analiza grilei și la generarea regulilor de asociere și regulilor de clasificare.

Astfel, în problemele de diagnostic medical, multe din descrierile de simptome sunt cuantificabile, exprimabile într-o primă descriere cantitativ (astfel de atribute ar putea fi temperatura, tensiunea arterială, rezultatele diverselor analize).

Expertul nu este constrâns în definirea completă a bazei de cunoștințe, sistemul putând să furnizeze rezultate ale analizei datelor introduse pentru date incomplete, prin ignorarea acelor care sunt în curs de specificare.

2.2 Analiza informațiilor

Sistemul dă posibilitatea expertului să-și vizualizeze rezultatele elicitării personale pe mai multe niveluri:

1. grila complet notată sub forma unei matrici care conține pe coloane exemplele și pe linii construcțiile (figura 1);
2. matricea distanțelor pentru exemple (figura 2) pe care o poate interpreta astfel: cu cât distanța este mai mică între două exemple, cu atât ele sunt mai similare din punctul de vedere al construcțiilor introduse. Expertul trebuie să urmărească ca setul lui de exemple să cuprindă exemple cât se poate de diferite, exemplele redundante trebuie să fie eliminate sau trebuie introduse noi atribute care să le diferențeze;
3. matricea distanțelor pentru atribute (figura 3). Expertul trebuie să urmărească în acest caz ca atributele să fie diferite. Astfel, două atribute prea apropiate se pot contopi într-unul singur, considerând unul de prisos sau pot fi lăsate amândouă, dar trebuie introduse exemple mai variate astfel încât să le diferențeze;
4. arborele exemplurilor/atributelor care sugerează o grupare a exemplurilor/atributelor după distanța între ele (figura 4). Reprezentarea grafică este foarte sugestivă în evidențierea grupelor de exemple/atribute asemănătoare;
5. grila ce conține valorile atributelor pentru exemple (figura 5);
6. arborele de implicații pe baza căruia se construiesc regulile de asociere, rezultate din analiza grilei (figura 12);
7. regulile de asociere, rezultate sub forma de reguli de producție (figura 13);
8. arborele de decizie, asociat grilei și unui atribut selectat ca scop al clasificării (figurile 16,17).

2.3 Derivare de noi cunoștințe

Derivarea de noi cunoștințe din analiza grilei se concretizează în obținerea de cunoștințe reprezentate sub formă de arbori de decizie, reguli descrieri de concepte.

Inducția de concepte (figura 14) se face pornind de la arborii construiți pentru exemple și, respectiv, pentru atribute. Expertul este invitat să coopereze la denumirea noilor concepte ce pot fi asociate grupelor de exemple sau de atribute. El are în permanență în față arborele și primește la selectarea unui nod informații care îl ajută să denumească nodul respectiv. De exemplu, la selectarea unui nod din arborele de exemple, expertul va fi informat care au fost exemplele care au generat acest nod și care sunt atributele cu valorile respective care sunt comune tuturor exemplor din clasa respectivă. Rezultatul este o ierarhie de concepte cu atât mai generale cu cât ne apropiem rădăcina arborelui. O astfel de clasificare poate fi utilă în domenii în care nu există clasificări, structurarea informației putând fi de mare ajutor expertului într-un domeniu.

Inducția de concepte pe arborele de clasificare a atributelor poate fi utilă când ne punem problema unor grile de mari dimensiuni pentru realizarea de griduri ierarhizate

Inducția de reguli (figurile 12, 13) se face pe baza analizei relațiilor dintre valorile atributelor pentru setul de exemple din domeniu introdus, cu alte cuvinte pe baza calculului implicațiilor dintre toate perechile atribut - valoare care apar în grilă. O astfel de analiză a implicațiilor pune în evidență noi aspecte în analiza calitativă a datelor :

- arată relația dintre atribute la un nivel de abstractizare mai înalt;
- măsoară dependența dintre atributele domeniului;
- ajută la construirea unei rețele și la derivarea pe această bază a regulilor de producție aferente, însoțite de factor de incertitudine.

Rețeaua de implicații este supusă judecății expertului care hotărăște dacă este de acord sau nu cu implicația generată. Un dezacord produce o revenire asupra notărilor din gridul inițial sau, dacă notările sunt corecte, problema se poate rezolva prin adăugarea unui nou exemplu care acoperă o astfel de situație de excepție. Punerea în evidență a unor implicații echivalente poate sugera expertului necesitatea introducerii de noi exemple care ar anula echivalența. În general, generarea implicațiilor și vizualizarea lor au efecte benefice asupra calității datelor din grid.

Sunt generați, de asemenea, **arbori de decizie** (figura 17), asociați unui atribut scop (figura 16), ales pentru clasificarea unui exemplu nou introdus.

Construirea arborelui de decizie permite eliminarea redundanțelor în dialogul cu utilizatorul care dorește o consultație.

2.4 Utilizarea cunoștințelor

Sistemul **ESYS** putea fi folosit pentru rezolvarea unor probleme de **consultanță** prin utilizarea ierarhiilor de exemple obținute pe baza aprecierilor calitative (prin notare), făcute de experți în faza de elicitare a cunoștințelor.

Sistemul de consultare se adresează, atât utilizatorilor nespecialiști în domeniu, care solicită o consultație în legătură cu un subiect, cât și experților, care au posibilitatea să introducă noi concepte prin subiectele propuse spre evaluare.

Pentru a se familiariza cu cunoștințele din domeniu, utilizatorul are acces la informații despre conținutul bazei de cunoștințe:

- informații despre **subiect** (nume, scopul elicitării, număr de experți care au elicitat, număr de exemple elicitate, număr de atribute);
- informații despre **atributele** specificate (numele atributului, procentul de notare întrunit luând în considerare toți experții și toate exemplele, valorile posibile ale atributului cu nota asociată fiecărei valori și cu procentul de notare întrunit de fiecare valoare);
- informații despre **exemplele** domeniului (numele exemplului, descrierea tuturor atributelor care i-au fost asociate, fiecare atribut apare cu valoarea sa, cu nota și procentul de consens întrunit în timpul elicitării).

Utilizatorul poate să-și selecteze atributele care îl interesează în legătură cu subiectul propus și, de asemenea, să selecteze anumiți experți a caror opinie îl interesează (figura 7).

Consultația propriu-zisă se poate face în două moduri:

1. utilizatorul nu propune un subiect, ci îl interesează exemplele cele mai apropiate sau cele mai departate de un exemplu fictiv, numit de sistem "optimal" deoarece i s-au atașat notele maxime la atributele selectate de utilizator;

2. utilizatorul propune un subiect ("ideal" pentru el) pe care îl notează pentru toate atributele pe care le-a selectat și obține lista exemplor domeniului, ordonată după apropierea față de subiectul său.

Sistemul are două modalități de lucru în ceea ce privește luarea în considerare a opiniei experților:

1. ia în considerare nota care a întrunit consensul experților (procent de notare maxim);

2. ia în considerare o notă medie a notelor acordate de toți experții care au elicitat.

În vizualizarea rezultatelor unei consultații apar ambele variante (figura 9). Consultația se poate considera că are succes dacă utilizatorul găsește în domeniu un exemplu care este apropiat cu mai mult de 50% de subiectul propus; dacă acest procent este mai mic, utilizatorul poate modifica notarea pentru subiectul sau încercând să obțină un procent mai mare de apropiere.

În concluzie, componenta de consultanță a lui ESYS se poate folosi pentru:

- luarea unei decizii multicriteriale - de exemplu, la alegerea "*locului de muncă*". Un loc de muncă este caracterizat de următoarele atribute: "*salariu*", "*condiții de muncă*", "*mod de promovare*", "*caracterul șefului*", "*program de lucru*", "*specific tehnic*", "*specific de conducere*", "*vacanță*", "*amplasare*". Un utilizator poate specifica și nota atributele care îl interesează și va obține drept răspuns care este "*locul de muncă*" cel mai apropiat de opțiunea sa.
- pentru adăugarea de noi concepte în domeniu - de exemplu, în baza de cunoștințe sunt descrise elemente din domeniul "*instrumente software pentru construirea de sisteme expert*"; un expert propune ca subiect un subdomeniu al domeniului și anume, *instrumentele necesare pentru construirea unor sisteme expert cu specific de planning*; expertul alege din atributele din domeniu numai atributele care consideră că interesează acest subdomeniu, de exemplu: "*mediu de dezvoltare al programelor bun*", "*disponibilitatea pe o varietate hardware*", "*interfața utilizator bună*" etc; răspunsul obținut va indica instrumentele care sunt potrivite pentru acest gen de sisteme expert cu planning, dar conceptul nou introdus de utilizator "*instrumente pentru sisteme expert cu planning*" poate fi reținut în baza de cunoștințe și folosit ulterior în consultații de genul "*un instrument are anumite caracteristici*" "*ce fel de instrument este*" și "*care sunt instrumentele din baza de cunoștințe, care sunt mai apropiate de descrierea utilizatorului*".

Problemele tipice care se pot rezolva cu acest sistem sunt, de exemplu, problemele de diagnostic într-un anumit domeniu. Într-o etapă ulterioară,

aceste descrieri de cazuri vor putea fi încărcate dintr-o bază de date. Sistemul ETS, ulterior devenit AQUINAS [2] a fost folosit inițial ca "front - end" pentru sistemul TEIRESIAS [11] al sistemului expert EMYCYN. Experții aveau nevoie de o rapidă prototipizare și ETS i-a ajutat în acest sens oferind tehnicile de elicitare triadică pentru identificarea atributelor și generarea regulilor de producție și a arborelui de decizie care stau la baza mecanismului inferențial. Un argument în plus pentru utilizarea cu succes a lui ETS a fost faptul că era un instrument cu o interfață grafică, ușor de utilizat. În dezvoltarea sistemului ESYS s-a încercat implementarea cam a aceleiași facilități ca și sistemul ETS.

Firma Boeing's Advanced Tehnologie Center, care a produs ETS și apoi AQUINAS, a folosit cu succes un astfel de instrument în construirea a sute de sisteme expert pentru aplicații de diagnostic cum ar fi: identificarea defecțiunilor la un avion, diagnosticare în controlul zborurilor, consultanță în alegerea compoziției materialelor, consultanță în alegerea tehnologiilor materialelor, consultanță în alegerea sistemului de propulsie etc.

Avantajele folosirii unui astfel de instrument sunt:

1. rapidă prototipizare, construirea și validarea rapidă a unui prototip de sistem expert fiind vitală în dezvoltarea lui ulterioară deoarece validarea lui poate fi făcută prin utilizare;
2. permite realizarea unor analize de fezabilitate rapide, pentru diferite proiecte de sisteme expert potențiale;
3. rapida achiziție a vocabularului unui domeniu;
4. întreținerea mai simplă a sistemului deoarece este mai ușor să întreții o grilă sau o bază de date decât un sistem de reguli;
5. bazele de cunoștințe produse cu ESYS pot fi combinate cu alte informații existente deja în alte baze de cunoștințe în cazul în care se dorește adăugarea de componente noi sistemului;
6. este un bun sistem de învățare deoarece este mai ușor să explici conceptele sistemului expert prin exemple.

3. Tehnica grilelor repertoriu

O grilă repertoriu conține trei componente: "elementele" sau "exemplele" care definesc domeniul grilei, "construcțiile" care reprezintă modalitățile în care expertul grupează și diferențiază elementele și un "mecanism de legare", care arată cum fiecare element este evaluat de fiecare construcție. O construcție are doi poli, două caracteristici care din punctul de vedere al

expertului sunt opuse unul altuia (un pol pozitiv și unul negativ).

O grila repertoriu este un instrument foarte folositor, care oferă multă **flexibilitate** în conceperea ei, cât și în cadrul aplicației în care este folosită și, mai mult, de obicei flexibilitatea ei este foarte stimulativă pentru utilizator. Elementele unei grile repertoriu trebuie să îndeplinească două cerințe: elementele trebuie se fie **omogene** (însemnând că trebuie să fie luate din același domeniu) și trebuie să fie o acoperire reprezentativă pentru domeniul investigat. De asemenea, este foarte important de a include în grilă alături de dimensiunea pozitivă, pe cea negativă, și o modalitate de a proceda astfel este de a include perechi de **elemente opuse**. Mai mult, dacă o aceeași grilă trebuie să fie completată de un grup de persoane (SOCIOGRIDS [20]), toate persoanele trebuie să aibă legătură cu domeniul exemplificat de elementele grilei.

Unui subiect i se pot atașa mai multe grile, fie ca rezultat al cooperării mai multor experți, fie ca rezultat al elicitării aceluiași expert în momentele diferite de timp sau în ipostaze diferite. În acest caz, mulțimea exemplurilor (elementelor) și, respectiv, mulțimea construcțiilor (atributelor) sunt comune tuturor grilelor introduse. Sarcina primului expert care este intervievat în legătură cu subiectul ales este cea mai dificilă deoarece el pornește de la zero, în timp ce, ceilalți experți au posibilitatea să lucreze pe exemplele și atributele introduse de experții care au elicitat anterior cu posibilitatea, evident, de a adauga, modifica, șterge exemple și / sau construcții neconvenabile pe grila proprie.

Tehnicile de elicitare **multi-expert** a grilelor, bazate pe **teoria construcțiilor personale** conduc la obținerea unor grile de note care exprimă aprecierea calitativă a experților pe o scală de notare, motiv pentru care este posibilă compararea lor.

Grila poate fi extrasă, de asemenea, dintr-o **bază de date** clasică (figura 10), caz în care se lucrează cu o singură grilă; metoda este adecvată când există o astfel de bază de date care descrie exemple concrete dintr-un domeniu, și nu părerile subiective ale unor experți despre un anumit domeniu.

În cazul în care există mai multe grile asociate la un subiect, se pot obține rezultate interesante selectând o grilă de lucru, adică o grilă obținută din toate grilele subiectului sau dintr-un număr de grile selectate, caz în care nota respectivă va fi calculată după dorință, fie ca aceea care a întrunit maximumul de consens de la toți experții, fie că o medie a notelor acordate de toți experții. Selecția acestei opțiuni dă posibilitatea activării unei componente de **consultanță**.

Pentru a înțelege utilitatea ei, vom prezenta câteva considerații privind **analiza datelor, bazată pe grupări ierarhizate (clusters)**.

Scopul unei metode de analiză ar trebui să fie evidențierea trăsăturilor importante ale grilei, care ar putea să nu fie evidente la prima vedere pentru un utilizator care nu are multă experiență în analizarea datelor.

Metoda de analiză a grilelor, utilizată în sistemul **ESYS**, este metoda analizei grupurilor ierarhizate. Aceasta scoate în evidență grupuri de elemente sau construcții similare și construiește doi arbori care ilustrează diferențele între asemenea grupuri. O asemenea analiză începe prin calcularea distanțelor între toate perechile posibile de elemente sau construcții și are drept rezultat un arbore care reflectă similitudinile și diferențele care există între elemente sau construcții.

Analiza grupurilor ierarhizate furnizează totuși putine rezultate dacă este executată separat pe elemente sau pe construcții. Adăugarea, în versiunea actuală, a determinării **grupurilor de date** a eliminat dezavantajul de a nu evidenția interacțiunile între grupuri de elemente și grupuri de construcții. Sistemul pune în evidență aceste grupuri prin colorarea diferită a celulelor matricii, care corespund la la grupuri de exemple cu atribute asemănătoare. Excepțiile dintr-un grup ies în evidență prin faptul că sunt colorate diferit. Analiza grupurilor de date compensează absența analizei componentelor principale.

Metoda grupurilor ierarhizate are în plus un avantaj față de metoda componentelor principale, avantaj care este utilizat în continuarea analizei grilei pe un plan superior prin inducția de reguli și de concepte. Cei doi arbori, al elementelor și al construcțiilor, se pot analiza separat, dar și împreună. Ei pot fi utilizați pentru a rearanja liniile și coloanele grilei inițiale astfel încât elementele și construcțiile care sunt similare să apară grupate în grilă.

Analiza elementelor necesită mai întâi calcularea diferențelor între toate perechile posibile de elemente. Pentru măsurarea distanțelor între elemente/construcții o tehnică uzuală este de a calcula **distanța euclidiană/coeficientul de corelație** între ele. Distanțele vor fi calculate ignorând intrările lipsă. Această metodă este potrivită și pentru **grile dihotomice**. Pentru grila din figura 1, a fost obținută **matricea de distanțe** pentru elemente/ construcții din figurile 2,3 și doi arbori laterali ce reprezintă grupurile și legăturile între ele (apar în figura 4), atașat matricii reordonate. Fiecare frunză a arborelui este un element, iar fiecare nod interior al său este un grup de elemente. Legătura între două noduri se face la un anumit nivel dacă există în subarborii denumiți de cele două noduri două exemple situate la o distanță mai mică sau egală cu nivelul curent. Condiția de apartenență la un grup de elemente

este că distanța dintre toate elementele grupului să fie mai mică sau egală decât distanța corespunzătoare nivelului nodului. Cei doi arbori laterali ne dau un rezumat util al caracteristicilor principale ale grilei. Vom observa acum și interacțiunile în corpul grilei, care poate să confirme sau să ne facă să modificăm interpretarea bazată pe arborii laterali (figura 4).

Primul stadiu în analiza interacțiunilor este de a reordona liniile și coloanele grilei pentru a se alinia la cei doi arbori laterali. Pentru a simplifica descrierea analizei interacțiunilor și pentru a face structura grilei mai evidentă, trebuie să ne asigurăm că sunt pozitive toate corelațiile între construcții din fiecare grupare. Aceasta implică inversarea polilor unor construcții pentru a schimba semnul corelațiilor cu alte construcții.

4. Inducția de reguli

Inducția de reguli se face pe grila curent selectată, indiferent de proveniența ei (grila introdusă de expert, extrasă dintr-o bază de date sau grila de lucru). Pentru ca regulile și conceptele să fie sugestive trebuie însă ca în cazul grilelor extrase de la expert, să fi fost efectuată rafinarea construcțiilor și transformarea lor în descrieri de atribute complete. Pentru grilele extrase din baza de date, această rafinare se face la activarea selecției și rafinării de atribute din baza de date. Pentru a înțelege natura acestor reguli pe care ESYS le extrage din analiza logică a grilei, se vor prezenta, în continuare, câteva considerații privind acest subiect.

Există o altă posibilitate de a interpreta un grid, nu ca o matrice de relații între o mulțime de elemente dintr-un domeniu și un set de construcții (atribute) care descriu aceste elemente. Astfel, dacă se presupune exemplul mai simplu al construcțiilor bipolare (atributele respective pot lua doar două valori - 0 și 1), se poate interpreta gridul ca un set de asignări de valori de adevăr la predicate logice. Se poate considera polul negativ al fiecărei construcții ca fiind un predicat logic, care se aplică la un element din domeniu care întoarce valoarea T (true), dacă valoarea specificată în grid la elementul și construcția respectivă este 0, și F (false) dacă valoarea respectivă este 1.

Pentru o grilă se determină relațiile dintre toate propozițiile ce se pot construi pornind de la valorile asignate prin elicitație și se construiesc mai multe rețele care pun în evidență toate implicațiile dintre propozițiile rezultate. Pot rezulta mai multe rețele deoarece, între anumite propoziții, nu există nici o legătură. O propoziție apare o singură dată în mulțimea de rețele produsă.

Figura 12 arată implicațiile derivate pentru grila din figura 1; implicațiile sunt evidențiate prin desenarea unui graf direcționat, ținându-se

cont de proprietatea de tranzitivitate a operației de implicație. Deasemenea, sunt puse în evidență relațiile de echivalență acolo unde ele apar printr-o dublă implicație.

Vizualizarea acestor implicații aduce un plus de informații calitative celui care studiază cunoștințele dintr-un domeniu, dar ea dă posibilitate construirii de reguli (figura 13) atașate domeniului care pot fi încărcate într-o bază de cunoștințe și utilizate într-un sistem expert.

ESYS lucrează cu grile care acceptă nu numai scala bipolară, dimensiunea scalei de notare a construcțiilor putând fi definită de utilizator ca o scală cu N puncte, de regulă N fiind un număr impar, pentru a permite definirea unui punct "neutru" în mijlocul scalei. O astfel de abordare a fost necesară, pentru a permite definirea așa ziselor nuanțe de gri între cei doi poli ai construcției care-ți permit exprimarea numai de judecăți în alb-negru. De aceea, a fost necesară extinderea semnificației propozițiilor astfel încât să se poată trata grile în care apar nu numai două valori. Pentru a înțelege mai bine contextul problemei, se reamintește că, acolo unde se folosește o scală multipolară, utilizatorului care elicită i se cere, în afară de cei doi poli extremi ai construcției și de notele acordate pe scala cerută, o continuare a descrierii construcției. Rafinarea construcției inițiale produce un nume semnificativ asociat tuturor notelor aflate pe scală între cei doi poli extremi (figura 9).

Astfel, propoziția "*elementul E este asignat polului negativ al construcției n*" în propoziția "*elementul E are valoarea x în construcția n*" sau, mai semnificativ, "*atributul n are valoarea x pentru elementul E*", unde atributul este numele unui concept de tip atribut pe care utilizatorul îl asociază construcției. Aceasta dă posibilitatea construirii de reguli sugestive pentru un utilizator. Într-o implicație, o valoare a unui atribut sau un "pol" al unei construcții implică o altă valoare a altui atribut sau a altui "pol" al altei construcții. Rafinarea construcțiilor în ESYS permite transformarea unei construcții într-o descriere de atribut care are atașată o descriere de cazuri sau fațete posibile ale valorii atributului (figura 11).

Astfel, implicația exprimată în termeni de "poli" ai construcției: "*SCUMP → DEPARTE*" poate fi transformată în regula: "*If COST = SCUMP then DISTANTA = DEPARTE endif; (100%)*". Deci, aceste reguli pun în evidență legătura dintre atribute.

Între atributele declarate ale domeniului trebuie să apară și un atribut care exprimă scopul problemei de rezolvat deoarece sistemul de reguli generat va fi folosit ca să răspundă unor consultații cerute de utilizatorul final. Acesta va putea să specifice condițiile care îl interesează, de exemplu, simptomele unei boli și va obține răspuns în legătură cu boala de care este suspectat și

tratamentul care trebuie să-l aplice. În acest exemplu, boala și tratamentul sunt astfel de atribute scop. Revenind la exemplul de mai sus, dacă se introduce atributul *SOLUTIE* se vor putea obține reguli care să arate legătura valorilor celorlalte atribute cu atributul scop *SOLUTIE*. De exemplu, dacă soluția *PARIS* a fost notată cu 5 pentru atributul *DISTANTA (APROAPE(1) - DEPARTE(5))*, atunci următoarea regulă va putea fi generată: "If *DISTANTA = DEPARTE* then *SOLUTIA = PARIS endif; (100%)*

S-a subliniat proprietatea de asimetrie a relației de implicație. Sunt interesante de studiat și relațiile dintre două construcții pentru care nu este satisfăcută relația de incluziune strictă dintre mulțimile de valori corespunzătoare, dar pentru care intersecția celor două mulțimi nu este vidă. În această situație, se poate determina așa zisa "putere a implicației", calculată pe baza probabilității de apariție a implicației.

O astfel de măsură este necesară deoarece dacă mai multe detalii referitoare la analiza grilei. Deși nu satisface criteriul de a menține analiza la nivelul datelor introduse de utilizator, care ar putea să fie nesemnificative pentru persoana care a generat grila, aceste date devin importante atunci când se pune problema generării de reguli cu factori de incertitudine din astfel de rețele de implicații.

Sistemul oferă expertului posibilitatea să examineze regulile induse din analiza grilei, atât sub formă unor rețele de implicații, cât și sub formă de reguli de producție cu factori de incertitudine. Utilizatorul are diverse posibilități de corectare a regulilor dacă nu i se par convenabile, fie prin ștergere/adăugare/modificare de reguli, fie prin modificarea directă a grilei pentru exemplele care au generat regula neconvenabilă.

5. Inducția de concepte

Pentru a înțelege mai bine natura problemelor ce pot fi rezolvate în acest mod este binevenită o comparație cu activitatea de diagnosticare a unui medic. Modul în care un medic rezolvă o problemă de diagnostic seamănă destul de bine cu un sistem incremental de formare a conceptelor. Un medic vede un pacient zilnic (incremental). Sarcina medicului este de a da un diagnostic corect și de a găsi un tratament adecvat. Cu fiecare caz tratat, medicul colectează o mulțime de informații pentru a-și îndeplini sarcina cu succes. Pentru diagnostic el face apel la experiența sa trecută și la informațiile noi, care descriu cazul de tratat. În consecință, medicul este pus în fața sarcinii de a acumula în memoria sa, informații care să-i permită regăsirea eficientă a conceptelor care au legătură cu situația de tratat.

Formarea conceptelor constă dintr-un proces incremental de construire a unei ierarhii de descrieri de concepte care caracterizează obiectele dintr-un domeniu. Un sistem care formează o ierarhie de descrieri de concepte poate fi un instrument utilizabil în organizarea, sumarizarea și în memorarea datelor complexe. Același sistem poate fi folosit ca un sistem de regăsire, capabil să prezică proprietățile unor obiecte noi. Un astfel de sistem poate fi atractiv pentru domeniile în care clasificarea nu există, cunoștințele sunt incomplete sau experții umani sunt puțini. Un astfel de sistem (dacă este "user-friendly") ar putea deveni un instrument productiv pentru un profesionist, cum este medicul descris anterior.

Sistemul *ESYS* realizează clasificarea exemplilor dintr-un domeniu prelucrând grila de notare, care exprimă relațiile atribute-exemple. Arborele exemplilor se calculează pe baza evaluării distanțelor dintre exemple pentru toate atributele, distanță care exprimă gradul de similitudine dintre două exemple.

Nodurilor arborelui de exemple li se pot asocia concepte mai mult sau mai puțin specifice, depinzând de poziția nodului în arbore. Asocierea de concepte se poate face numai în colaborare cu expertul care este pus în fața unei ierarhii de clase, care inițial au doar un nume generat de sistem (figura 14). Arborele respectiv este vizualizat și expertul este invitat să selecteze unul din conceptele nedefinite și să încerce să-i asocieze un nume. În această acțiune, expertul are acces prin selectarea nodului la informații care să-l ajute să depisteze cu ușurință acest nou concept:

- lista exemplilor din domeniu, care aparțin acestei clase de obiecte;
- lista de perechi atribut - valoare, comune acestor exemple, împărțite în două categorii: specifice clasei și moștenite de la un concept mai general.

Observăm ca expertul nu trebuie decât să dea un nume sugestiv clasei respective, *ESYS* fiind responsabil de generarea descrierii clasei respective și a relațiilor de moștenire dintre clase (figura 14).

Odată completată o astfel de ierarhie, ea poate fi folosită în sine ca o clasificare a exemplilor din domeniul descris sau poate fi folosită pentru generarea unor frame-uri care țin cont de descrierile atributelor comune și de relația de moștenire ce rezultă din această clasificare.

6. Construirea de arbori de decizie

Pentru a înțelege modul în care pot fi folosiți arborii de decizie, se vor prezenta în continuare câteva considerații privind descrierea și generarea

lor. Generarea de arbori de decizie este o procedură de clasificare eficientă pentru un număr mare de exemple, care asigură extragerea de cunoștințe din cantități de date mari prin procesul de inferență inductivă.

Datele sunt exemple relativ nestructurate ale unui concept de o complexitate considerabilă. Informația care se determină este un mijloc de a identifica exemplele unui concept sau cu alte cuvinte o regulă de clasificare. Specific acestei proceduri este numărul mare de exemple utilizate în formarea conceptelor și eficiența regulilor de clasificare, descoperite în acest fel.

Procedura ia obiecte dintr-o clasă cunoscută, descrisă în termenii unei colecții fixate de proprietăți/atribute și produce un arbore de decizie pentru aceste atribute care clasifică în mod corect exemplele date. În acest scop, se folosește un mecanism simplu de comparare pentru descoperirea unei reguli de clasificare pentru o colecție de obiecte aparținând la două clase. Cum s-a menționat mai sus, fiecare obiect trebuie să fie descris în termenii unui set fixat de atribute, fiecare dintre atribute având un mic set de valori posibile, ca de exemplu, "culoare" poate fi {roșu, verde, albastru}.

O regulă de clasificare în forma unui arbore de decizie poate fi construită din orice colecție de astfel de obiecte C . Dacă C este vid, acesta va fi asociat în mod arbitrar la mai multe clase. Dacă toate obiectele din C aparțin aceleiași clase, atunci arborele de decizie este de fapt o frunză cu numele acestei clase. Altfel, C conține exemple din ambele clase; se selectează un atribut și se împarte C în mulțimi disjuncte C_1, C_2, \dots, C_n unde C_j conține acei membri din C care au a j -a valoare a atributului selectat.

Fiecare din aceste subcolecții sunt tratate pe rând, după aceeași strategie de formare a regulilor. Rezultatul este un arbore în care fiecare frunză poartă un nume de clasă și fiecare nod specifică atributul de testat, cu câte o ramură pentru fiecare valoare posibilă a atributului respectiv.

Această procedură de formare a regulii va merge, cu condiția să nu existe două obiecte care să aparțină la două clase diferite și să aibă valori identice pentru fiecare atribut; în astfel de cazuri, alegerea atributului nu este satisfăcătoare, și situația se poate îndrepta prin adăugarea unor atribute pentru care cele două obiecte devin diferite.

Totuși, este de dorit, în general, ca arborele să fie capabil să clasifice obiecte care n-au fost utilizate în construcția sa și, deci, frunzele corespunzătoare la un set vid de exemple (când o clasă este aleasă aleator) ar putea fi minimizate ca număr. Dacă se adoptă un algoritm simplu "selectează primul atribut pentru rădăcina arborelui, al doilea atribut pentru următorul nivel

ș.a.m.d", rezultatul va tinde către un arbore complet, cu o frunză pentru fiecare punct din spațiul atributelor, o situație total nesatisfăcătoare.

Cheia în acest stil de inducție constă în selectarea celui mai util atribut de testat astfel încât arborele de decizii final să fie minim.

7. Concluzii și dezvoltări ulterioare

Sistemul ESYS încearcă să reunească toate modalitățile de elicitare cunoscute, precum și toate metodele de analiză a grilelor, toate într-o manieră "user friendly", cât mai ușor, comod și benefic de utilizat. Fiind scris sub mediu Windows, beneficiază din plin de facilitățile grafice oferite, de noua modalitate de programare "event driven", de posibilitatea de a lucra practic cu nenumarate ferestre la un moment dat pe ecran. În plus, beneficiază din plin și eficient de facilitățile oferite de limbajul orientat pe obiecte ACTOR, care permite rapida prototipizare și extinderea cu ușurință a sistemului. Din motive de eficiență, autorii vor produce o versiune a sistemului în C++.

Din păcate nu a beneficiat de o valorificare, în sensul folosirii practice de către un utilizator interesat, ceea ce ar fi condus la o dezvoltare mai orientată de necesitățile experților, dar se speră că, realizarea interfeței cu fișiere tip baze de date clasice (DBASE, PARADOX) și, respectiv, a interfeței cu shell-ul M.4, să-i deschidă alte posibilități.

ESYS poate fi dezvoltat în continuare pentru a deveni un mediu grafic cât mai complet pentru elicitarea cunoștințelor dintr-un domeniu. În continuare, se vor urmări:

- transformarea sa într-un sistem de achiziție de cunoștințe mai general, primul pas fiind făcut în această fază prin introducerea reprezentărilor grafice, asociate arborilor și grafurilor de care a fost nevoie pentru vizualizarea regulilor și a conceptelor induse. Alte posibile paradigme de reprezentare a cunoștințelor ce se pot implementa pot fi obiecte ca frame-urile, restricțiile, descrierile de reguli pe bază de grile, de descriere de priorități și preferințe, generarea de chestionare etc.;
- introducerea descrierii de obiecte ca frame-uri ar putea fi legată de introducerea unui algoritm de grupare conceptuală a exemplelor, care ar elimina dezavantajul manevrării de descrieri de obiecte nestructurate;
- reevaluarea algoritmilor folosiți în contextul folosirii unui număr mare de exemple și studierea numărului

semnificativ de exemple care ar fi necesar pentru inducția cu o probabilitate cât mai mare de reguli corecte.

Bibliografie

1. **BOOSE, J.H.** : Rapid Acquisition and Combination of Knowledge from Multiple Experts in the Same Domain. În: IEEE, 1985, pp. 461-467.
2. **BOOSE, J.H. BRADSHAW, J.M.** : From ETS to ACQUINAS: Six years of knowledge acquisition tool development, EKAW, July 1989.
3. **CARBONELL, J.G., MICHALSKI, R. S., MITCHELL, T. M.** : An Overview of Machine Learning, Machine Learning - An Artificial Intelligence Approach, Springer Verlag Berlin, 1984.
4. **DIETRICH, G. T., MICHALSKI, R. S.** : A Comparative Review of Selected Methods for Learning from Examples, Machine Learning - An Artificial Intelligence Approach, Springer Verlag, Berlin, 1984.
5. **EASTERBY-SMITH, M.** : The Design, Analysis and Interpretation of Repertory Grids. În: International Journal Man-Machine Studies, 1980, 13, pp. 3-24.
6. **ESHRAGH, F.** : Subjective Multi-Criteria Decision Making. În: Int. Journal Man-Machine Studies, 1980, 13, pp. 117- 143.
7. **GAPPA, U.** : A Toolbox for Generating Graphical Knowledge Acquisition Environments. În: Proc. of The World Congress on Expert Systems, 1991, pp. 797-811.
8. **GAINES, B.R., SHAW, M.L.G.**: New Directions in the Analysis and Interactive Elicitation of Personal Construct Systems. În: International Man-Machine Studies, 1980, 13, pp. 81-117.
9. **GAINES, B.R.**: Knowledge Acquisition: The Continuum Linking Machine Learning and Expertise Transfer, EKAW, July 1989, pp.
10. **DUPAS, R, MILLOT P.** : A Learning System for Preventive Maintenance, EKAW, Paris, July 1989, .
11. **DAVIS, R., LENAT, D. G.**: Knowledge-Based Systems in Artificial Intelligence, McGraw-Hill Int. Book Comp., 1982.
12. **HADZIKADIC, M., ELIA, P., WILSON, R.**: Desining a Diagnostic Knowledge-Based System by a Novice. În: Proc. of The World Congress on Expert Systems, 1991, pp. 1450-1457.
13. **KEEN, T.R., BELL, R.C.** : One Thing Leads to Another: A New Approach to Elicitation in the Repertory Grid Technique. În: International Man-Machine Studies, 1980, 13, pp. 25-39.
14. **KELLY, G.A.** : The Psychology of Personal Constructs, New York, 1955.
15. **LEACH, C.** : Direct Analysis of Repertory Grid. În: International Man-Machine Studies 1980, 13, pp. 151-167.
16. **LINSTER M.**: Integrating Acquisition, Representation and Interpretation EKAW, July 1989.
17. **USER'S GUIDE - A Guide to Developing and Delivering Knowledge Systems**, 1992.
18. **MICHALSKI, R.S.**: Learning from observation: Conceptual Clustering, Machine Learning - An Artificial Intelligence Approach, Springer Verlag, Berlin, 1984.
19. **QUINLAN, J. R.**: Learning Efficient Classification Procedures and their Application to Chess End Games, Machine Learning - An Artificial Intelligence Approach, Springer Verlag, Berlin, 1984.
20. **POPE, M.L., SHAW, M.L.G.** : Personal Construct Psychology in Education and Learning. În: International Journal Man-Machine Studies , 1981, 2, pp. 223-233.
21. **PORTMAN, M.M., EARTERBROOK, S.M.**: PMI: Knowledge Elicitation and De Bono's Thinking Tools, Current Developments in Knowledge Acquisition - EKAW'92, 1992
22. **HAYES-ROTH, F.** : Using Proofs and Refutations to Learn from Experience, Machine Learning - An Artificial Intelligence Approach, Springer-Verlag, Berlin, 1984.
23. **SHAW, M.L.G.** : An Interactive Knowledge-Based System for Group Problem Solving. În: IEEE, 1988.
24. **SHAW, M.L.G., GAINES, B.R.**: Knowledge Acquisition: Some Foundation. În: Manual Methods and Future Trends, EKAW, July 1989.
25. **SHAW, M.L.G., GAINES, B.R.** : Comparing Conceptual Structures: Consensus, Conflict, Correspondence and Contrast, Knowledge Acquisition ,1989, 1, pp. 341-363.
26. **SHAW, M.L.G., THOMAS, L.E.**: FOCUS and Educations - An Interactive Computer System for the Development and Analysis of Repertory Grids. În: International Journal Man-Machine Studies, 1980, 10, pp. 139-173.
27. **SLATER, P.**: Cluster Analysis Versus Principle Component Analysis, A Reply to

Electrical Engineering Dept. În: British Journal of Social and Clinical Psychology 13, 1974, pp. 427-430.

28. ȚILIVEA, D., STĂNCIOIU, C.: ESYS - instrument de achiziție și combinare a cunoștințelor de la mai mulți experți, extins cu un sistem de consultanță. În: Manual de referință. I.C.I.,1992.

29. ȚILIVEA, D., STĂNCIOIU, C.: ESYS : Interactive Acquisition from Multiple Experts Using Personal Construct Theory. În: Studies in Informatics and Control, Vol.2, No.2, June 1993.

8. ESYS - un sistem "user-friendly"

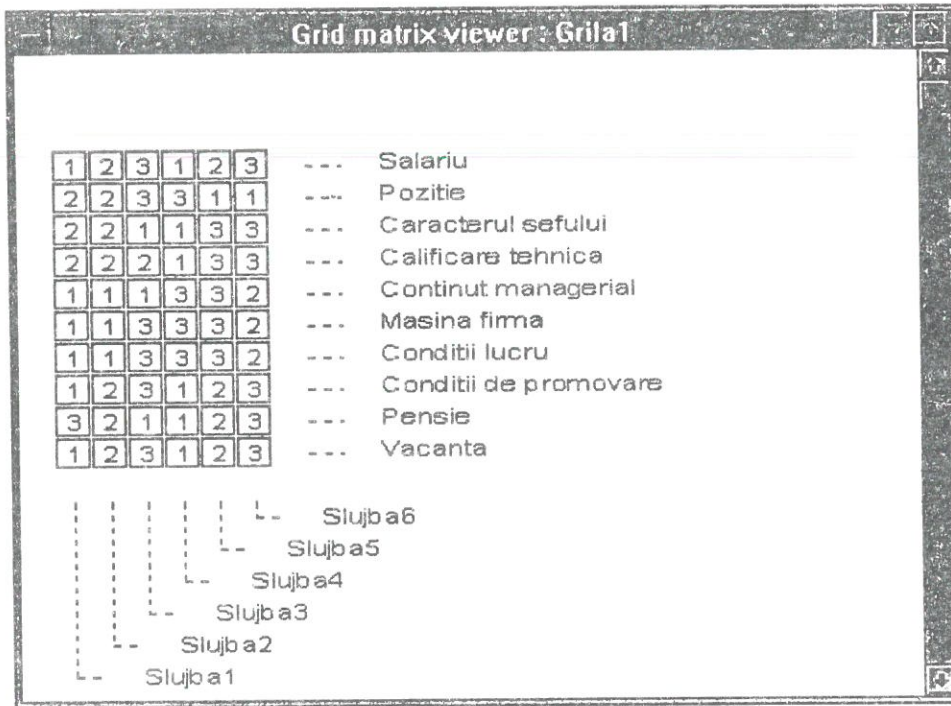


Figura 1. Grila inițială

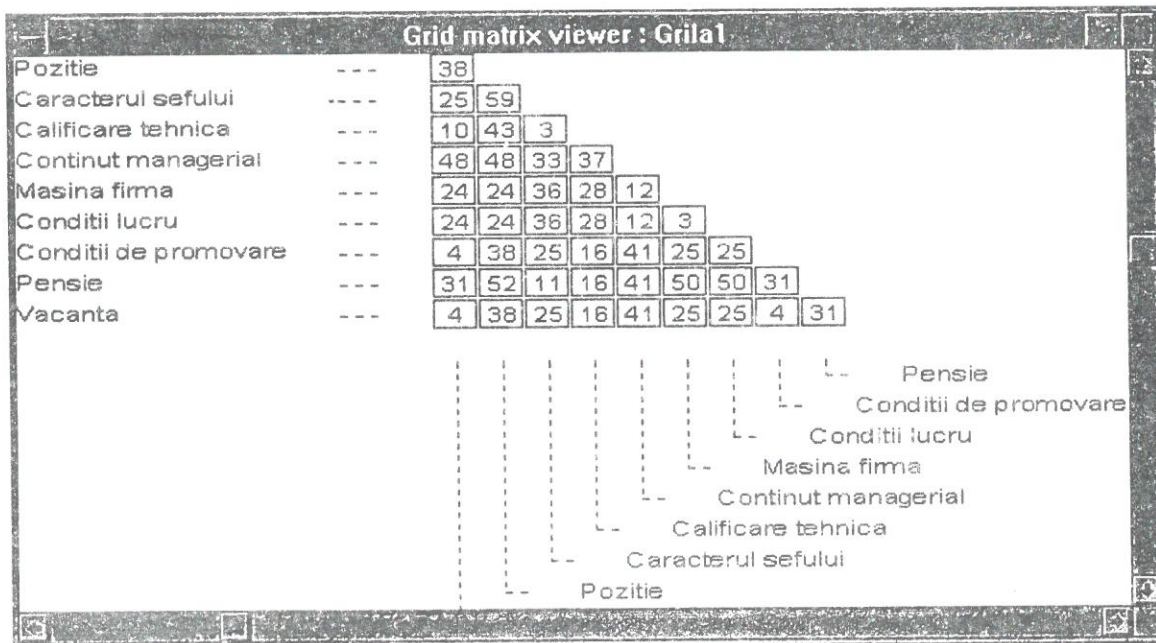


Figura 2. Matricea distanțelor pentru construcții

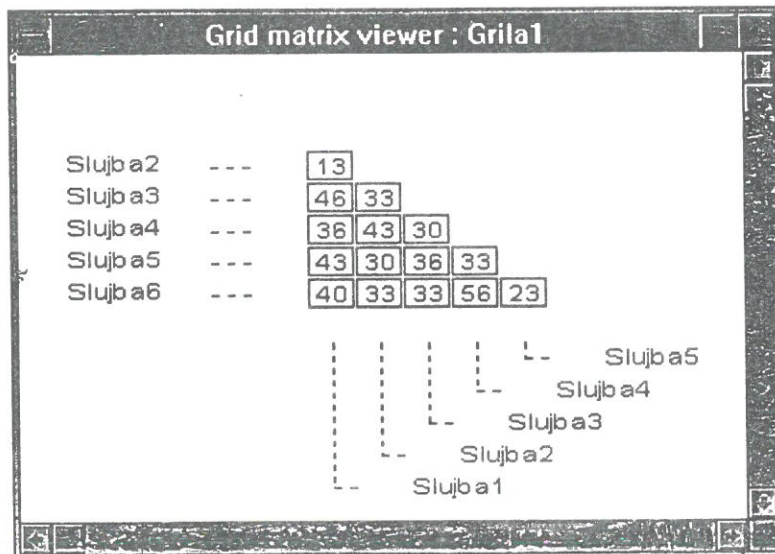


Figura 3. Matricea distanțelor pentru exemple

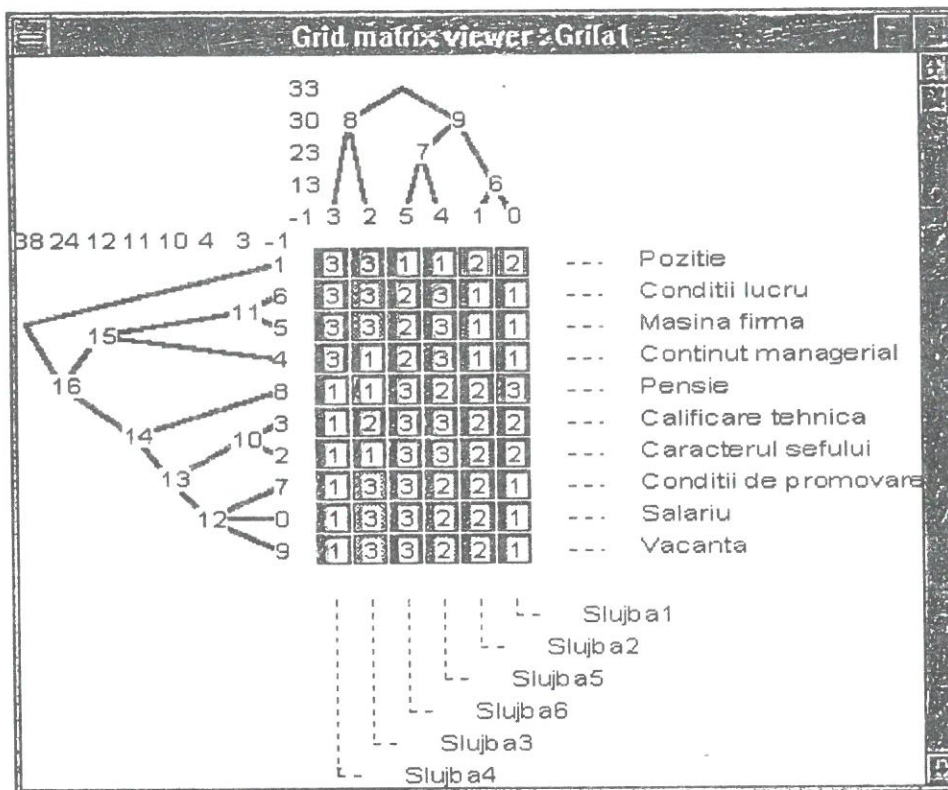


Figura 4. Grila reordonată și arborii marginali

Values grid						
Grid:	Grila1					
Number of Attributes:	10					
Number of Examples:	6					
	Slujba1	Slujba2	Slujba3	Slujba4	Slujba5	Slujba6
Salariu	mic	mediu	mare	mic	mediu	mare
Poziție	periferic	periferic	central	central	suburbic	suburbic
Caracterul sefuii	rigid	rigid	foarte rigid	foarte rigid	deschis	deschis
Calificarea tehnica	medie	medie	medie	fara	superioara	superioara
Continut managerial	fara	fara	fara	avansat	avansat	mediu
Masina firma	nu	nu	sigura	sigura	sigura	probabila
Conditii lucru	proaste	proaste	bune	bune	bune	medii
Conditii de promovare	nu	probabila	sigura	nu	probabila	sigura
Pensie	sigura	probabila	nu	nu	probabila	sigura
Vacanta	scurta	medie	lunga	scurta	medie	lunga

Figura 5. Grila valorilor atributelor după rafinarea construcțiilor

RATE WINDOW

Test
Editing the evaluations on the characteristics:
Salariu = mare Salariu = mic

Go To

Next

Previous

End

Exit

OK

Cancel

This display shows the ratings of all the grid Grila1 elements on one characteristic. You can click on any of the elements in the list box and drag them to the rating bar and put the top left corner of the window (without the bar, and drag them to a place where you would like it to be. You can change the rating moving the rating bar to any place and then begin the action described up. Click in this window if you want to remove the advice.

Salariu = mic

Slujba1
Slujba2
Slujba3

Salariu = mare

Figura 6. Notarea grafică a exemplelor introduse

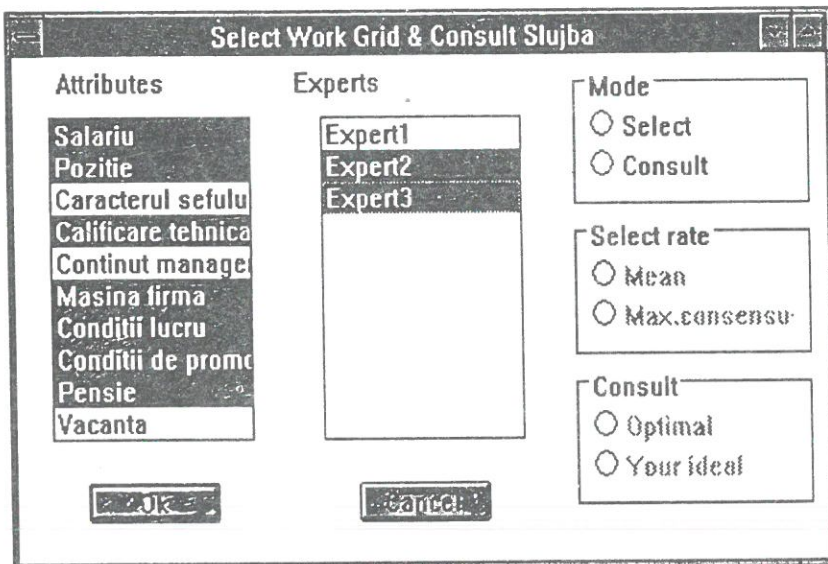


Figura 7. Selectarea atributelor și /sau experților pentru o grilă de lucru sau o consultație

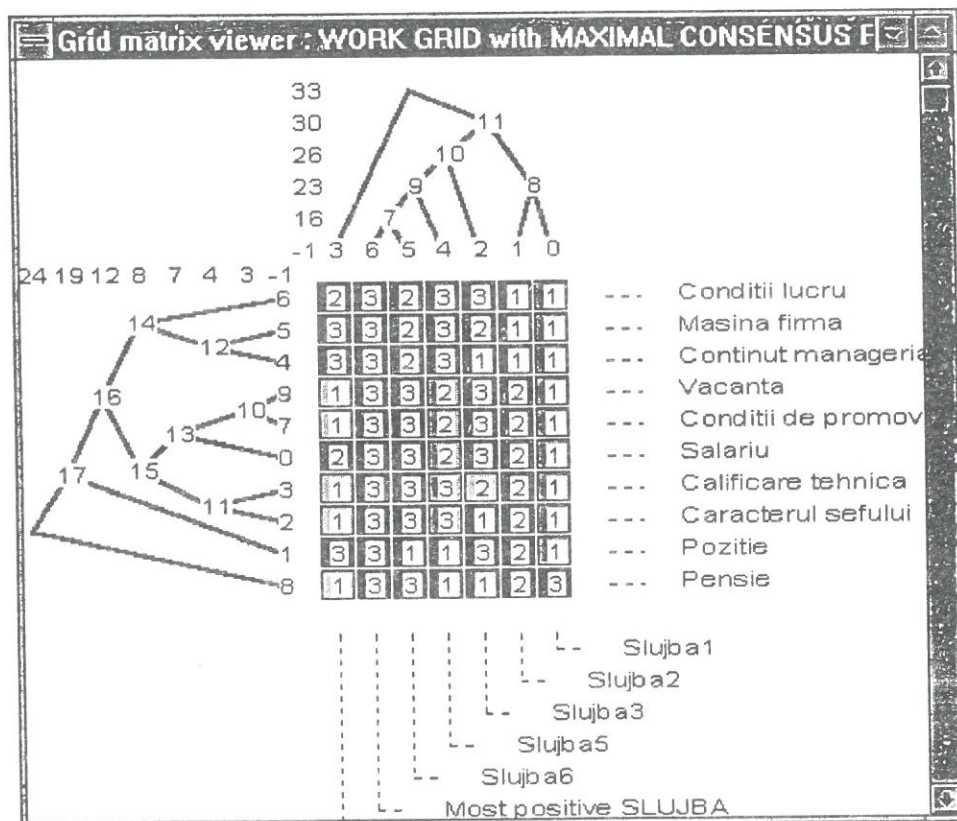


Figura 8. Grila de lucru obținută din notele care au întrunit maximumul de consens

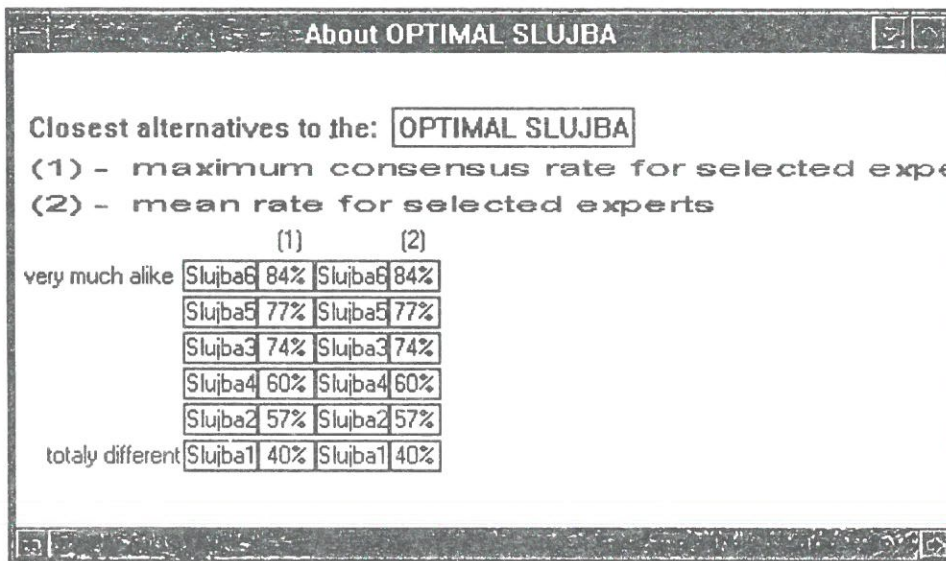


Figura 9. Rezultatul unei consultații

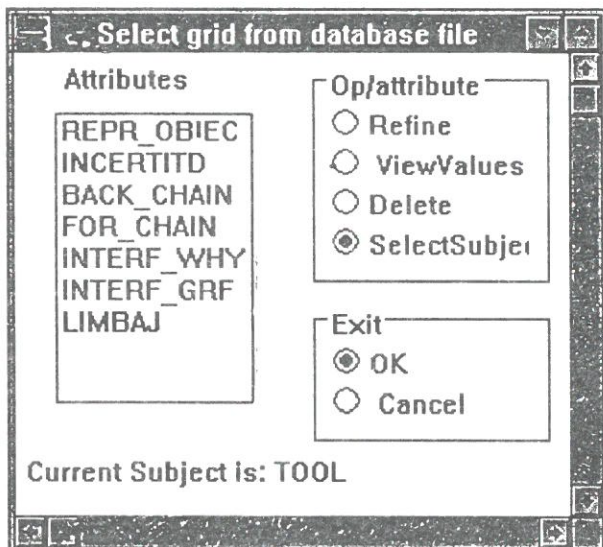


Figura 10. Selectarea informațiilor din baza de date

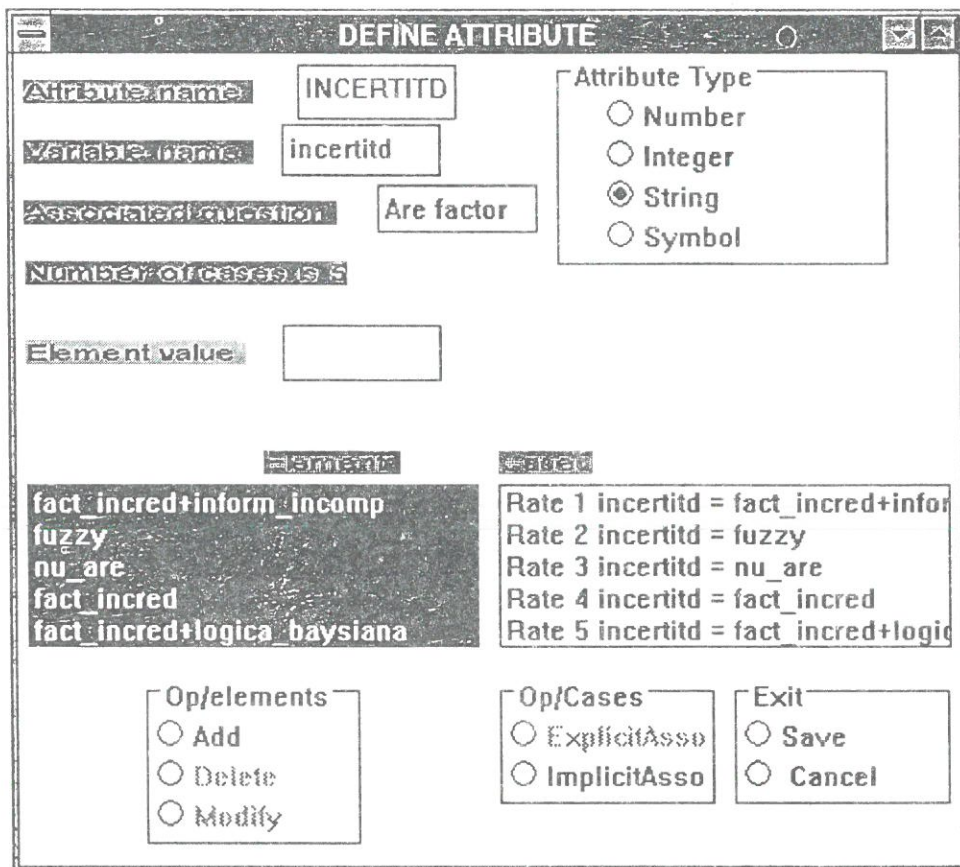


Figura 11. Rafinarea unei descrieri de atribut

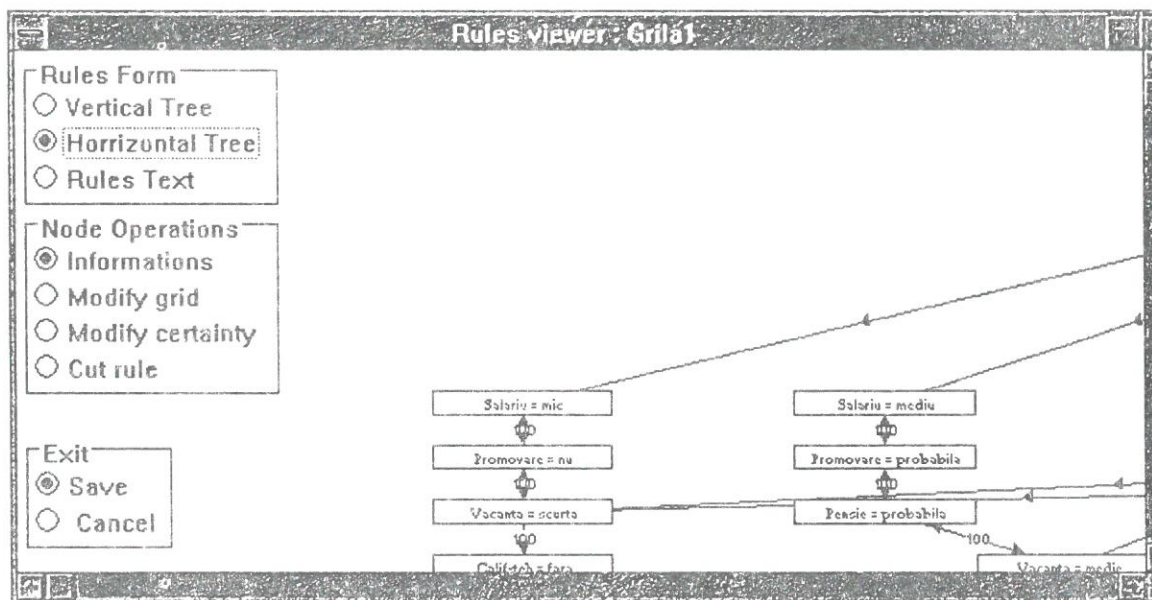


Figura 12. Rețeaua de implicații extrasă din analiza grilei

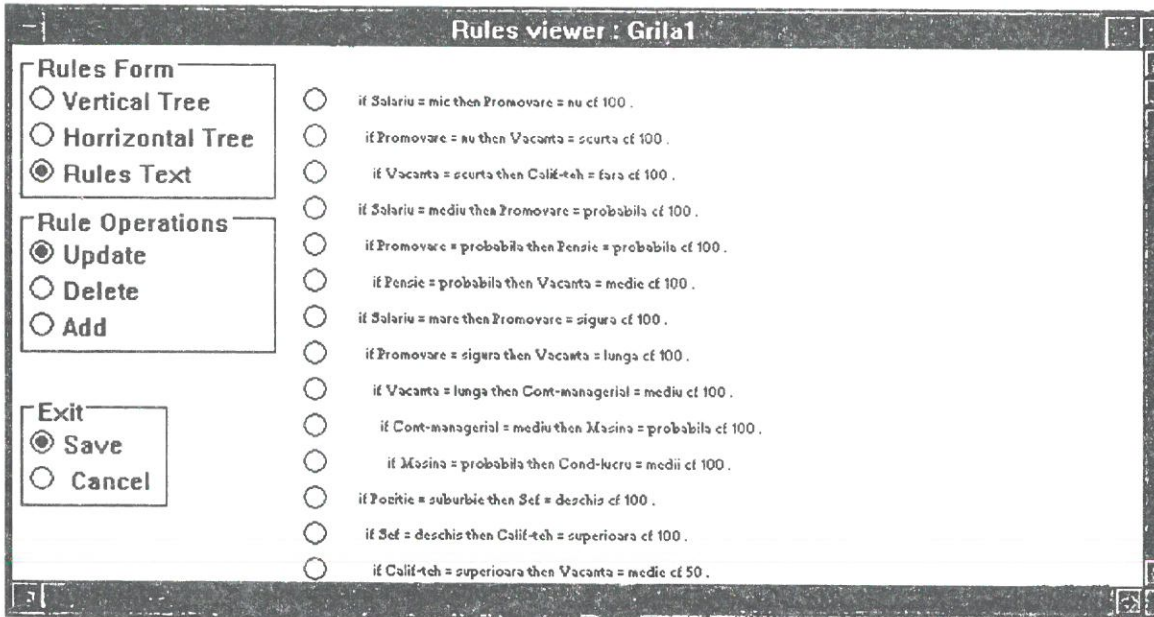


Figura 13. Reguli de producție extrase din analiza implicații

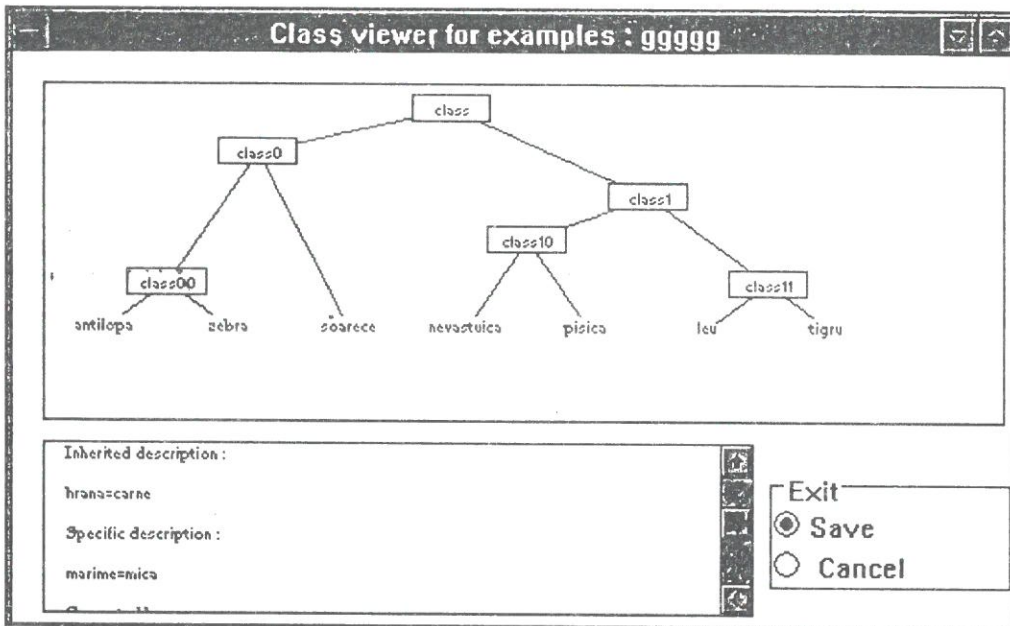


Figura 14. Ierarhia de clase și informațiile asociate unei clase selectate

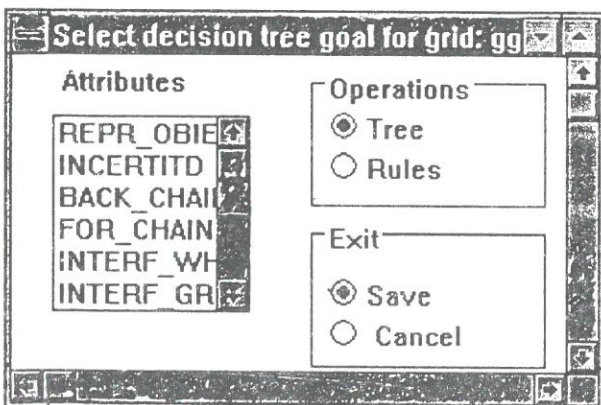


Figura 15. Selectează atributul scop al arborelui de decizie

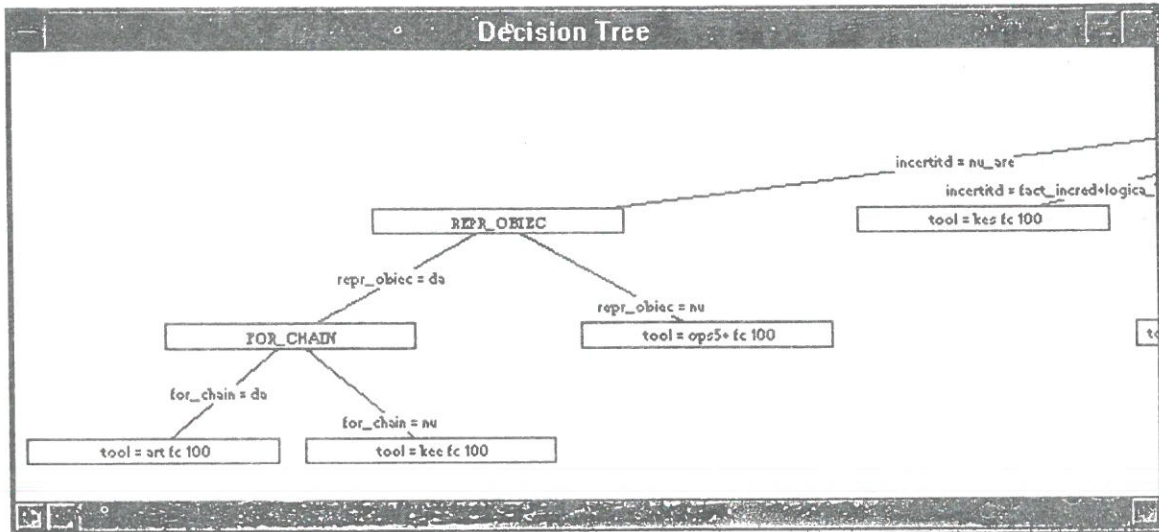


Figura 16. Arborele de decizie produs din analiza grilei

```

RulesFor decision tree
goal = tool
/* Attributes description */
question(repr_obiec) = 'REPR_OBIEC ? ' .
automaticmenu(repr_obiec) .
legals(repr_obiec) = automaticmenu(repr_obiec) .[da,nu] .
question(incertitd) = 'INCERTITD ? ' .
automaticmenu(incertitd) .
legals(incertitd) = automaticmenu(incertitd) .[fact_incred+inform_incomp,fuzzy,nu_are,fact_incr
question(back_chain) = 'BACK_CHAIN ? ' .
automaticmenu(back_chain) .
legals(back_chain) = automaticmenu(back_chain) .[da,nu] .
question(for_chain) = 'FOR_CHAIN ? ' .
automaticmenu(for_chain) .
legals(for_chain) = automaticmenu(for_chain) .[da,nu] .
question(interf_why) = 'INTERF_WHY ? ' .
automaticmenu(interf_why) .
legals(interf_why) = automaticmenu(interf_why) .[da,nu] .
question(interf_grf) = 'INTERF_GRF ? ' .
automaticmenu(interf_grf) .
legals(interf_grf) = automaticmenu(interf_grf) .[da,nu] .
question(dimbei) = 'DIMBEI ? '

```

Figura 17. Cunoștințele extrase dintr-un arbore de decizie în limbajul țintă M4