

Sistem expert pentru generarea unui playlist potrivit

Studenti:

Bujor Raluca

Pascu Monica-Nicoleta

Ştefan Gabriel Daniel

Coordonator:

Hristea Florentina

Cuprins

Descriere	3
Soluțiile sistemului	4
Atribute și arbori	9
Instrucțiuni pentru consolă	12
Exemple de interogări	15
Explicarea cerințelor	18
Concluzie	24

Descriere

Sistemele Expert au fost pentru prima dată introduse împreună cu conceptul de Inteligență Artificială. Acestea fac parte dintr-o gamă de instrumente indispensabile pentru realizarea de instrumente automate sau interactive capabile să realizeze sarcini complexe. Sistemele Expert pot fi considerate programe soft care operează într-un sistem sofisticat la fel cum o fac și experții umani. Cu alte cuvinte, acestea sunt aplicații software care imită judecata experților umani dintr-un domeniu bine conturat.

Un Sistem Expert este un "sfătuitor computerizat" care este capabil să raționeze (și nu doar să arate o enumerare de date), dar care se ocupă de obicei de un spațiu mic de cunoștințe, numite de altfel "domeniu de cunoștințe". Cunoștința pe care o posedă este activă deoarece ea poate fi folosită pentru a deduce noi informații din ceea ce este deja cunoscut despre o problemă. Având în vedere faptul că, cunoștința pe care o posedă un sistem expert tinde să fie legată de aplicații specifice, ele nu sunt așa de răspândite ca bazele de date, a căror cunoștință poate fi folosită în diferite aplicații. Cea mai des întâlnită reprezentare de cunoștințe la Sistemele Expert o reprezintă regulile.

Despre sistemul dezvoltat în scopul acestui proiect:




Creșterea în popularitate a serviciilor de streaming de muzică precum Spotify și Google Music a adus cu ea schimbări în felul în care utilizatorii consumă muzica. Dacă în era casetelor și a CD-urilor aceștia ascultau albume întregi, datorită noilor medii de difuzare a pieselor ce permit accesul la un catalog aproape interminabil de artiști și piese, aceștia și-au mutat atenția către listele de redare, fie construite de ei înșiși fie de alți utilizatori. Furnizorii de astfel de servicii au observat această tendință și au investit în sisteme care ajută utilizatori să descopere sau să își creeze liste de redare cât mai personalizate.

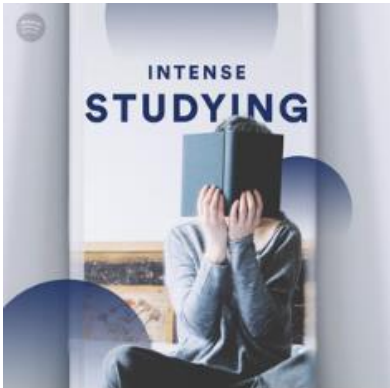


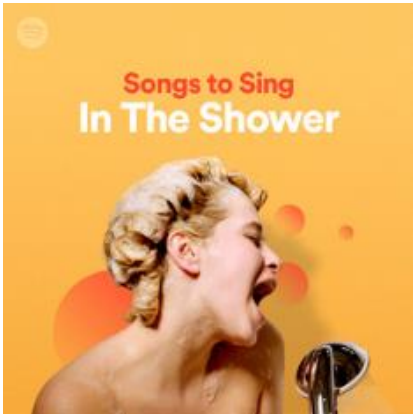
O astfel de funcționalitate se poate dovedi utilă atunci când dorim să avem un fundal muzical potrivit contextului curent, dar fără a fi necesare cunoștințe vaste când vine vorba de artiști și piesele lor. De exemplu, un astfel de sistem poate fi folosit de o cafenea cu o tematică specifică pentru a crește imersiunea experienței clienților săi.



Un serviciu de streaming de muzică (asemănător cu Deezer, Spotify, Google Music etc.) dorește să le pună la dispoziție utilizatorilor săi posibilitatea de a descoperi liste de redare pe baza stărilor lor emoționale. Acestea vor fi alese pe baza unui set de attribute precum: genul, starea emoțională, tematica, perioada (anii '90, '00, '10) etc. Sistemul sprijină utilizatorii în căutarea fundalului sonor potrivit, făcând o selecție din totalitatea listelor de redare existente pe platformă.

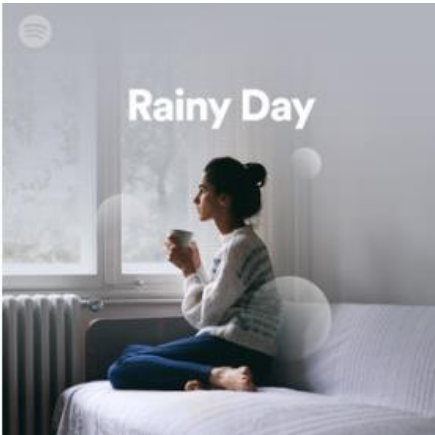



Soluțiile sistemului



În tabelul de mai jos sunt prezentate playlist-urile incluse în sistemul expert alături de o scurtă descriere și o lista de specificații.

Nr.	Imagine	Nume playlist	Descriere	Specificații
1		Playlist pentru relaxare	Playlistul perfect pentru nopțile răcoroase și liniștite de vară.	Gen: clasic Durată: medie
2		Playlist pentru dormit	Dacă întâmpinați dificultăți în a adormi, încercați din nou cu acest playlist.	Activitate: somn Durată: scurtă
3		Playlist pentru citit	Fie că citiți Shakespeare, Andy Weir sau E.L. James, acest playlist vă va ajuta să vă relaxați, să vă concentrați și să vă cufundați în lumea din spatele ochilor.	Activitate: citit Durată: foarte lungă

4		Playlist pentru studiat	Examenele se apropie iar tu ai lenevit tot semestrul. Permite acestui playlist să fie fundalul pentru nopțile de nesomn ce vor urma.	Activitate: studio Durată: foarte lung
5		Playlist pentru alergat	Acest playlist vă va motiva să faceți acel efort extra, să mergeți până la capăt.	Activitate: sport Durată: lungă Tip sport: alergat
6		Playlist pentru sală	Acest playlist vă va motiva să faceți acel efort extra, acea flotare în plus, să vă depășiți limitele.	Activitate: sport Durată: medie Tip sport: sală
7		Playlist pentru cântat în duș (sub 30 ani)	Să fim sinceri, toată lumea cântă în duș, chiar și tu, așa că de ce să nu lași acest playlist să te acompanieze?	Activitate: cântat Durată: scurtă Era: '00, '10

8		Playlist pentru cântat în duș (30-60 ani)	Să fim sinceri, toată lumea cântă în duș, chiar și tu, așa că de ce să nu lași acest playlist să te acompanieze?	Activitate: cântat Durată: scurtă Era: '90, '80
9		Playlist pentru cântat în duș (60+ ani)	Să fim sinceri, toată lumea cântă în duș, chiar și tu, așa că de ce să nu lași acest playlist să te acompanieze?	Activitate: cântat Durată: scurtă Era: '70
10		Playlist pentru naveta de dimineață	Îngrozitoarea navetă de dimineață, chinul existenței pentru mulți dintre noi. Nu putem să vă facem drumul mai scurt, dar cu siguranță putem să îl facem mai suportabil cu acest playlist.	Activitate: naveta Durată: lungă
11		Playlist pentru când ești furios	Dacă sunteți furios și nu intenționați să schimbați acest lucru, acest playlist este exact ce vă trebuie.	Gen: metal Durată: lungă

12		Playlist pentru zile ploioase	Dacă sunteți genul de persoană care adoră ploaia, luați o cana fierbinte de ceai și lăsați stresul și oboseala să fie spălate de ploaie și de acest playlist.	Gen: alternativ Durată: foarte lungă Vreme: ploaie
13		Playlist pentru zile însorite	Zilele pline de ploaie sunt minunate câteodată, dar nimic nu bate o după-amiază caldă și însorită la sfârșit de săptămână. Lăsați aceste melodii să vă însoțească, indiferent de cum alegeți să vă petreceți ziua.	Gen: pop Durată: foarte lungă Vreme: însorit
14		Playlist pentru atunci când ești nostalgic	Anii tăi de glorie s-ar putea să se fi terminat, taxele trebuie plătite, copiii duși la școală, dar nu este nimic rușinos în a te întoarce și a retrăi acei ani cu ajutorul acestui playlist.	Gen: clasic rock Durată: medie
15		Playlist pentru petreceri	Dacă ești la o petrecere jalnică sau dacă pur și simplu vrei să dansezi singur în camera ta, acest playlist nu numai că va înveseli atmosfera, dar te va aduce involuntar pe ringul de dans.	Gen: electronic Durată: foarte lungă

16		Playlist pentru o zi la plajă	Vara este aici, soarele strălucește orbitor pe cer, gențile tale sunt pline, rezervorul de asemenea, iar marea te așteaptă. Tot ce mai lipsește este acest playlist.	Activitate: plajă Durată: foarte lungă
17		Playlist pentru când ești la volan (anii 90)	Dacă adori să călătorești, iar șoseaua și pustietatea sunt siguri tăi însoțitori, atunci acest playlist este pentru tine.	Activitate: condus Durată: foarte lungă Era: '90
18		Playlist pentru când ești la volan (anii 80)	Dacă adori să călătorești, iar șoseaua și pustietatea sunt siguri tăi însoțitori, atunci acest playlist este pentru tine.	Activitate: condus Durată: foarte lungă Era: '80

Atribute și arbori

Arborii dezvoltati și inclusi în sistemul expert sub formă de reguli au la bază atributele incluse mai jos. Atributele pot fi directe, a căror valoare este obținută direct de la utilizator, sau deduse, a căror valoare este dedusă pe baza altor atribute(directe sau deduse la randul lor). Atributul scop reprezintă scopul sistemului expert și constituie ramura cea mai îndepărtată a arborilor.

Atribut scop: playlist

Atribute deduse:

- **gen:** clasic, metal, alternativ, pop, clasic_rock, electronic, latino - depinde de **emotie_relaxat**, **emotie_furios**, **emotie_fericit**, **emotie_melancolic**, **emotie_nostalgic**, **emotie_romantic**, **emotie_energic**

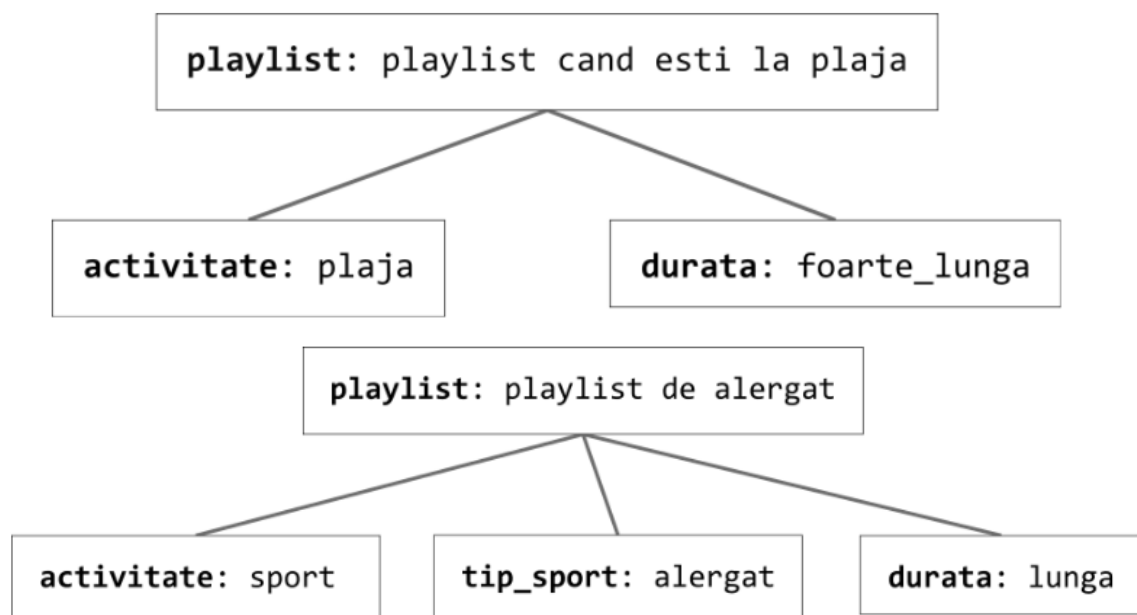
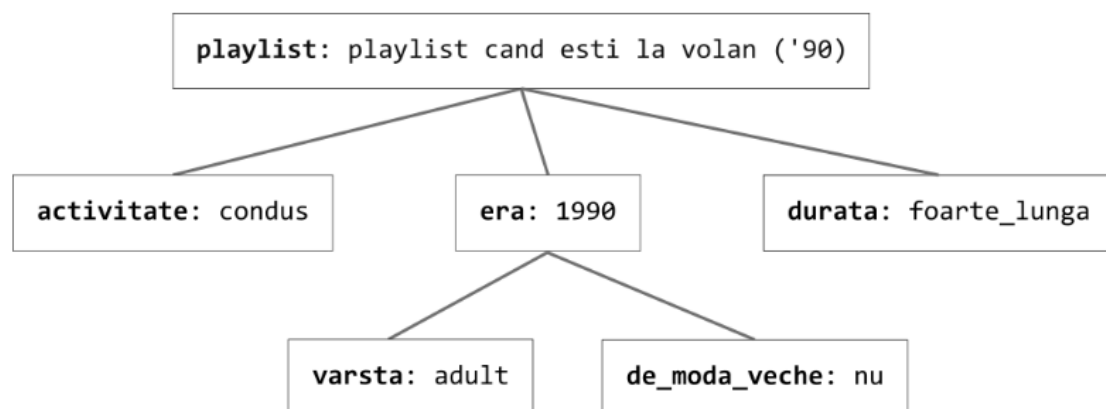
	relaxat	furios	fericit	melancolic	nostalgic	energic
clasic	40	0	10	30	20	0
metal	0	60	10	0	0	30
alternativ	25	0	0	70	5	0
pop	5	0	60	0	15	20
clasic_rock	0	5	0	15	60	20
electronic	0	0	25	0	0	75

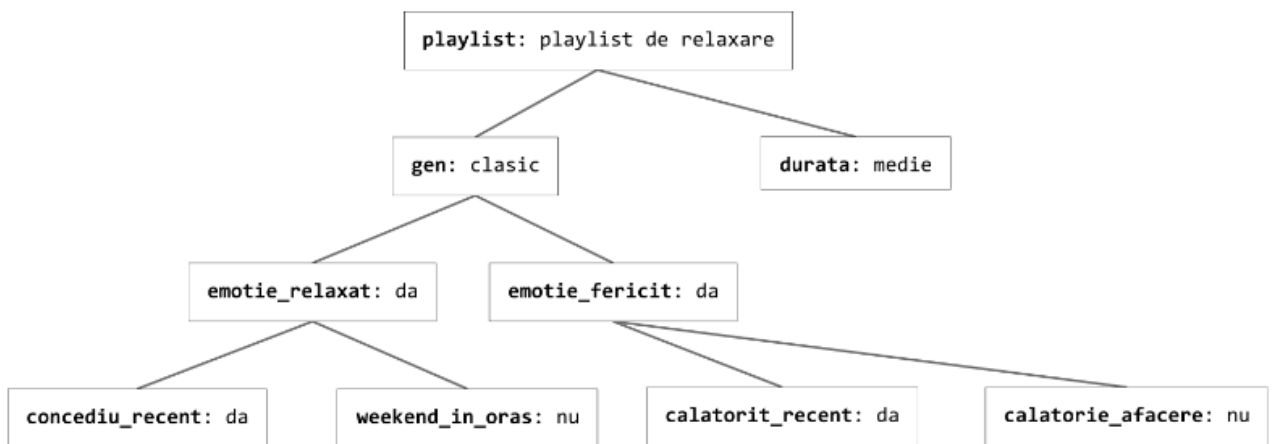
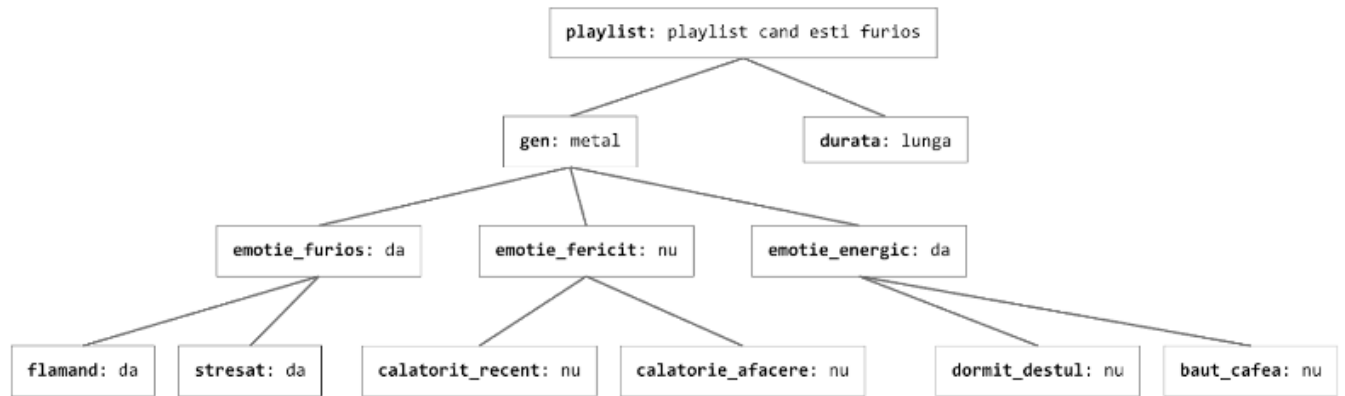
- **era :** '70, '80, '90, '00, '10 - depinde de **varsta**, **de_moda_veche**
- **emotie_relaxat:** (da/nu) - depinde de **concediu_recent**, **weekend_in_oras**
- **emotie_furios:** (da/nu) - depinde de **flamand**, **stresat**
- **emotie_fericit:** (da/nu) - depinde de **calatorit_recent**, **calatorie_afacere**
- **emotie_melancolic:** (da/nu) - depinde de **ore_suplimentare**, **apreciere**
- **emotie_nostalgic:** (da/nu) - depinde de **dor_persoana**, **dor_perioada**,
- **emotie_energic:** (da/nu) - depinde de **dormit_destul**, **baut_cafea**

Atribute a căror valoare e obținută din răspunsurile utilizatorilor:

- **durata:** scurta, medie, foarte_lunga
- **activitate:** citit, somn, studiu, sport, cantat, naveta, plaja, condus (nu conteaza)
- **tip_sport:** sala, alergat
- **concediu_recent:** (da/nu)
- **weekend_in_oras:** (da/nu)

- **flamand:** (da/nu)
- **stresat:** (da/nu)
- **calatorit_recent** (da/nu)
- **calatorie_afacere** (da/nu)
- **ore_suplimentare** (da/nu)

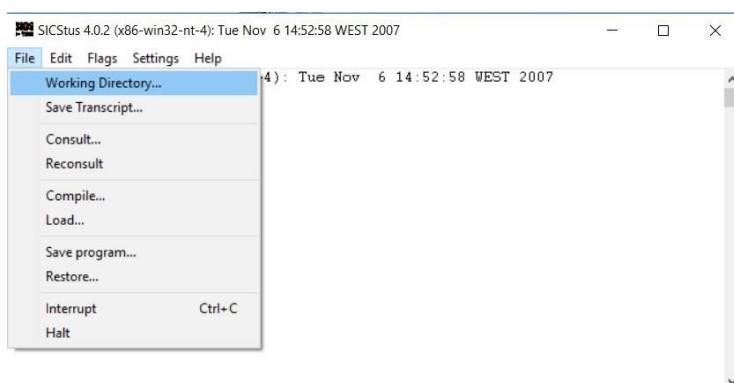




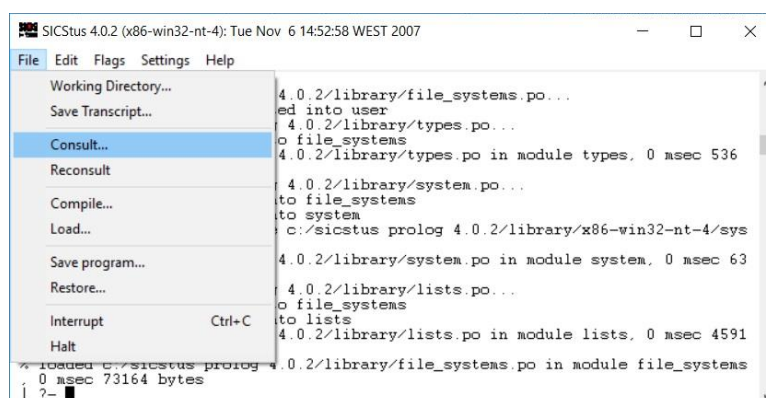
Instrucțiuni pentru consolă

Pentru a rula sistemul expert în consola SICStus sunt necesari următorii pași:

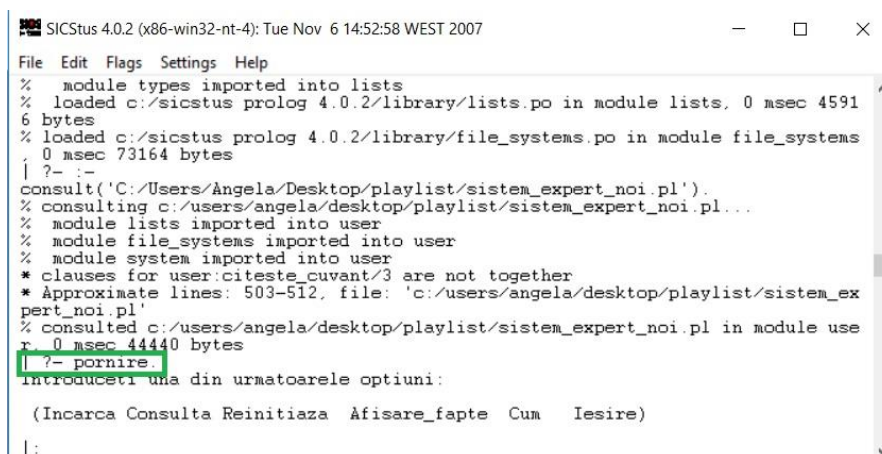
1. Setarea directorului in care se gasesc fisierul prolog si fisierul txt pentru reguli
File -> Working Directory



2. Încărcarea fișierului prolog. File -> Consult



3. Pornirea programului. Introduceți textul “pornire.”



4. Încărcarea regulilor.

Introduceți textul "incarca" + enter după care introduceți numele fișierului între ghilimele

```
SICStus 4.0.2 (x86-win32-nt-4): Tue Nov 6 14:52:58 WEST 2007
File Edit Flags Settings Help
% module system imported into user
* clauses for user:citeste_cuvant/3 are not together
* Approximate lines: 503-512, file: 'c:/users/angela/desktop/playlist/sistem_expert_noi.pl'
% consulted c:/users/angela/desktop/playlist/sistem_expert_noi.pl in module user, 0 msec 44440 bytes
| ?- pornire.
Introduceti una din urmatoarele optiuni:

(Incarca Consulta Reinitiaza Afisare_fapte Cum Iesire)
| : incarca
Introduceti numele fisierului care doriti sa fie incarcat:
| : 'reguli.txt'

Fisierul dorit a fost incarcat
Introduceti una din urmatoarele optiuni:

(Incarca Consulta Reinitiaza Afisare_fapte Cum Iesire)
| : | :
```

Pentru a consulta sistemul, introduceți comanda 'consulta' și alegeți din opțiunile oferite până când vi se oferă o soluție.

```
SICStus 4.0.2 (x86-win32-nt-4): Tue Nov 6 14:52:58 WEST 2007
File Edit Flags Settings Help
Introduceti numele fisierului care doriti sa fie incarcat:
| : 'reguli.txt'

Fisierul dorit a fost incarcat
Introduceti una din urmatoarele optiuni:

(Incarca Consulta Reinitiaza Afisare_fapte Cum Iesire)
| : | : consulta
'Ce activitate intreprindeti momentan?'
( citit somn studio sport cantat naveta plaja condus nu_conteaza )
| : sport

'Ce tip de sport practicati?'
( sala alergat )
| : sala

'Pe ce durata de timp va doriti acest playlist?'
( scurta medie foarte_lunga )
| : medie

playlist ( playlist_sala )
factorul de certitudine este 100

Introduceti una din urmatoarele optiuni:

(Incarca Consulta Reinitiaza Afisare_fapte Cum Iesire)
| : █
```

Comanda "afisare_fapte" va afisa toate faptele introduse de utilizator.



The screenshot shows a window titled "SICStus 4.0.2 (x86-win32-nt-4): Tue Nov 6 14:52:58 WEST 2007". The menu bar includes "File", "Edit", "Flags", "Settings", and "Help". The main text area displays the following text:

```
Introduceti una din urmatoarele optiuni:  
(Incarca Consulta Reinitiaza Afisare_fapte Cum Iesire)  
|: afisare_fapte  
Fapte existente in baza de cunostinte:  
(Atribut,valoare)  
(playlist,playlist_sala), certitudine 100  
(durata,medie), certitudine 100  
(tip_sport,sala), certitudine 100  
(activitate,sport), certitudine 100  
Introduceti una din urmatoarele optiuni:  
(Incarca Consulta Reinitiaza Afisare_fapte Cum Iesire)  
|:
```

Comanda "reinitiaza" șterge din memorie faptele.

Comanda "iesire" închide sistemul expert.

Comanda "cum" afișează regulile folosite pentru deducerea soluției și le scrie în fișierul "dem_[numesol#nr]_ora_min_secunde.txt".

Exemple de interogări

Mai jos sunt prezentate două exemple de rulare ale sistemului expert:

Primul este un caz cu mai multe soluții:

```
SICStus 4.0.2 (x86-win32-nt-4): Tue Nov 6 14:52:58 WEST 2007
File Edit Flags Settings Help
|: consulta
'Ati avut concediu recent?'
( da nu )
: nu

'In weekend, ati iesit cu prietenii in oras?'
( da nu )
: nu

'Ati calatorit recent in ultima perioada?'
( da nu )
: nu

'Ati calatorit in interes de afaceri in ultima perioada?'
( da nu )
: nu

'Ati lucrat ore suplimentare in ultima perioada?'
( da nu )
: nu

'Ati fost apreciat de cineva in ultima perioada?'
( da nu nu_stiu )
: nu_stiu

'Ati dormit destul?'
( da nu )
: da

'Ati baut cafea de curand?'
( da nu )
: nu
```

```
SICStus 4.0.2 (x86-win32-nt-4): Tue Nov 6 14:52:58 WEST 2007
File Edit Flags Settings Help

'Sunteti flamand?'
( da nu )
: nu

'Sunteti stresat(a)?'
( da nu )
: da

'Pe ce durata de timp va doriti acest playlist?'
( scurta medie lunga foarte_lunga nu_conteaza )
: nu_conteaza

'Va este dor de cineva?'
( da nu )
: nu

'Va este dor de o perioada?'
( da nu )
: nu

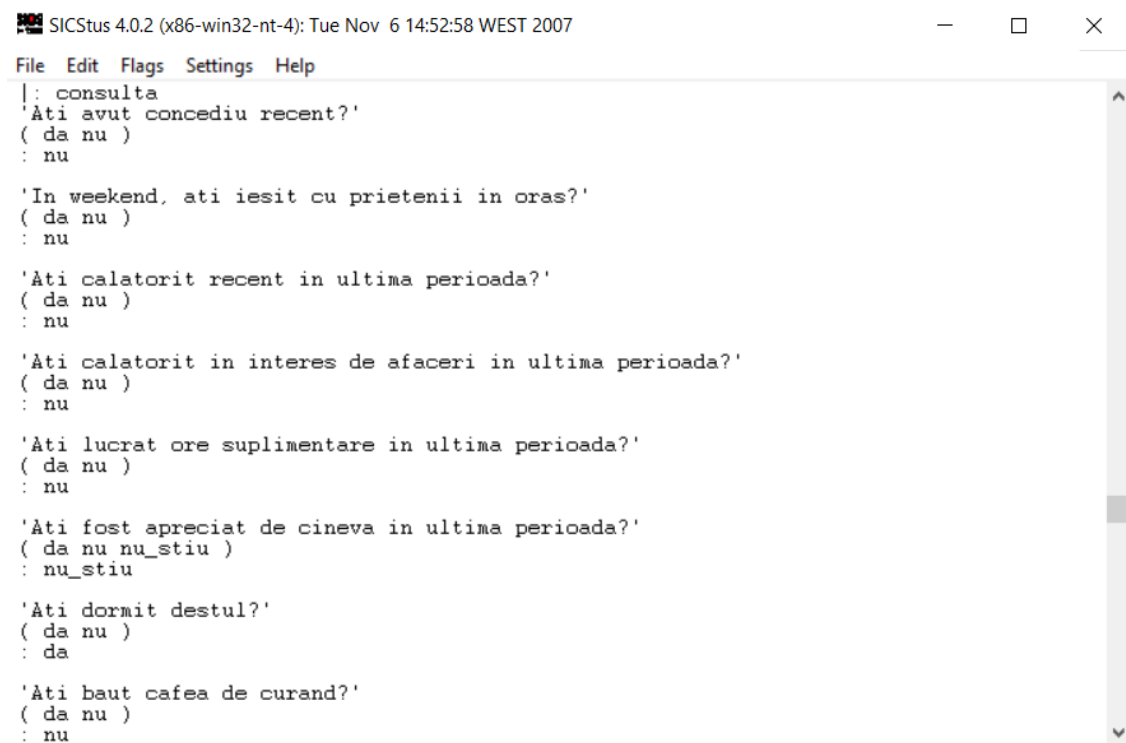
'Ce activitate intreprindeti momentan?'
( citit somn studiu sport cantat naveta plaja condus nu_conteaza nu_stiu )
: citit

playlist ( playlist_citit )
    factorul de certitudine este 100
playlist ( playlist_nervos )
    factorul de certitudine este 72
playlist ( playlist_petrecere )
    factorul de certitudine este 60
```


Răspunsuri:

activitate - citit
durată – scurtă
concediu_recent – nu
weekend_in_oras – nu
calatorit_recent – nu
calatorie_afacere – nu
ore_suplimentare – nu
apreciere – nu_stiu
dormit_destul – da
baut_cafea – nu
flamand – nu
stresat – da
dor_persoana – nu
dor_perioada – nu

Al doilea este un caz fără soluții:



```
SICStus 4.0.2 (x86-win32-nt-4): Tue Nov 6 14:52:58 WEST 2007
File Edit Flags Settings Help
|: consulta
'Ati avut concediu recent?'
( da nu )
: nu

'In weekend, ati iesit cu prietenii in oras?'
( da nu )
: nu

'Ati calatorit recent in ultima perioada?'
( da nu )
: nu

'Ati calatorit in interes de afaceri in ultima perioada?'
( da nu )
: nu

'Ati lucrat ore suplimentare in ultima perioada?'
( da nu )
: nu

'Ati fost apreciat de cineva in ultima perioada?'
( da nu nu_stiu )
: nu_stiu

'Ati dormit destul?'
( da nu )
: da

'Ati baut cafea de curand?'
( da nu )
: nu
```

```
SICStus 4.0.2 (x86-win32-nt-4): Tue Nov 6 14:52:58 WEST 2007
File Edit Flags Settings Help
'Sunteti flamand?'
( da nu )
: nu

'Sunteti stresat(a)?'
( da nu )
: nu

'Pe ce durata de timp va doriti acest playlist?'
( scurta medie lunga foarte_lunga nu_conteaza )
: scurta

'Va este dor de cineva?'
( da nu )
: nu

'Va este dor de o perioada?'
( da nu )
: nu

'Ce activitate intreprindeti momentan?'
( citit somn studiu sport cantat naveta plaja condus nu_conteaza nu_stiu )
: nu_stiu

Nu s-au putut gasi solutii

Introduceti una din urmatoarele optiuni:
(Incarca Consulta Reinitiaza Afisare_fapte Cum Iesire)
|: █
```

Răspunsuri:

activitate – nu_stiu
durată – scurtă
concediu_recent – nu
weekend_in_oras – nu
calatorit_recent – nu
calatorie_afacere – nu
ore_suplimentare – nu
apreciere – nu_stiu
dormit_destul – da
baut_cafea – nu
flamand – nu
stresat – nu
dor_persoana – nu
dor_perioada – nu

Explicarea cerințelor

Cerințe generale:

1. Să se afișeze opțiunile da/nu în cazul întrebărilor cu răspuns boolean.

Pentru a afișa opțiunile da/nu am modificat predicatul interogheaza:

```
interogheaza(Atr,Mesaj,Optiuni,Istorie) :-  
    write(Mesaj),nl,  
    citeste_opt(X,Optiuni,Istorie),  
    det_val_fc(X,Val,FC),  
    asserta( fapt(av(Atr,Val),FC,[utiliz]) ).
```

2. Să se afișeze opțiunile nu_stiu/nu conteaza în cazul întrebărilor cu răspuns boolean.

Pentru a afișa opțiunile nu_stiu/nu conteaza am modificat predicatul citeste_opt:

```
citeste_opt(X,Optiuni,Istorie) :-  
    append([''],Optiuni,Opt1),  
    append(Opt1,[''],Opt),  
    scrie_lista(Opt),  
    de_la_utiliz(X,Istorie,Optiuni).
```

3. Implementare nu_conteaza

Pentru implementarea nu_conteaza am modificat predicatul realizare_scop astfel:

```
realizare_scop(not Scop,Not_FC,Istorie) :-  
    realizare_scop(Scop,FC,Istorie),  
    Not_FC is - FC, !.
```

```
realizare_scop(Scop,FC,_) :-  
    fapt(Scop,FC,_), !.
```

```
realizare_scop(av(Atr,_),FC,_) :-  
    fapt(av(Atr, nu_conteaza),FC,_), !.
```

4. Afișarea soluțiilor în ordine descrescătoare

Pentru afișarea soluțiilor în ordine descrescătoare a factorului de certitudine am implementat un setof pentru a le ordona, în predicatul afiseaza_scopuri:

```
afiseaza_scopuri :-  
    nl,  
    (setof(st(FC,av(Atr,Val)),F^B^(scop(Atr),fapt(av(Atr,Val),F,B),F >= 20,FC is -F), L) ->  
    (L = [],  
    write('Nu s-au putut gasi solutii');  
    L \= [],  
    (member(st(FC,av(Atr,Val)), L),  
    F is -FC,  
    scrie_scop(av(Atr, Val), F), fail; true)  
    );  
    write('Nu s-au putut gasi solutii')), nl, nl, !.
```

5. Afișare mesaj “nu am soluții”

Pentru a avea mesajul “nu am soluții” am modificat predicatul afiseaza_scopuri, astfel:

```
afiseaza_scopuri :-  
    nl,  
    (setof(st(FC,av(Atr,Val)),F^B^(scop(Atr),fapt(av(Atr,Val),F,B),F >= 20,FC is -F), L) ->  
    (L = [],  
    write('Nu s-au putut gasi solutii');  
    L \= [],  
    (member(st(FC,av(Atr,Val)), L),  
    F is -FC,  
    scrie_scop(av(Atr, Val), F), fail; true)  
    );  
    write('Nu s-au putut gasi solutii')), nl, nl, !.
```

Cerințe individuale:

1. Parsarea

Regulile vor avea formatul :

```
rg//id (unde id este numărul regulii)
atribut_concluzie(valoare) cu fact_cert//nr (concluzia; nr este factorul de certitudine)
daca -> (condițiile; premisele încep cu un +)
    + atr =? valoare (pentru attribute cu valori multiple)
    + atr (pentru attribute booleene, valoare true)
    + nu(atr) (pentru attribute booleene, valoare false)
.
```

Concluzii booleene:

```
atribut_concluzie(T) cu fact_cert//nr(true)
atribut_concluzie(F) cu fact_cert//nr(false)
```

Sa se permita ca premisele sa poata incepe si cu # in loc de +.

Întrebarile vor avea formatul:

```
intreb//atribut
text('continut intrebare')
optiuni ->
? val1
? val2
? val3
.
```

Scopul se va defini:

```
scop//atr.
```

dar să se poată defini și sub forma:

```
scopul ? atr.
```

Pentru parsare am modificat trad, lista_optiuni, lista_de_optiuni, propoz și propozPremise, astfel:

```
trad(R,L,[],assertz(R), !.
```

```
trad(scop(X)) --> [scop,/,/,X].
```

```
trad(scop(X)) --> [scop,?,X].
```

```
trad(interogabil(Atr,M,P)) --> [intreb,/,/,Atr],afiseaza(P),lista_optiuni(M).
```

```

trad(regula(N,premise(Daca),concluzie(Atunci,F))) -->
identificator(N),atunci(Atunci,F),daca(Daca).

trad('Eroare la parsare'-L,L,_).

lista_optiuni(M) --> [optiuni,'-', '>'],lista_de_optiuni(M).

lista_de_optiuni([]) --> [].

lista_de_optiuni([Element|T]) --> ['?', Element],lista_de_optiuni(T).

identificator(N) --> [rg,/,/,N].

atunci(Atunci,FC) --> propoz(Atunci),[cu, fact_cert,/,/,FC].

propoz(not av(Atr,da)) --> [Atr,'(,f,')'].

propoz(av(Atr,da)) --> [Atr,'(,t,')'].

propoz(av(Atr,Val)) --> [Atr,'(,Val,')'].

daca(Daca) --> [daca, '-', '>'],lista_premise(Daca).

lista_premise([]) --> [].

lista_premise([Prima|Celalalte]) --> propozPremise(Prima),lista_premise(Celalalte).

propozPremise(av(Atr, Val)) --> ['+', Atr, '=', '?', Val].

propozPremise(not av(Atr, da)) --> ['+', nu, '(', Atr, ')'].

propozPremise(av(Atr, da)) --> ['+', Atr].

```

2. În afişarea demonstraţiei regulile se vor afisa ca în fişierul de intrare.

Pentru asta am modificat afis_regula:

```

afis_regula(N) :-
    regula(N, premise(Lista_premise),
    concluzie(Scop,FC)),NN is integer(N),
    scrie_lista(['rg /',NN]),
    transformare(Scop,Scop_tr),
    append([' ',Scop_tr,L1),
    FC1 is integer(FC),append(L1,[cu,fact_cert,'/',FC1],LL),
    scrie_lista(LL),
    scrie_lista([' daca ->']),

```

```
scrie_lista_premise(Lista_premise),
nl.
```

4. **Se va crea prin program un folder numit output_sys_exp (daca exista, il va folosi pe acela, iardaca nu exista, il va crea).**

In interiorul folderului de output, programul va crea un director numit *demonstratii_raspunsuri_sistem* (daca exista, il va folosi pe acela, iar daca nu exista, il va crea), in care va exista un fisier numit *dem_[numesol#nr]_ora_min_secunde.txt* unde *ora, min, secunde*, sunt corespunzatoare momentului in care a fost obtinuta solutia. Daca ora, minutul sau secunda sunt mai mici decat 10 se vor afisa cu 0 in fata (sa aiba mereu 2 cifre). Orele sunt de la 0 la 23. Cuvantul *numesol* va fi inlocuit cu valoarea solutiei, iar *nr* cu factorul de certitudine. Prima linie a fisierelor de acest gen va fi de forma "Se demonstreaza *raspuns* avand factorul de certitudine: *nr*", unde *raspuns* e raspunsul pentru care este afisata demonstratia, iar *nr* e factorul de certitudine. A doua linie va contine descrierea solutiei. In demonstratii, afisarea regulilor se va face exact in forma in care au fost scrise in fisierul de intrare.

In cazul in care atributul explicat in demonstratie are valoarea obtinuta de la utilizator, se va afisa:

Atributul *atr* a fost obtinut de la utilizator cu valoarea *val* si factorul de certitudine *nr*.

Pentru a scrie demonstrația am modificat codul nostrum astfel:

pregateste_output(Scop):-

```
Nume_Dir = 'demonstratii_raspunsuri_sistem',
(\+directory_exists(Nume_Dir),
make_directory(Nume_Dir); true),
nume_fisier(Scop, Nume_Fis),
transformaNume([Nume_Dir, '/', Nume_Fis], Nume),
write('Demonstratia a fost scrisa in '), write(Nume), nl, nl,
tell(Nume),
scrie_intro(Scop).
```

reseteaza_output:-

```
told.
```

nume_fisier(av(Scop, Val), Nume) :-

```
fapt(av(Scop, Val), F, _), number_chars(F, F2), atom_chars(FC, F2),
timp(Ora, Minut, Secunda),
Nume_Fis_L = ['dem_', Val, '#', FC, '_', Ora, '_', Minut, '_', Secunda, '.txt'],
transformaNume(Nume_Fis_L, Nume).
```

transformaNume([H], Atom) :-

```
atom_concat(H, "", Atom).
```

transformaNume([H | T], Atom) :-

```
transformaNume(T, Atom2),
atom_concat(H, Atom2, Atom).
```



```
timp(Ora, Minut, Secunda):-  
    datetime(datetime(_,_,_, O, M, S)),  
    format_timp(O, Ora),  
    format_timp(M, Minut),  
    format_timp(S, Secunda).
```

```
format_timp(T, TF) :-  
    number_chars(T, TFL),  
    (T < 10, atom_chars(TF, ['0' | TFL]));  
    T >= 10, atom_chars(TF, TFL)).
```

```
scrie_intro(Scop) :-  
    fapt(Scop, FC, _),  
    transformare(Scop, PG),  
    append(['Se', demonstreaza], PG, Aux1),  
    append(Aux1, [avand, factorul, de, certitudine, ':', FC], Aux2),  
    scrie_lista(Aux2),  
  
    nl.
```

Concluzie

Oricum le-am numi, sistemele cognitive sau sistemele inteligente, Sistemele Expert constituie o subramură a tehnologiei Inteligenței Artificiale și s-au remarcat deja în numeroase aplicații (medicină, chimie, electrotehnică, agricultură etc.). Sistemele Expert, generatoarele de Sisteme Expert și chiar generatoarele de sisteme neuronale sunt deja disponibile pe piață și contribuie din plin la modelarea întreprinderii viitorului.