Logicus – Soft Educațional 2.0

Budişteanu Ionuț – elev în clasa a X-a la C.N. „Mircea cel Batran”,
ibudisteanu@acm.org
Mlisan Mirela – profesor îndrumător – C.N. „Mircea cel Batran”,
mirela_mlisan@yahoo.com

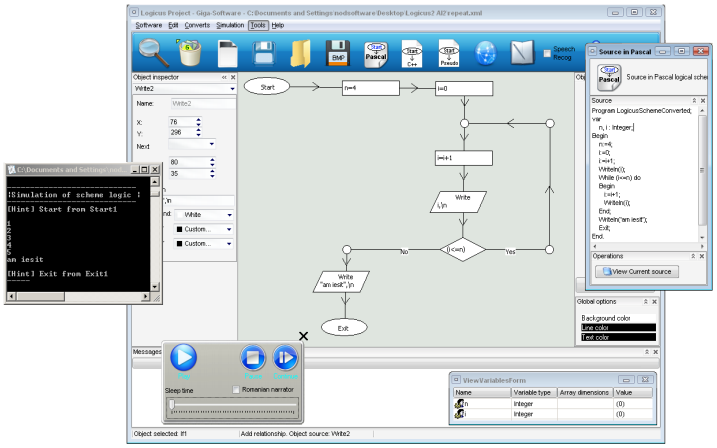
Abstract

Aplicația prezintă un nou concept - software educational 2.0, softuri cu Inteligență Artificială şi nu se bazează pe conținutul informațional (lecții). Elevul poate să creeze orice schemă logică în aplicație cu drag&drop, după care soft-ul poate genera dinamic codul sursă a schemei logice create de elev, astfel se pot creea o infinitate de scheme logice. Soft-ul are inclusă o interfață vocală (Vocal User Interface), ce permite recunoaşterea a anumitor cuvinte vorbite la microfon, soft-ul generează răspunsuri audio pentru elev. S-au realizat diverse teste, unde elevul este obligat să răspundă desenând răspunsul, iar soft-ul reuşeste să recunoască scrisul de mână şi verifică dacă este corect. Permite recunoaşterea facială, pentru a se autentifica la catalog.

1. Introducere

Permite crearea de scheme logice, simularea, si să genereze coduri în limbajele: (Pascal, C/C++ şi Pseudocod) a schemei logice, construite de utilizator.

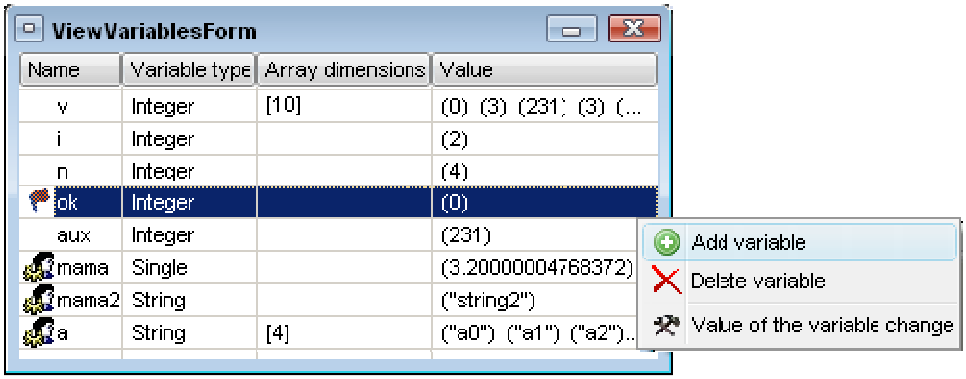
Soft educațional 2.0 este un software, care conține paradigme ale inteligenței artificiale. El permite studiul anumitor noțiuni, a obiectelor create de elev(utilizator). Practic nu se bazează pe conținut, permite interactivitate maximă, în soft, putând fi create un număr nelimitat de exemple, neefiind limitat la câteva animații.



2. Tabela de variabile

Adăugarea de variabile se face în fereastra: Watch – PopMenu -> Adăugă variabila. Momentan softul permite 8 tipuri de date, din fiecare tip de date se pot face matrici unidimensionale, bidimensionale, tridimensionale, acestea fiind: *Integer, SmallInt,*

ShortInt, Int64, Byte ,Word ,LongWord ,Real,String.

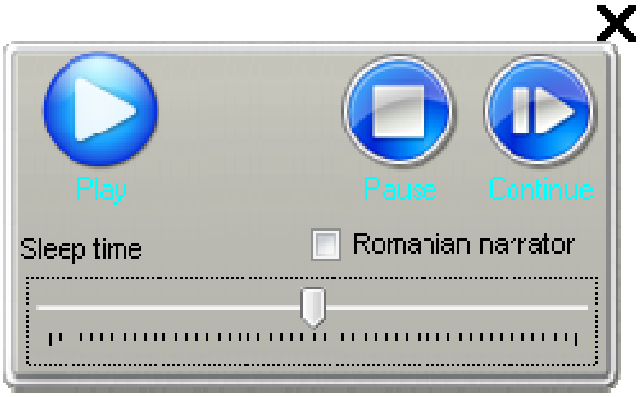


Când se introduce o nouă variabilă, se verifică, dacă mai există, dacă este validă ca şi sintaxă, ca şi dimensiuni.

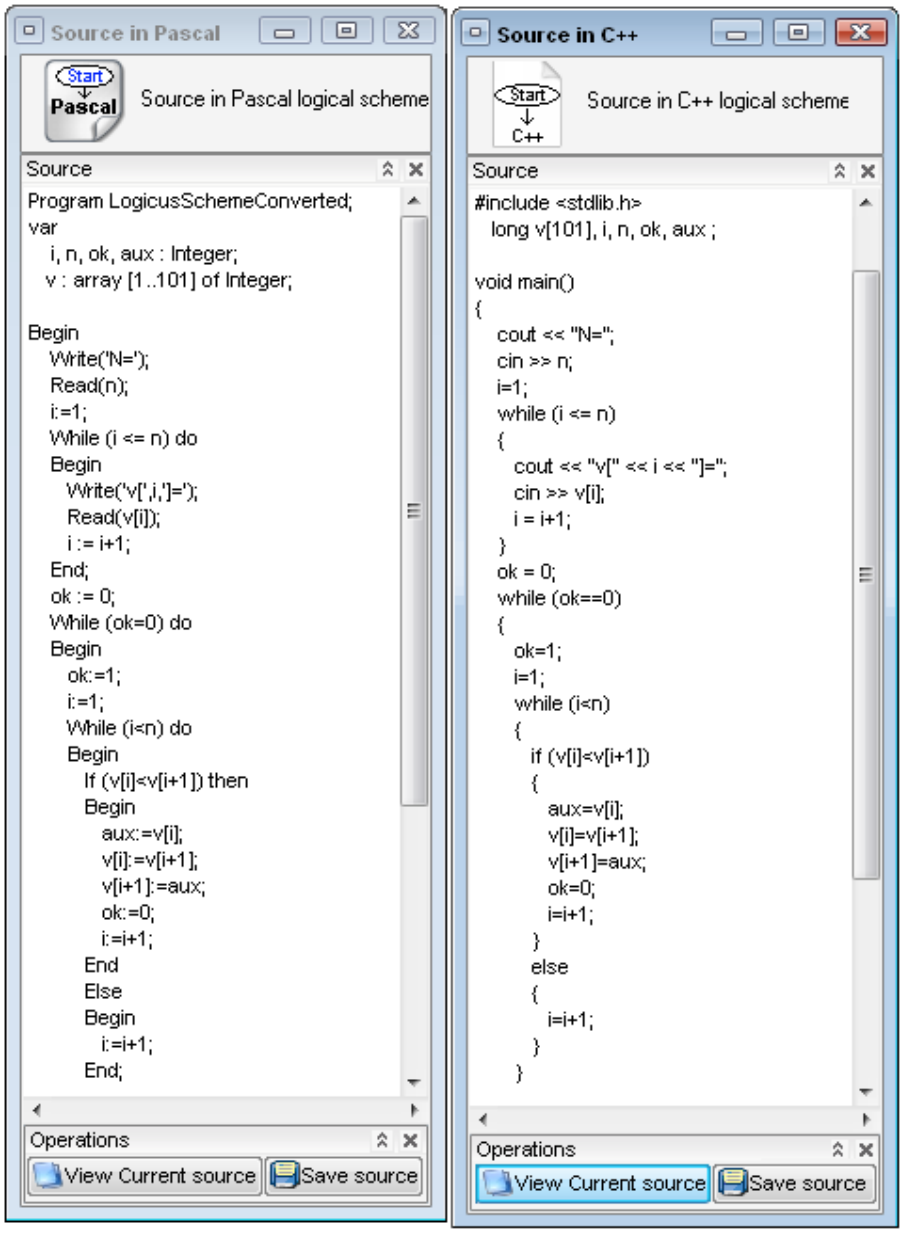
În imagine softul-ul ne arată, că variabila nu se poate introduce, că mai există, iar dimensiunile nu sunt bune alese (parantezele nu sunt închise corect).

3. Simularea schemei logice

Player-ul aplicației – prezintă simularea schemei logice, creată de utilizator. Simulatorul, este un thread sincronizat. În simulator, se poate configura, timpul, de execuție, dintre două instrucțiuni. În timpul execuție se pot seta din watch valorile variabilelor.



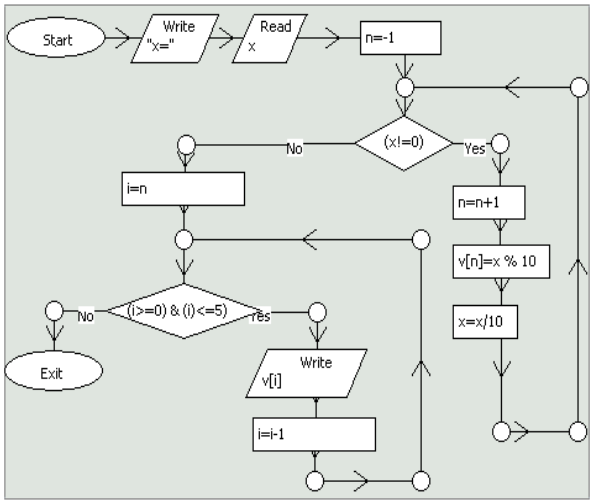
Exportarea schemei logice, în diverse limbaje de programare. Deci softul generează codul sursă- nu are nici o sursă salvată. Exportarea se face in C++,Pascal şi Pseudocod.



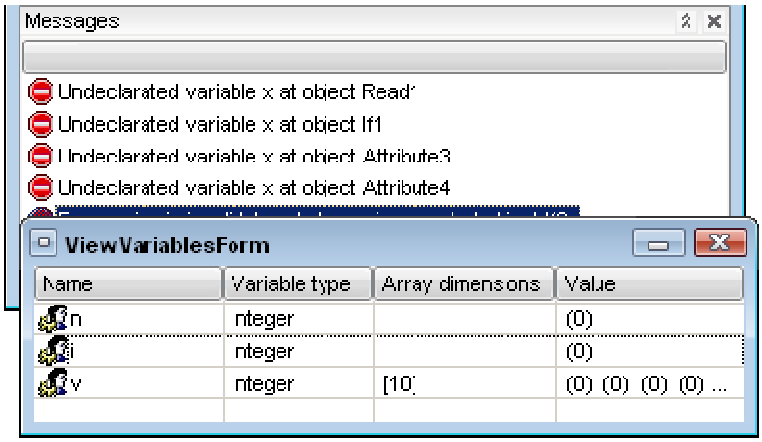
4. Analiza sintaxei şi analiză lexicală.

Soft-ul are un *thread* sincronizat, care tot verifică concurent analiza sintaxei şi analiza lexicală a schemei logice create de elev.

Exemplu - Având următoarea schemă logică



Automat, softul detectează următoarele erori în schema creată de utilizator:



5. Inteligența artificială

Softul conține un singur paradigm al inteligenței artificiale, şi anume perceptronul. S-au creat rețele neuronale Multi-Layer-Perceptron în NeurosLab, au fost antrenate tot în NeurosLab (aplicație scrisă de mine şi prezentată la foarte multe competiții), şi după aceea exportate. S-au scris clasele care permit sa citească ponderile rețelei antrenate şi să le simuleze. Rețelele neuronale au fost folosite la:

- Speech recognition, care permite recunoaşterea câtorva comenzi precum Search, Delete, Save, Load, New, Stop – cuvinte vorbite la microfon.
- Optical Character Recognition, care permite ca elevul să deseneze numere (rezultatele), iar softul permite recunoaşterea lor.
- Face recognition, ce permite recunoaşterea fețelor umane şi clasificarea lor.

6. Vocal User interface

VUI este un nou concept de interfața, ce permite comenzi vocale, iar interfața generează răspunsuri audio pentru utilizator. Deasemenea se observă migrarea de la Graphic User Interface către Vocal User Interface, în foarte multe aplicații comerciale. Pentru generarea audio a cuvintelor, s-a folosit o aplicație scrisă de mine, DictRO care poate nara orice text in limba română. Din păcate interfața nu reuşeste interpretarea limbajului uman, ci doar anumite cuvinte, deasemenea softul permite recunoaşterea facială, urmând ca feţele să fie clasificate, pentru a crea o logare pe bază de feţe, prin aceasta s-a dorit sa se arate că un domeniu atât de închis, softul educațional, poate cuprinde elemente de noutate, arătând o nouă direcție de dezvoltare a softurilor educaționale.

7. Speech recognition

Întrucât ANN nu ştiu să facă decât clasificări, trebuie să transformăm, problema într-o problemă de clasificare. Sunetul trebuie transformat într-o poză, pe care rețeaua neuronală să încerce să o clasifice. Crearea unei poze în domeniul amplitudine-timp nu este utilă. Trebuie lucrat în domeniul timp-frecvență.

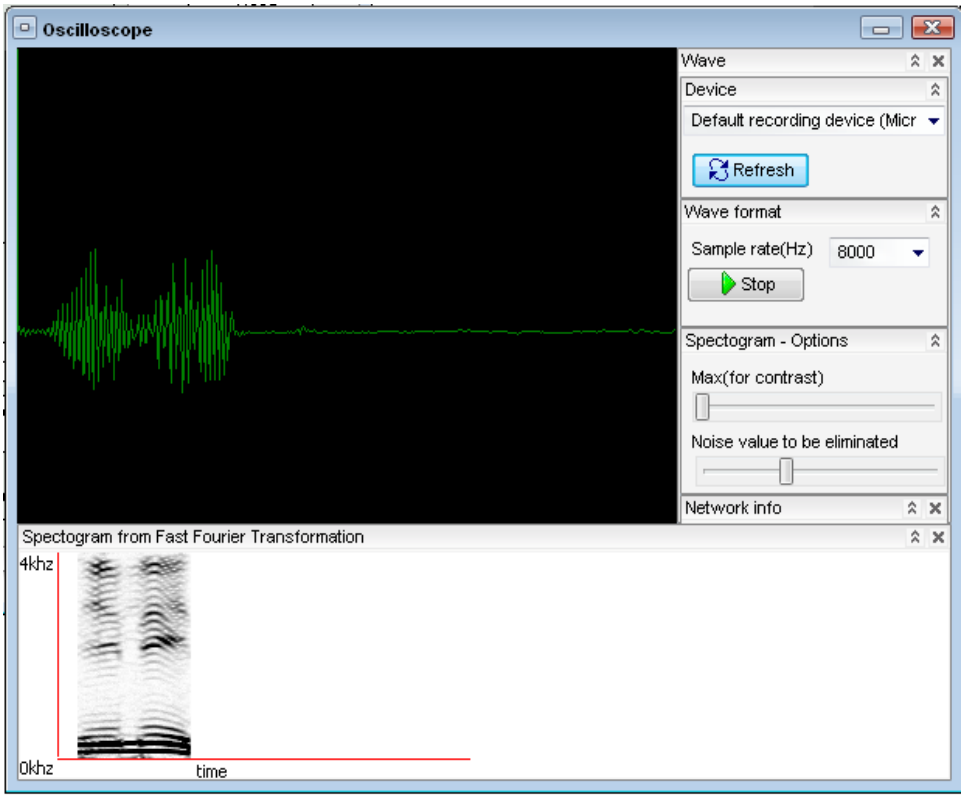
După aceea, s-a dorit să se vadă dacă o rețea neuronală MLP poate să clasifice sunete. Astfel s-a creat un osciloscop, în domeniul timp amplitudine. S-a lucrat cu Microsoft Multimedia System (mmsystem) luând astfel eşantioanele de la microfon. Eşantioanele sunt pe 8 biți, s-a setat frecvența de eşantionare pe 8000hz. După aceea s-a făcut o clasă care să ploteze eşantioane. Când s-au desenat eşantioanele, nu s-a scos componenta continuă (DC). După aceea, s-a scăzut DC(componenta continuă) şi inmulțit cu fereastra Hamming pentru a îmbunătăți calitatea sunetului preluat de la microfon.

Fereastra hamming este: hamming(x)=0,54-0,46*cos(2*pi*(x/256)) Această funcție este clasică. Funcția este din MatLab, face corecția de frecventă.

Algoritmul cu fereastra hamming: Se iau primele 256 [1..256] de eşantioane se introduc într-un buffer şi se inmulțeşte cu valorile hamming. După ce se calculează prima fereastră se introduce la analiza Transformata Rapida Fourier, din rezultat se face modulul numerelor complexe dupa transformată şi se obține câte o coloană, din imaginea gray-scale, a spectrogramei sunetului. Deci se va obține 8000(numărul de eşantioane, luate într-o secundă)/30, unde 30 este deplasamentul ferestrei hamming. După aceea, se reia procedeul, pe intervalul [30,286] şi se aplică aceeasi înmulțire. După aceea [60,316], etc...

Valorile care se obțin, în spectograma, se normează, şi se înmulțesc cu 255, pentru a le transforma în valori de la [0..255]. Se aplică diverse metode pentru a elimina zgomotul prin mediere, componenta continuă(DC) şi a separa cuvintele de zonele de pauză. Se aplică diverse nivele de separarea zgomotului şi a intervalului de pauză. Pentru realizarea FFT-ului s-a utilizat singurul unit folosit din toată aplicația, Don Cross unit, care face doar transformata Fourier, procesarile sunt realizate manual.

După ce avem spectrograma, se separă în cuvinte. Fiecare spectrogramă a fiecărui cuvânt se scalează la o dimensiune standard(deoarece pe stratul de intrare trebuie să fie fix 1218 intrări) la 42*29 pixeli, o dimensiune atât de redusă, deoarece pentru algoritm informația este redundantă. Atât prima image, cât şi a doua sunt aproape la fel, din punct de vedere informațional.



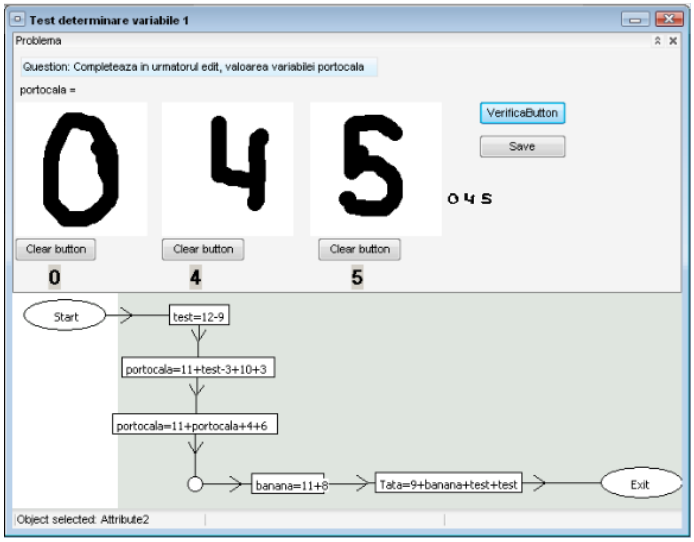
8. Partea educațională

Softul generează un tip de test educațional în care se generează o schemă logică random, iar elevul trebuie să o reaşeze şi este obligat să răspundă după execuția codului ce valoare are o variabilă. Elevul completează scriind de mână, rezultatul, softul foloseşte un OCR(scris manual) pentru detecția cifrelor, iar rezultatul este vorbit.

Optical Character recognition are 100 de intrări, 10 ieşiri (deoarece sunt 10 clase, numerele de la 0..9). Elevul va desena răspunsul în suprafața de desen(de dimensiuni de 128*128 pixeli). După ce desenează, se face Regional of Interes(ROI) şi va "decupa" doar cifra, se va scala către o imagine mai mică de (10*10 pixeli), deoarece pentru algoritm informația este redundantă. Atât prima image cât şi a doua sunt aproape la fel informațional.

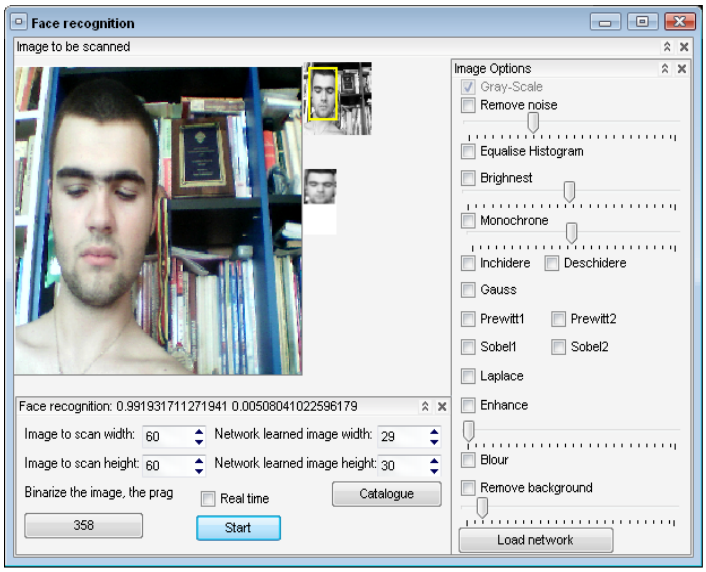
La antrenare rețeaua a fost invățată cu 5 exemple pentru fiecare clasă.

Se pot creea teste manual. Profesorul poate crea teste răspunsuri cu coloană şi lecții cu fişier (fişiere suportate de browser IE deci flv, html, pdf, doc). Profesorul scrie un fişier XML, care softul îl parsează, şi creează lecția. Deasemenea, se pot descărca lecții cu update-ul aplicației. Logicus reprezintă o viziune nouă, a softurilor educaționale, prin utilizarea noilor tehnologii.



9. Face recognition

Softul permite logarea in zona de administrare dupa ce are loc o recunoaştere facială + o clasificare facială. Atât partea de recunoaştere facială, cât şi partea de clasificare faciala este perfect functională, însă recunoaşterea real-time mai are probleme. Totuşi scopul dorit a fost atins, implementarea acestor concepte în softurile educaționale. Menționez ca partea de recunoaştere faciala cât şi partea de clasificare a feţelor, este complet scrisă manual, şi prezentată in diverse concursuri. Nu s-a folosit nici un SDK. Alți programatori pot utiliza SDK-uri gata scrise. Ex: OpenCV.



10. Asemănarea dintre Logiucus şi un Compilator

Logicus foloseste foarte multe concepte dintr-un compilator şi teoria compilării:

- Face analiză sintactică – caută cuvintele care nu aparţin vocabularului
- Crează tabele de variabile
- Face analiză lexicală variabile nedeclarate verifică corectitudinea unei expresii
- Fiecare instrucţiune poate fi considerată un token
- La fel ca şi compilatoarele Logicus crează translatarea schemei in cod sursă. Compilatoarele fac mai multe translatări din codul sursă în variante intermediare până ajunge in limbaj de asamblare. Translatările sunt echivalente.
- Crează graf asociat echivalent schemei logice. Compilatorul creaza graful asociat codului sursă.

11. Informaţii suplimentare

Film: http://www.trilulilu.ro/gigasoftware/f1cbc7574b9446
Documenaţie: http://www.neuroslab.com/Projects/Logicus/files/romana.doc

Bibliografia

[1] ACM Digital Library
[2] Virgil Tiponut, Catalin Caleanu, Reţele neuronale –arhitecturi şi algoritmi, Univ de Vest
[3] Virgil Tiponut, Catalin Daniel Caleanu, Reţele neuronale Aplicaţii, Universitatea de Vest
[4] Nicolae Ţăndăreanu , Reţele neuronale - Nicolae Ţăndăreanu
[5] Răzvan Andonie, Inteligenţă artificială, Universitatea Babeş-Bolyai Cluj-Napoca

NOTA editorului: *Deşi în perioada 1960-1980 s-au utilizat schemele logice în învăţarea algoritmicii şi programării, astăzi acestea nu trebuie să mai fie utilizate. De aceea s-a inventat programarea structurată (structurile de control)*. Conf. Dr. M. Vlada.