

## GENERAREA ȘI REZOLVAREA TESTELOR DE EVALUARE LA DISCIPLINA "SISTEME SUPORT PENTRU DECIZII"

*M. Beldiga (Vasilache), doctorandă  
Universitatea de Stat din Moldova*

### INTRODUCERE

Dezvoltarea rapidă a tehnologiilor informaționale au adus schimbări în metodele de predare și evaluare în cadrul instituțiilor de învățământ.

În ultimii ani pe piața autohtonă și pe cea internațională au apărut noi metodologii de e-Learning cum ar fi: Cd-uri multimedia de instruire, evaluarea pentru diferite discipline școlare, universitare și diverse platforme educaționale. În instituțiile de învățământ a început să fie practică tot mai des instruirea asistată de calculator (IAC), evaluarea asistată de calculator (EAC), instruirea și evaluarea la distanță.

În urma unui studiu de caz a mai multor platforme educaționale s-a ajuns la concluzia că, cele mai reușite platforme educaționale implementate în instituțiile de învățământ din Republica Moldova sunt: MOODLE, ILIAS, Claroline (colegii, universități), AeL (școli, licee, colegii) [1].

În Universitatea de Stat din Moldova (USM) este implementată platforma educațională MOODLE (*Modular Object-Oriented Dynamic Learning Enviroment*). Platforma reprezintă un pachet Software destinat pentru crearea și organizarea de cursuri online, precum și coordonarea online a activităților studenților.

### 1. SUPORTUL DIDACTIC LA DISCIPLINA „SISTEME SUPORT PENTRU DECIZII”

Disciplina "Sisteme Suport pentru Decizii" (SSD) reprezintă un curs online realizat în cadrul platformei MOODLE, inclusă în planurile de studii la USM, specialitățile de Informatică. Evaluarea studenților pe platforma MOODLE a ușurat un pic lucrul profesorului din motiv că verificarea testelor de evaluare formativă, sumativă și notarea studenților se face automat. Dar a apărut o altă problemă legată de interacțiunea actorilor procesului educațional și MOODLE – elaborarea unui număr considerabil de itemi pentru formarea

testelor de evaluare formativă, sumativă, specificărilor lucrărilor de laborator, precum și verificarea acestor lucrări necesită efort și timp considerabil din partea titularului disciplinei. Din acest motiv autoarea prezentului articol și-a propus elaborarea unui Sistem Suport Inteligent (SSI) orientat pe generarea și rezolvarea problemelor decizionale conținute în cursul "Sisteme Suport pentru Decizii".

Un test de evaluare sumativă la disciplina "SSD" este format din patru itemi, fiecare item reprezintă o familie de probleme decizionale (FPD), care, la rândul lor, se clasifică în submulțimi de probleme decizionale. Mulțimea problemelor decizionale (PD) inclusă în cadrul cursului "SSD" constă din patru submulțimi:

$$PD = D_1 \cup D_2 \cup D_3 \cup D_4,$$

unde:

- $D_1$  - familia de probleme decizionale monocriteriale (abreviat  $FPD_{M_0}$ );
- $D_2$  - familia de probleme decizionale multiatribut (abreviat  $FPD_{M_u}$ );
- $D_3$  - familia de probleme decizionale modelate cu arbori de decizie monoperoadă (abreviat  $FPD_A$ );
- $D_4$  - familia de probleme decizionale multiatribut modelate cu ajutorul mulțimilor fuzzy (abreviat  $FPD_F$ ).
- $D_i \cap D_j = \emptyset, 1 \leq i \leq 4, 1 \leq j \leq 4, i \neq j.$

Mulțimea PD reprezintă reuniunea submulțimilor  $FPD_{M_0}$ ,  $FPD_{M_u}$ ,  $FPD_A$ ,  $FPD_F$ , fiecare dintre submulțimi sunt disjuncte două câte două.

Fiecare problemă decizională din cadrul cursului „Sisteme Suport pentru Decizii” reprezintă o specificare a unei lucrări de laborator și, concomitent, un element a unei oarecare familii decizionale. De aceea, fiecare familie decizională poate fi privită ca o teorie formală (axiomatizată).

**Definiția 1** (adaptată după E. Mendelson [2]). *Teoria formală (axiomatizată)  $\mathfrak{T}$  este considerată definită, dacă se respectă următoarele condiții:*

- (1) Este dată o oarecare mulțime finită de simboluri – alfabetul teoriei  $\mathfrak{T}$ . Șirurile finite

de simboluri ale acestui alfabet sunt numite **expresii ale teoriei**  $\mathfrak{T}$ .

- (2) Există o submulțime a expresiilor teoriei  $\mathfrak{T}$ , numită **mulțimea formulelor** a teoriei  $\mathfrak{T}$ .
- (3) În mulțimea formulelor este conturată o submulțime, numită **mulțimea axiomelor** a teoriei  $\mathfrak{T}$ .
- (4) Există o mulțime finită  $R_1, \dots, R_n$  de relații dintre formule, numită **mulțimea regulilor de inferență**. Pentru fiecare regulă  $R_i$  există un număr natural  $j$  așa că pentru fiecare mulțime din  $j$  formule și pentru fiecare formulă  $A$  efectiv se determină dacă aceste  $j$  formule și formula  $A$  se află în relația  $R_i$  - iar dacă da, atunci  $A$  este numită **consecință logică** a acestor  $j$  formule prin relația  $R_i$ .

## 2. GENERATOARE DE PROBLEME DECIZIONALE

Generarea testelor și a formulărilor problemelor de laborator personalizate (in continuare – probleme personalizate) pentru studenții înscriși la curs reprezintă activități care solicită din partea profesorului eforturi și volum de timp considerabil. De aceea, cercetările legate de generare automată a problemelor personalizate la disciplinele de informatică și evaluarea

automatizată a răspunsurilor studenților la aceste lucrări sunt foarte actuale.

Fiecare temă a disciplinei „SSD” reprezintă o *teorie formală* în care, alfabetul, mulțimea formulelor și mulțimea regulilor de inferență constituie *limbajul teoriei formale* (a vedea, de exemplu, [2]). Acest limbaj este un *limbaj profesional* - submulțime structurată a unui limbaj natural.

După construirea *modelului formal* a fiecărei teme se trece la elaborarea a două *componente inteligente* de e-Learning pentru tema corespunzătoare: *generatorul de probleme* și *rezolvitorul de probleme*.

Generatorul de probleme reprezintă un produs software inteligent care, utilizând factorul aleator, la fiecare accesare automat elaborează o problemă personalizată. Astfel, pentru fiecare temă pot fi obținute câte o *familie extensibilă de probleme decizionale*.

Generarea unei probleme decizionale cu ajutorul SSI parcurge următoarele etape.

1. Utilizatorul final introduce datele de intrare în SSI, sau SSI elaborează automat datele de intrare.
2. SSI primește datele de intrare din XML.
3. SSI prelucrează, analizează, generează și transmite pachetului Mathematica informația cu mulțimea problemelor generate spre rezolvare.

În fig. 1 pot fi observați pașii de selectare, generare a unei familii de probleme decizionale, după criteriile dorite de titularul cursului.

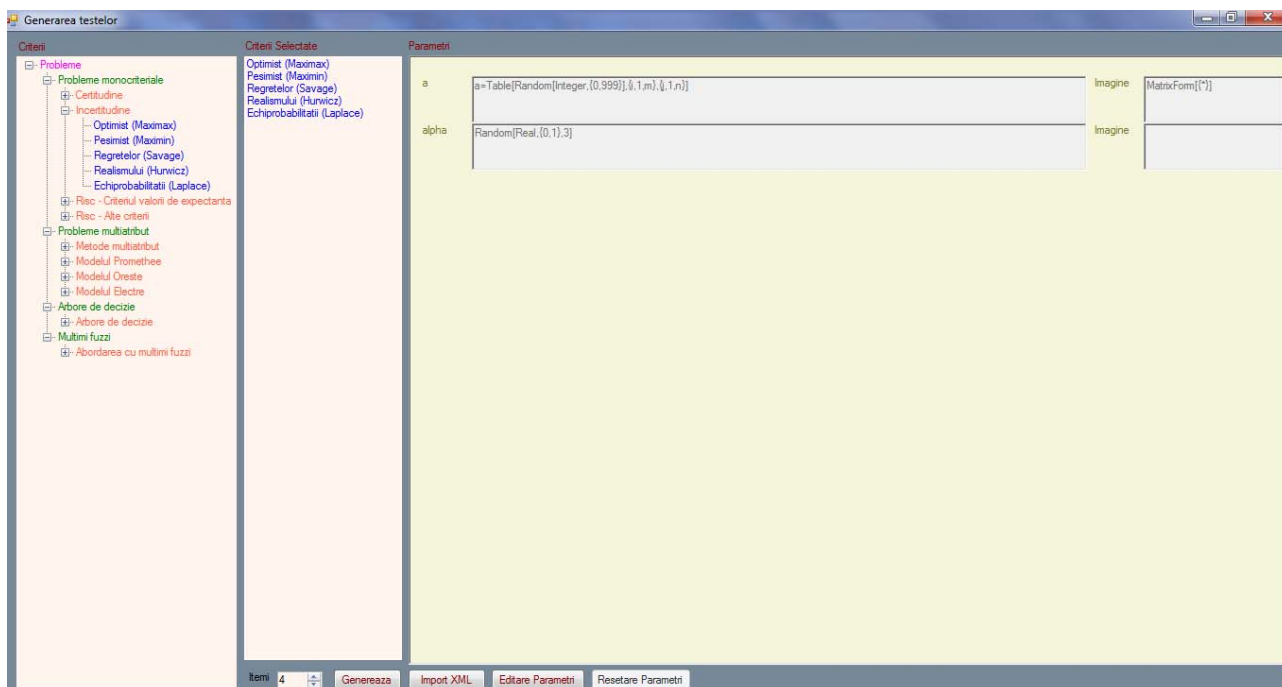


Figura 1. Generarea testelor.

În timpul selectării familiei de probleme decizionale din coloana *criterii*, în coloana *criterii de selecție*, se transferă automat doar criteriile specificate în prima coloană. În coloana *parametri*, apar parametrii ce vor fi folosiți la rezolvarea problemelor decizionale.

După ce au fost selectate datele necesare se alege numărul de itemi (probleme decizionale) ce vor fi generați în test (a se vedea fig. 1, butonul *itemi*), apoi se apasă butonul *generare*. Generatorul de probleme operativ elaborează datele de intrare și rezultatele problemelor generate ce sunt prezentate în fig. 2.

Nr.	Matricea de decizie	Coeficientul de pondere alpha	Optimist (Maximax)	Pesimist (Maximin)	Regretelor (Savage)	Realismului (Hurwicz)	Echibilității (Laplace)
1	{{788, 300, 361, 418}, {343, 276, 66, 780}, {700, 129, 275, 595}}	0.599	{"A>1.", "V->788."}	{"A>1.", "V->300."}	{"A>3.", "V->185."}	{"A>1.", "V->592.312"}	{"A>1.", "V->466.75"}
2	{{613, 901, 313, 19}, {51, 404, 725, 676}, {890, 630, 767, 709}}	0.782	{"A>1.", "V->901."}	{"A>3.", "V->630."}	{"A>3.", "V->271."}	{"A>3.", "V->833.32"}	{"A>3.", "V->749."}
3	{{673, 234, 815, 406}, {110, 195, 619, 395}, {723, 637, 39, 538}}	0.554	{"A>1.", "V->815."}	{"A>1.", "V->234."}	{"A>1.", "V->403."}	{"A>1.", "V->555.874"}	{"A>1.", "V->532."}
4	{{282, 944, 368, 703}, {275, 333, 509, 894}, {637, 287, 29, 671}}	0.576	{"A>1.", "V->944."}	{"A>1.", "V->282."}	{"A>1.", "V->355."}	{"A>1.", "V->663.312"}	{"A>1.", "V->574.25"}

Figura 2. Rezultatele generării.

### 3. REZOLVETOARELE DE PROBLEME DECIZIONALE

*Rezolvatoarele de probleme* asistă profesorul în procesul evaluării testelor și/sau a problemelor rezolvate de către cursanți. Rostul rezolvatoarelor de probleme este de a determina: „Este corect sau nu răspunsul studentului la testul propus?”

Un rezolvitor de teste reprezintă un produs inteligent pentru eLearning din motiv că construiește automat în calculator algoritmul rezolvării fiecărui test în momentul formulării testului corespunzător folosind cunoștințele referitoare la modelul formal al temei corespunzătoare.

Un rezolvitor de teste este un program de inteligență artificială: a) *faptele* programului sunt *axiomele teoriei formale*; b) *producțiile – regulile de inferență*; c) *întrebările* (numite și *scopuri*) – *problemele și/sau testele* propuse studenților pentru rezolvare, d) în momentul formulării de către utilizatorul final al unui *scop (test)* programul elaborat în C# cu ajutorul pachetului Mathematica automat elaborează algoritmul soluționării acestui test; e) soluțiile obținute (una, mai multe sau nici una) pot fi tratate ca *teoreme demonstrate* în cadrul

teoriei formale corespunzătoare temei de examinat, iar rezolvitorul de teste – ca un *sistem de demonstrare automată a teoremelor* în această teorie formală; f) programul în C# asistat de pachetul Mathematica în execuție reprezintă un rezolvitor de probleme.

Rezolvarea unei de familii de probleme decizionale cu ajutorul SSI se face conform următoarelor etape.

1. Pachetul Mathematica procesează informația primită de la SSI (rezolvă, modelează problema decizională, problemele propuse) și întoarce rezultatele SSI.
2. SSI unește (încapsulează) problema și rezultatele obținute în p 1.
3. Datele de ieșire sunt transmise utilizatorului final în format XML și XLS.

În figura 2 pe lângă faptul că pot fi vizualizate datele de intrare și rezultatele obținute se mai pot face și unele operații și anume: informația primită poate fi exportată în Xls (Excel) sau în Xml cu opțiunile: toate problemele pe o singură pagină, pe pagini aparte, cu includerea tuturor parametrilor sau doar a datelor de intrare.

Din motiv, că Fișierul Xml este compatibil cu platforma educațională MOODLE, problema consumului de timp considerabil necesar pentru

generarea și verificarea testelor la disciplina "Sisteme Suport pentru Decizii" – problemă discutată la începutul articolului a fost soluționată.

## CONCLUZII

A fost elaborat un SSI în calitatea de suport didactic la Disciplina „Sisteme Suport pentru Decizii”. SSI generează automat numărul necesar de itemi necesari studenților atât în cadrul lucrărilor de laborator cât și la evaluarea sumativă. În acest scop SSI operează cu următoarele concepte: *a)* familii de probleme decizionale, *b)* modele generice ale familiilor de probleme decizionale, *c)* modele specifice ale problemelor decizionale, *d)* generatoare de probleme decizionale, și *e)* rezolvitoare de probleme decizionale.

SSI demonstrează următoare calități:

- este conceput pentru a asista procesul de instruire a studenților la disciplina „Sisteme Suport pentru Decizii”;
- asigură în procesul de instruire la această disciplină interacțiunea flexibilă *student-calculator* și *calculator-profesor*;
- generează automat specificările lucrărilor de laborator (testelor) și evaluează automat rezultatele studenților în cadrul acestor activități la disciplina „SSD”.

SSI poate, de asemenea, asista procesele de luare a deciziilor în diverse domenii de activitate și în primul rând cel economic.

Un studiu de caz referitor la generarea, modelarea și rezolvarea unei familii de probleme decizionale cu ajutorul SSI, precum și o caracterizare mai amplă a structurii și principiului de funcționare a SSI a fost expusă în lucrarea [3].

## Bibliografie

1. **Beldiga M.** *Analiza comparativă a platformelor de e-Learning. Studias Universitatis Nr.2, Chișinău, pag. 31-34, 2012.*
2. **Mendelson E.** *Introduction to Mathematical Logic. 2nd. ed. New York: D. Van Nostrand, 1979.*
3. **Beldiga M., Căpățână Gh.** *Produse inteligente pentru asistența testării cunoștințelor la disciplina "Sisteme Suport pentru Decizii". // În: The 20<sup>th</sup> Conference on Applied and industrial mathematics dedicated to academician Mitrofan M. Cioban, Chișinău, pag. 134-138, 2012.*