



"Imaginatie, Creativitate,  
Design, Dezvoltare"



Lucrarile  
Sesiunii Nationale de Comunicari  
Stiintifice Studentesti  
Sibiu

ICDD 2009

# **IMAGINAȚIE, CREATIVITATE, DESIGN, DEZVOLTARE**

**Lucrările  
Sesiunii Naționale de Comunicări Științifice Studentești  
Sibiu, 10 – 11 Aprilie, 2009**

**UNIVERSITATEA “LUCIAN BLAGA” SIBIU, FACULTATEA DE ȘTIINȚE  
CATEDRA DE INFORMATICĂ**

# **Lucrările Sesiunii Naționale de Comunicări Științifice Studentești “Imaginație, Creativitate, Design, Dezvoltare”**

**Editor** *Conf. Univ. Dr. Dana Simian*

## **Comitet Științific**

*Prof. Univ. Dr. Valer Roșca* - Universitatea “Lucian Blaga” Sibiu

*Conf. Univ. Dr. Dana Simian* - Universitatea “Lucian Blaga” Sibiu

*Lector Univ. Dr. Florin Stoica* - Universitatea “Lucian Blaga” Sibiu

*Lector Univ. Dr. Ioan Pop* - Universitatea “Lucian Blaga” Sibiu

*Lector Univ. Dr. Mircea Neamțu* - Universitatea “Lucian Blaga” Sibiu

## **Colectiv tehnoredactare**

*Conf. Univ. Dr. Dana Simian*

*Prep. Laura Cacovean*

*Stud. Ciovică Laurențiu*

## **Design copertă**

*Stud. Liviu Ciovică*

**ISSN 2065 – 927X**

**Motto:**

*“There are no limits, only your imagination”*

# Prefață

Prezentul volum reunește lucrările prezentate la prima ediție a Sesiunii Naționale de Comunicări Științifice Studentești “Imaginație, Creativitate, Design, Dezvoltare”, desfășurată la Universitatea Lucian Blaga din Sibiu, în perioada 10-11 Aprilie 2009, organizată de către Catedra de Informatică din cadrul Facultății de Științe.

Scopul sesiunii de comunicări este de a reuni studenți din toate centrele universitare din țară pentru a prezenta și a discuta rezultate originale obținute în toate ariile tematice ale domeniului informatică: informatică teoretică, algoritmică, proiectarea și construirea de software, trimiterea datelor prin rețele, noi abordări în probleme de securitate, etc.

Sesiunea științifică include și o secțiune specială dedicată elevilor de liceu.

Mulțumim tuturor participanților, colectivului de organizare și colectivului științific, pentru contribuția adusă la succesul acestei manifestări științifice și la realizarea prezentului volum.

Conf. Univ. Dr. Dana Simian

# CUPRINS

|  |    |
|--|----|
| <b>Partea 1 - Secțiunea dedicată studenților</b> .....   | 3  |
| <i>Microsoft Excel ca instrument de planificare grafică a activităților</i><br>Francesco Capuzzo .....             | 4  |
| <i>Compiler COOL pentru arhitectura MIPS</i><br>Alin – Mircea Danciu .....   | 12 |
| <i>Aplicație geometrică în Matlab</i><br>Ana Maria Dragut, Cristian Florea, Simina Lunceanu .....                  | 18 |
| <i>Platformă e-Learning</i><br>Ancuța-Raluca Epure .....   | 33 |
| <i>Sistem de management și gestiune a fișierelor</i><br>Iulia – Maria Lazăr .....                                  | 39 |
| <i>Image Deblurring</i><br>Valentina Lazăr .....   | 44 |
| <i>Sistem de Fișiere Multimedia</i><br>Sorin Lacrițeanu, Tudor Miu .....   | 48 |
| <i>Matlab - COM Client Support – Examples of Matlab as an Automation Client</i><br>Ramona Marinela Nițu .....      | 54 |
| <i>Metode de procesare a textului pentru sinteză vocală</i><br>Mihai Stancu .....                                  | 62 |
| <i>Mobile Earth</i><br>Lucian Stoica .....   | 66 |
| <i>Construirea de Jocuri Interactive in Matlab</i><br>Cosma Ștefan, Andreea Firescu, Vasile Nicușor Nechifor ..... | 73 |
| <i>Joc de Strategie</i><br>Sopa Christian .....  | 77 |

|  |           |
|--|-----------|
| <b>Partea 2 - Secțiunea dedicată elevilor (abstracte extinse) .....</b>  | <b>80</b> |
| <i>Aplicarea programei școlare în învățământul preuniversitar</i><br>Profesor Georgeta Preda .....   | 81        |
| <i>Desfășurarea unor activități practice, educative pentru formarea de abilități de comunicare și tehnice în secolul XXI</i><br>Profesor Monica Oancea ..... | 85        |
| <i>Fotbal Logic</i><br>Victor Badea .....  | 87        |
| <i>Graful hamiltonian, program oprire calculator</i><br>Dorin-Ionuț Cherciu .....  | 89        |
| <i>Data Modeling: Environmental Friendly Technology</i><br>Alexandru Cristian, Mareș Radomir .....   | 91        |
| <i>Teoria grafurilor reprezentată grafic în C++</i><br>Alexandru Cristian, Bogdan Diaconescu, Mareș Radomir .....  | 93        |
| <i>Fields of Gold</i><br>Claudiu Iaru, Eduard Bonci, Flaviu Bulat, Luana Făgărășan .....   | 94        |

**Sesiunea de Comunicări Stiințifice a Studenților  
"Imaginație, Creativitate, Design, Dezvoltare"  
Sibiu – România, 2009**

## **PARTEA 1**



## Microsoft Excel ca instrument de planificare grafică a activităților

**Autor:** Francesco Capuzzo, Universitatea "Lucian Blaga" Sibiu,  
Facultatea de Științe – profil Informatică,  
**Coordonator didactic:** Conf. Univ. Dr. Dana Simian

*Abstract: the present work wants to clarify my previous works mentioned in the bibliography. The procedure is described step by step. Anyone can use self made formulas to implement this work in a personal manner.*

*Cuvinte-cheie: planning, operational, production.*

### 1. Introducere

Lucrarea de față este o complectare a lucrărilor trecute în bibliografia. Foaia de calcul Microsoft Excel poate fi folosită pentru a obține o afișare grafică a unui set de activități. În practică, aplicabilitatea ei se poate desfășura, de exemplu:

- într-un mediu de producție industrială
- într-o firmă de construcții de proporții mari, care execută diferite lucrări în același timp
- etc.

Factorii principali care trebuie ținuți în considerare sunt:

Tabelul 1

| Nr. | Pentru o firmă de producție industrială                                | Pentru o firmă de construcții mare           |
|-----|--|--|
| 1   | Centru de lucru (manual/mecanic)                                       | Echipă de muncitori                          |
| 2   | Articole de produs   | Lucrări de executat                          |
| 3   | Timpu transformării/prelucrării/controlării/ambalării fiecărui articol | Timpu efectuării fiecărei lucrare/sublucrare |
| 4   | Termenul predării  | Data/ora programată                          |

### 2 Formularea Problemei

Referitor la tabela de mai sus, la punctul unu avem resursele. La punctul doi avem obiectul/obiectele de lucru ale resurselor. La punctul trei, timpul este considerat o constantă care poate să fie fixată de la bun început sau o variabilă care trebuie stabilită și reactualizată la fiecare operație efectuată. La punctul patru, termenul trebuie să fie o dată sau dată/oră bine stabilită, dar se poate modifica ușor; reactualizarea sistemului va recalcula toate celelalte evenimente.

Cerințele primite se vor suprapune cu siguranță, cel puțin în momentul primirii. Pentru cazul firmei de producție industrială se vor primi, pe una sau mai multe comenzi, o serie de articole care pot avea același termen de execuție și vor trebui prelucrate în același centru de lucru. Acestea trebuie planificate în mod de a exclude posibilitatea de a le suprapune.

Termenul este cea variabilă care trebuie prelucrată în așa fel încât să nu existe mai multe elemente din punctul doi din tabela de mai sus care să se suprapună. În cazul firmei de producție, de exemplu, nu se va putea executa prelucrarea a două sau mai multor articole, în același timp, în același centru de lucru. Asemenea, pentru cazul firmei de construcție, muncitorii la o locație nu vor putea lucra în același timp și în altă locație.

Pentru cazurile specificate nu este de ajuns să se folosească un simplu calendar pentru a organiza activitățile. Din contră trebuie folosit un instrument informatic care să permită, pe de o parte, înlănțuirea activităților, pe de altă parte folosirea continuă a resurselor. Nevoia aceasta se naște de la faptul că activitățile pot fi de lungă durată și, odată făcute modificările, acestea ar trebui recalculat manual pe un calendar. Va fii cu totul altfel folosind conceptele prezentate în lucrarea de față.

Microsoft Excel ne permite să obținem în formă grafică încărcarea de lucru a resurselor. Foarte important este faptul că resursele trebuie folosite din plin. Orice oprire reprezintă un cost, fie mașini, fie muncitori.

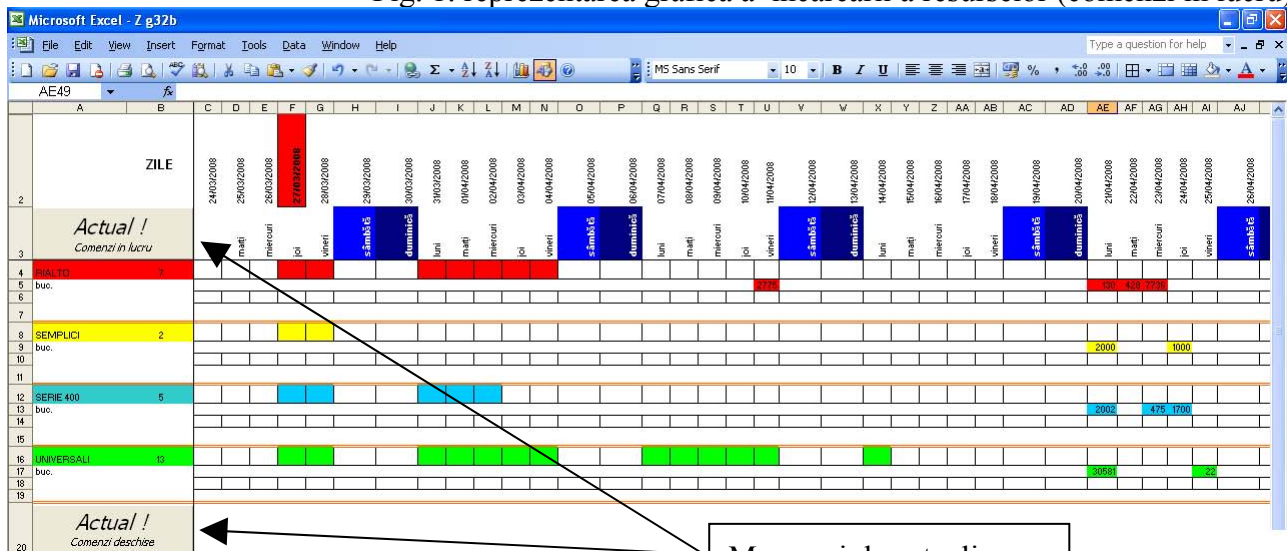
Nu este necesar să folosim limbajul VBA pentru astfel de planificare, fiindcă macrourele, în momentul creării, generează codul respectiv în mod automat. Limbajul VBA poate, totuși, să fie folositor pentru a îmbunătăți toată lucrarea și pentru a adăuga caracteristici utile.

Obiectul de lucru din punctul de vedere a resurselor scade, cantitativ, cu trecerea timpului. Acest fapt trebuie cuantificat. Practic, la sfârșitul zilei, lista lucrărilor de executat se va scădea cantitativ în funcție de activitatea zilei.

Afișarea finală (fig. 1), împărțită pe 4 resurse de tipologie diferită, va arăta după cum urmează:

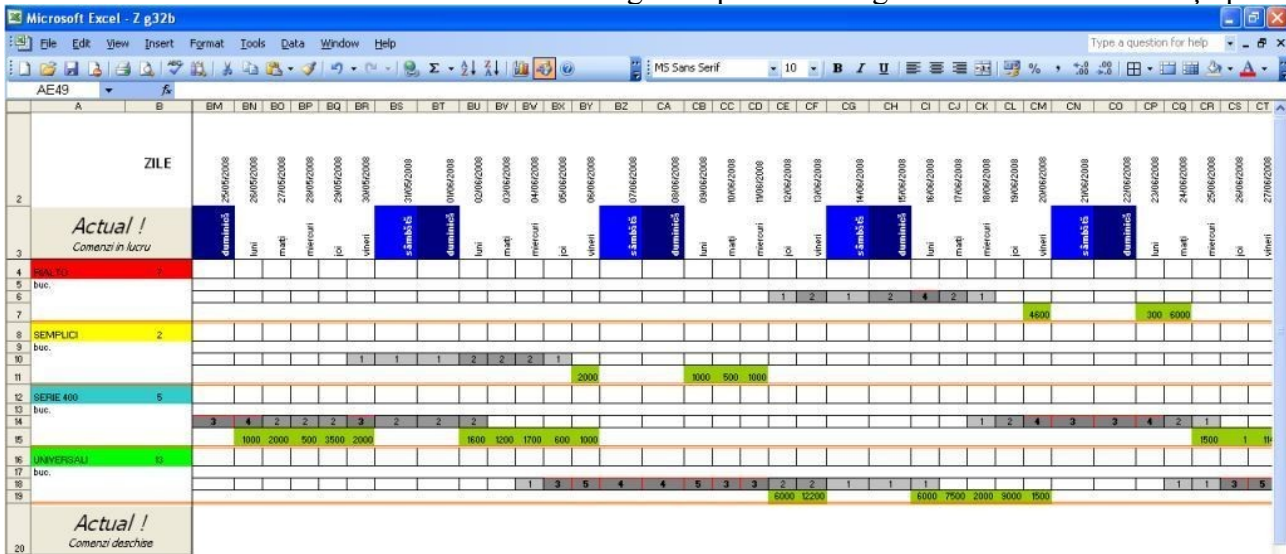
1. încărcarea resurselor, din data de astăzi, până la terminarea lucrării (linie 4, 8, 12, 16)
2. cantitatea de articole de produs în funcție de data maximă utilă pentru a le putea trimite la timp. Pentru firmele de construcție, această informație va fi înlocuită cu obiectul activității (linie 5, 9, 13, 16).

Fig. 1: reprezentarea grafică a încărcării a resurselor (comenzi în lucru)



Comenziile în așteptare sunt de asemenea importante, cum se vede în fig. 2.

Fig. 2: reprezentarea grafică a comenzilor în așteptare



O reprezentare grafică de acest gen permite să se reacționeze foarte repede la apariția unei modificări. Cazurile pot fi diferite: schimbarea unui termen de predare, adăugarea unui articol/activitate de urgență mai ridicată, apariția unei defecțiuni la o mașină sau îmbolnăvirea unui muncitor etc. De asemenea, cu o privire, putem afla pentru câte zile o resursă este ocupată, când trebuie să ne preocupăm să punem în lucru următoarea activitate, dacă o activitate activă se va sfârși după data maximă a începerii unei activități în așteptare (situație care ne arată că planificarea a fost făcută în mod greșit sau au fost acceptate comenzi care nu vom putea onora fără să recurgem la măsuri extraordinare, cum ar fi orele suplimentare sau repartizarea obiectului muncii la terți).

## 2.1 Procedeu

Toată lucrarea poate încăpea într-o singură foaie de calcul. În două pagini separate vom avea comenzi în lucru și comenzi în așteptare. În fig. 4 urmează un exemplu pentru un mediu de producție industrială. În coloana a cincea, se poate observa cum sunt repartizate articolele în funcție de resursa respectivă. Aceste informații pot fi importate foarte ușor din sistemul de gestiune folosit pentru administrare.

Fig. 3: exemplu de comenzi în lucru/în așteptare

| 1  | cod            | denumire                           | termexec   | cantitate | centru de lucru |
|----|----------------|------------------------------------|------------|-----------|-----------------|
| 10 | I.1292.N*105   | PRIZA MOBILA MULTIPLA 4U UNIV. CA  | 21/04/2008 | 1000      | universali      |
| 11 | I.1293*104     | PRIZA UNIV. 5 LOCASURI ALBA (NA)   | 21/04/2008 | 972       | universali      |
| 12 | I.1293*104     | PRIZA UNIV. 5 LOCASURI ALBA (NA)   | 21/04/2008 | 378       | universali      |
| 13 | I.1293*104     | PRIZA UNIV. 5 LOCASURI ALBA (NA)   | 21/04/2008 | 4962      | universali      |
| 14 | I.1294*104     | PRIZA MOB.MULT. 5U UNIV CABLU AV   | 21/04/2008 | 21        | universali      |
| 15 | I.0403.C*103   | PRIZA MULTIPLA 3LOCASURI NEAGRA    | 21/04/2008 | 1000      | serie 400       |
| 16 | I.0406.B*102   | S.IMB.PRIZA MULT. 6BP B            | 21/04/2008 | 1000      | serie 400       |
| 17 | I.0406.C*103   | S.IMB.PRESA MULT. 6BP C/N          | 23/04/2008 | 475       | serie 400       |
| 18 | I.0406.C.B*103 | S.IMB.PRESA MULT. 6BP C/B          | 30/04/2008 | 1600      | serie 400       |
| 19 | I.0407.C.F*100 | PRIZA MULT.6P17/11+S17 FUMURIE     | 24/04/2008 | 1700      | serie 400       |
| 20 | I.1243*105     | PRIZA TRIPLA B (NA)                | 21/04/2008 | 30000     | simplici        |
| 21 | I.1244*105     | PRIZA MULTIPLA 3P17/11 CU CABLU CI | 24/04/2008 | 1000      | simplici        |
| 22 | I.1287*102     | PRIZA UNIV.3 LOCASURI(NA)          | 21/04/2008 | 3600      | universali      |
| 23 | I.1291*105     | PRIZA MOB. MULT. 4U UNIV. ALBA     | 21/04/2008 | 1248      | universali      |
| 24 | I.1294*104     | PRIZA MOB.MULT. 5U UNIV CABLU AV   | 21/04/2008 | 3600      | universali      |
| 25 | 05695*103      | DEVIATOR 1P 6A CREM                | 20/04/2008 |           | rialto          |
| 26 | 05695*103      | DEVIATOR 1P 6A CREM                | 21/04/2008 | 130       | rialto          |
| 27 | 05656.B*103    | DEVIATOR 1P 6A ALB                 | 23/04/2008 | 476       | rialto          |
| 28 | 05656.B*103    | DEVIATOR 1P 6A ALB                 | 22/04/2008 | 428       | rialto          |
| 29 | 05656*103      | DEVIATOR 1P 6A CREM                | 11/04/2008 |           | rialto          |

Ready

Este necesar să fie la dispoziție, într-o altă foaie de calcul, norma la fiecare articol/lucrare, după cum se vede în fig. 5

Fig. 4: norma

| 1      | A            | B     | C          | H |
|--------|--------------|-------|------------|---|
| nr crt | ART          | BUC/h | centru c   |   |
| 35     | 34 00409.C.B | 150,5 | serie 400  |   |
| 36     | 35 00424     | 245   | serie 400  |   |
| 37     | 36 00424.B   | 245   | serie 400  |   |
| 38     | 37 00424.C   | 157,5 | serie 400  |   |
| 39     | 38 00424.C.B | 157,5 | serie 400  |   |
| 40     | 39 00424.K   | 157,5 | serie 400  |   |
| 41     | 40 00424.K.B | 157,5 | serie 400  |   |
| 42     | 41 00426.C.B | 157,5 | serie 400  |   |
| 43     | 42 00426.K.B | 157,5 | serie 400  |   |
| 44     | 43 01243     | 445   | simplici   |   |
| 45     | 44 01243.N   | 445   | simplici   |   |
| 46     | 45 01244     | 237,5 | simplici   |   |
| 47     | 46 01244.N   | 237,5 | simplici   |   |
| 48     | 47 01261     | 425   | simplici   |   |
| 49     | 48 01261.N   | 425   | simplici   |   |
| 50     | 49 01262     | 237,5 | simplici   |   |
| 51     | 50 01262.N   | 237,5 | simplici   |   |
| 52     | 51 01263     | 420   | simplici   |   |
| 53     | 52 01264     | 237,5 | simplici   |   |
| 54     | 53 01264.N   | 237,5 | simplici   |   |
| 55     | 54 01287     | 350   | universali |   |
| 56     | 55 01287.N   | 350   | universali |   |
| 57     | 56 01290     | 270   | universali |   |
| 58     | 57 01290.N   | 270   | universali |   |
| 59     | 58 01291     | 330   | universali |   |
| 60     | 59 01291.N   | 330   | universali |   |
| 61     | 60 01292     | 260   | universali |   |

Cum se arată în fig. 6, o pagina va fi dedicată calculării datei maxime pentru executarea lucrării, fie producție sau altă activitate. Data respectivă cuprinde ținerea în calcul a tuturor articolelor/activităților care trebuie executate de către o resursă. În cazul unui mediu de producție industrială, această dată va reprezenta termenul maxim de producție la articolul celui cu termen mai îndepărtat. Fig.5, la căsuțele J32-J35, afișează rezultatele pentru patru resurse. Folosind funcții de tip datetime, calculul ține cont numai de orele zilei în care firma este deschisă. Se exclud nopțile, pauzele, sfârșiturile de săptămână și concediile.

Fig. 5: calcularea datei maxime pentru executarea lucrării

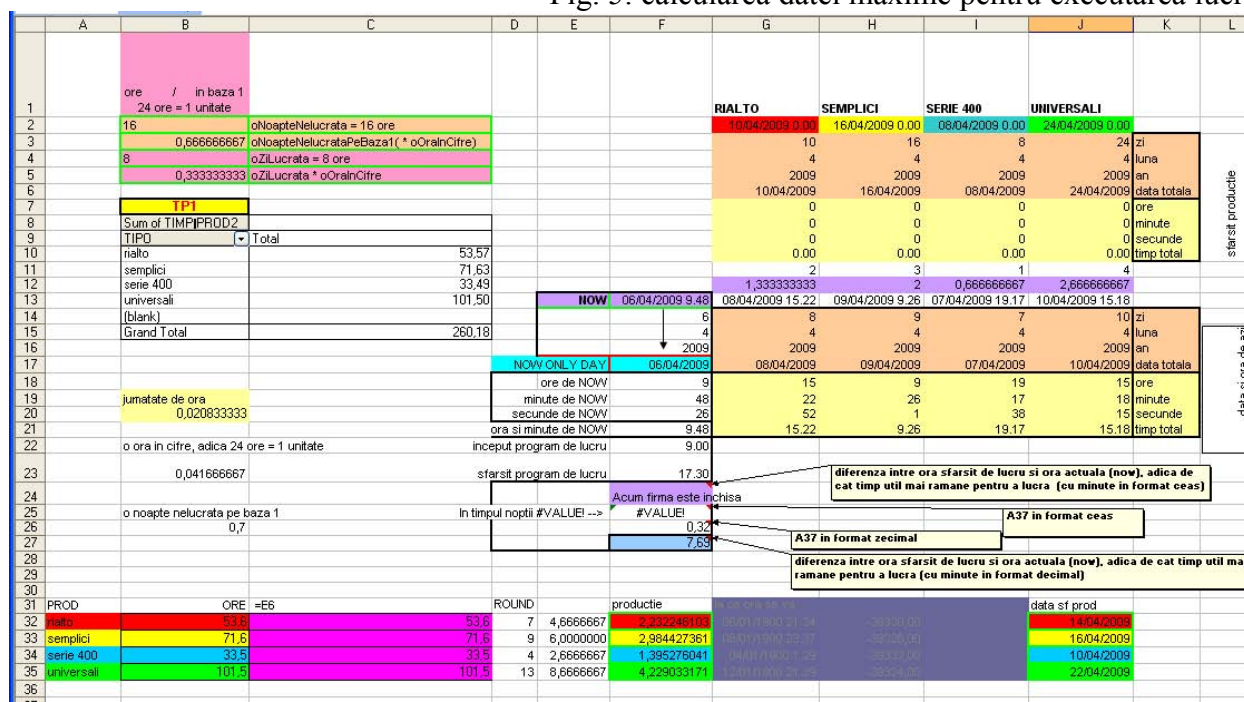
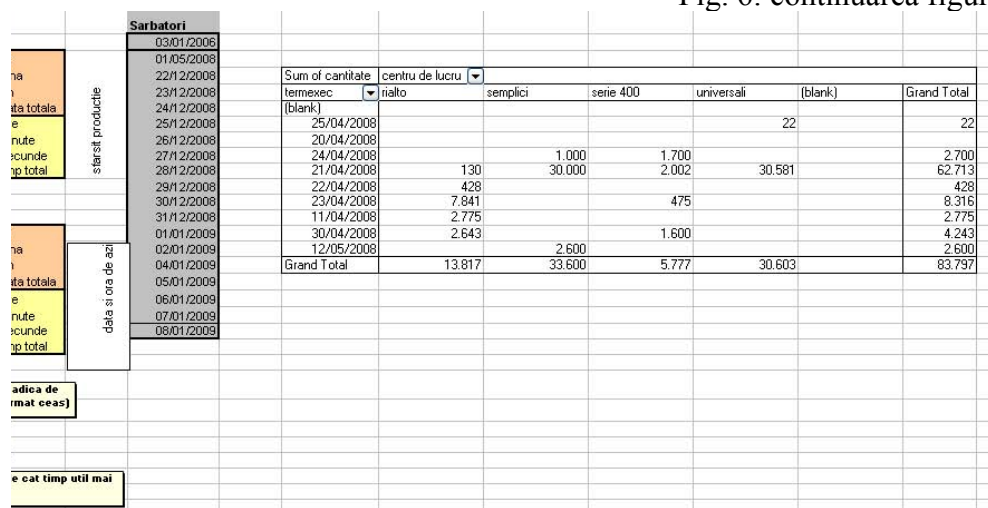


Fig. 6: continuarea figurii 4, spre dreapta



Într-o pagină separată pentru fiecare resursă, se execută toate calculele care duc la înlănțuirea activităților.

Fig. 7: să se vadă textul de mai jos

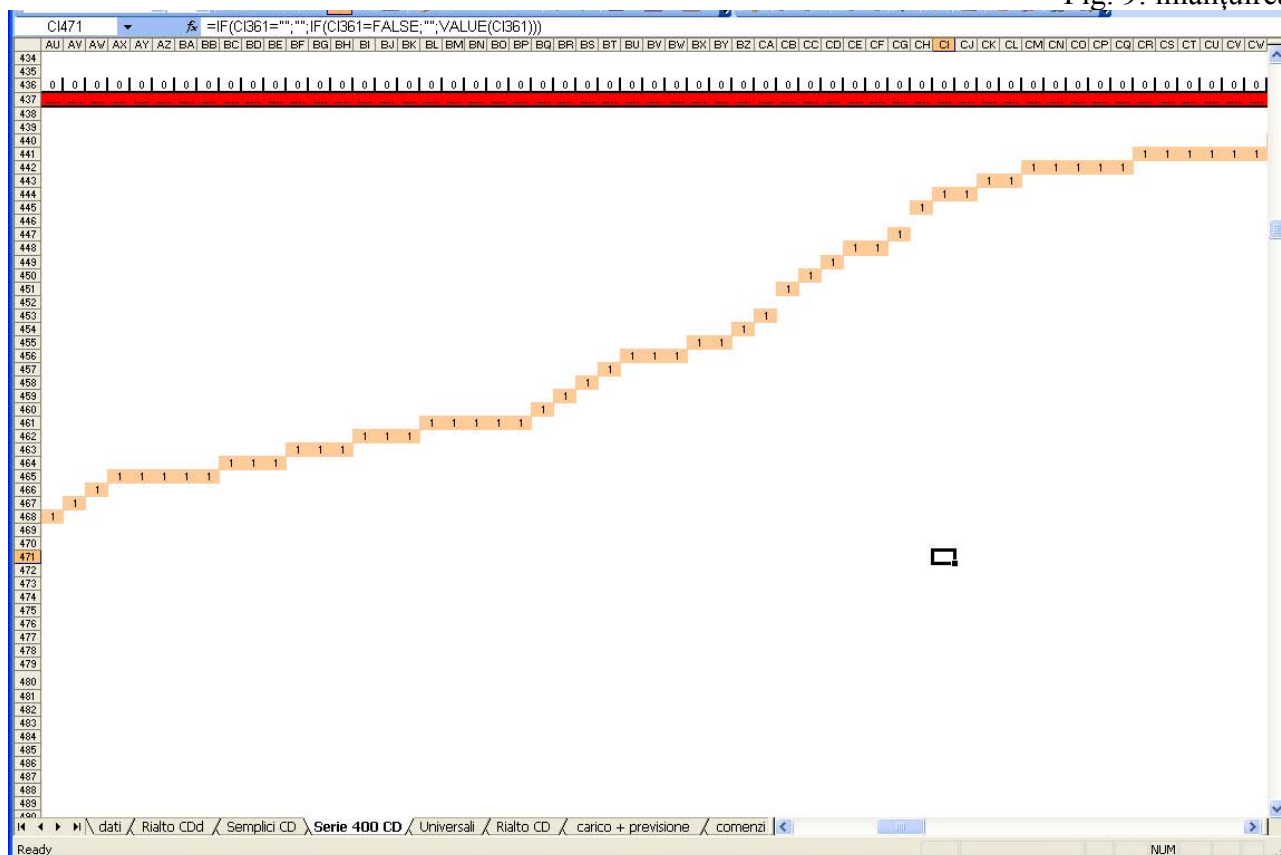
The screenshot shows an Excel spreadsheet with a Gantt chart. The main title is 'Serie 400 - comenzi deschise'. The chart area is filled with horizontal bars representing task durations. The task list on the left includes columns for task name, start date, end date, and other details. The right pane shows task details for a selected task, including its name, start date, end date, and other attributes. The spreadsheet interface includes standard Excel elements like the ribbon, formula bar, and status bar.

Fig. 8: continuare spre dreapta din fig. 5

This screenshot continues the Gantt chart from Figure 5. It shows a large number of tasks, each represented by a horizontal bar in the chart area. The task list on the left is extensive, listing tasks with their respective start and end dates. The right pane shows the details for a selected task. The overall layout is consistent with the previous figure, showing a detailed project schedule in Excel format.

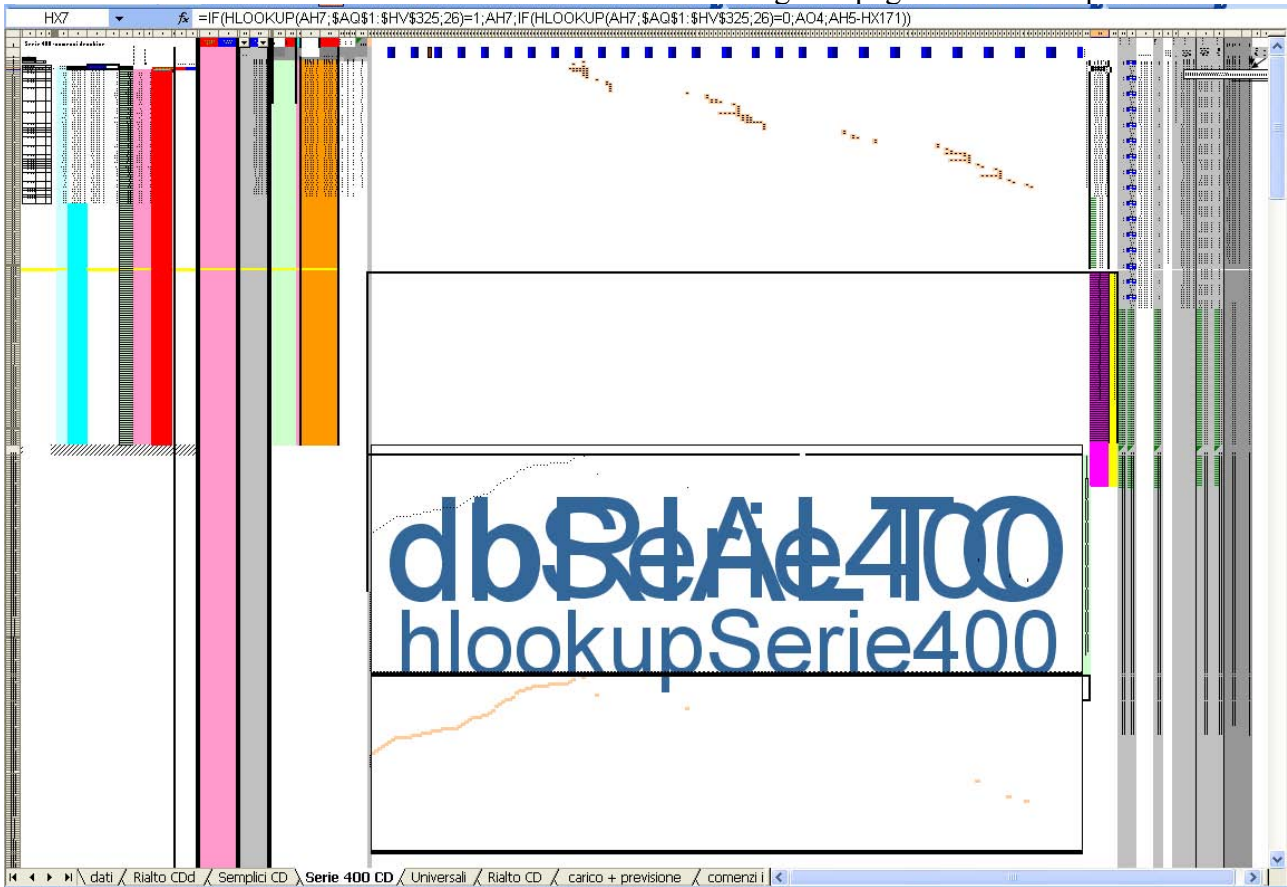
În fig.7 se calculează, pentru fiecare articol/lucrare, data în care ar trebui început lucrul pentru a fi terminat în timp util pentru livrare. În fig. 8, se vede reprezentarea grafică a perioadelor de lucru, pentru fiecare articol/activitate. În coloanele AJ/AK, se face înlănțuirea propriu zisă, care va fi arătată în fig. 9. Sub imaginea din figură 9 vor fi două zone de calcule. În partea dreaptă mai sunt coloane cu formule care nu sunt afișate în poză.

Fig. 9: înlănțuirea



În fig. 9, fiecare celulă care conține un „1” reprezintă o zi de lucru. Fiecare celulă afișată în fig. 9 conține o formulă. De asemenea zonele de sub ea, care sunt afișate în fig. 10. Următorul pas este afișarea finală din fig. 1, unde acest lanț este întrerupt și lungit din pricina ținerii în considerare a sfârșiturilor de săptămână și concedii. Urmează o imagine de ansamblu a unei pagini de calcul pentru o resursă.

Fig. 10: pagina de calcule pentru o resursă



### 3 Rezultate principale

Lucrarea de față este folosită în prezent la firma Vimar România pentru a planifica producția. Ea oferă situația la zi în orice moment și, prin flexibilitatea ei, poate fi modificată ușor. De asemenea se pot modifica datele de intrare și cu apăsarea unui singur buton se reactualizează toate formulele, macrouri și tabele Pivot.

### 4 Concluzii și dezvoltări viitoare

Pasul următor va fi adaptarea lucrării la o firmă de construcții care execută mai multe lucrări. Vreau să încerc s-o testez cu un serviciu de free spreadsheet cum este zoho.com, pe Internet, și s-o transform într-un produs comercial ușor de întreținut și adaptat.

### Bibliografie

1. Capuzzo F., Planificarea grafică a producției folosind Microsoft Excel, Editura Universității „1 Decembrie 1918”, Alba Iulia, 2008
2. Capuzzo F., Gestiunea activităților în timp cu Microsoft Excel: formule, macro-uri și dezvoltare grafică, Universitatea din Petrosani, Petrosani, 2008

**Capuzzo Francesco**  
**Universitatea „Lucian Blaga” din Sibiu**  
**Facultatea de Științe**  
**Specializarea Informatica, anul II**  
e-mail: francesco.capuzzo@gmail.com



## Compiler COOL pentru arhitectura MIPS

**Autor:** Danciu Alin – Mircea

*Abstract: Lucrarea își propune prezentarea unui compiler complet pentru limbajul COOL, cu target arhitectura MIPS, scris în java și ANTLR.*

*Keywords : compiler, COOL, MIPS, AST, generare cod, Kildall*

### 1. Introducere

Compilerul este primul program care este scris atunci când un limbaj nou ia naștere, fiind folosit pentru crearea de software de la sisteme de operare, aplicații grafice până la programarea echipamentelor periferice industriale și a microcontrolerelor. La ora actuală există o multitudine de compilatoare, cel mai cunoscut fiind probabil faimosul gcc.

Ceea ce definește un compiler este în primul rând limbajul (sau limbajele) pe care acest compiler le recunoaște și arhitectura pentru care este construit. Orice compiler folosește instrucțiunile specifice arhitecturii pentru care este scris pentru a transforma programul sursă, din limbajul compilat în cod obiect specific arhitecturii.

### 2. Arhitectura MIPS

O arhitectura reprezintă modul în care funcționează un sistem de calcul, în special modul de construcție și operare al procesorului. Prezentarea arhitecturii MIPS nu face obiectul lucrării de față, pentru o prezentare completă se pot consulta [1], [2] și [3]. Aici ne vom rezuma la prezentarea unei scheme bloc a procesorului MIPS – prezentat în figura 1. Cu toate acestea unele elemente ale arhitecturii vor fi prezentate, deoarece au un rol deosebit în deciziile luate în arhitectura compilerului.

Arhitectura MIPS a fost proiectată inițial pe 32 de biți, deși la ora actuală există și variante pentru 64 de biți. Astfel a fost ales tipul de date întreg în java pentru a corespunde cu dimensiunea cuvântului de pe arhitectura MIPS. La ora actuală există mai multe seturi de instrucțiuni MIPS printre care MIPS I, MIPS II, MIPS V, MIPS 32 și MIPS 64. Compilerul folosește setul de instrucțiuni MIPS 32. La acest set de instrucțiuni pot fi adăugate extensiile noi, corespunzătoare capacităților hardware introduse în procesoarele mai noi MIPS, MIPS-3D, MDMX și MIPS MT. Această variantă de compiler nu suportă nici una dintre aceste extensii ele fiind utilizate în practică doar în sistemele embeded, pentru aplicația multimedia.

În al doilea rând arhitectura MIPS pune la dispoziția compilerului un număr mare de registre de uz general. Acesta simplifică etapa de alocare a registrelor, care este una dintre cele mai importante optimizări în faza de IR. Registrele MIPS sunt prezentate în tabelul 1. După cum se observă din acest tabel toți registrii marcați cu <nu> pot fi folosiți în orice funcție fără a fi necesară salvarea lor explicită de către compiler la apelul funcției. În plus din punctul de vedere al compilerului registrii temporari pot fi folosiți în alocarea registrelor în cadrul analizelor inter-procedurale pentru a optimiza fluxul de date între funcții.

### 3. Structura limbajului COOL

Limbajul COOL (Classroom Object Oriented Language) este un limbaj orientat obiect, cu o structura simpla, dar care prezinta unele particularitati inexistente in limbaje uzuale orientate obiect cum ar fi Java sau C#. Asemănător cu secțiunea 2, lucrarea nu își propune o prezentare completa a limbajului COOL ci doar trece în revista unele aspecte importante și relevante în proiectarea unui compilator pentru acest limbaj. Pentru o prezentare completa se pot consulta [4] și [5].

COOL este un limbaj puternic tipat, fiecare definiție de membru (denumit în COOL atribut) sau metoda (denumită în COOL metoda) trebuie să aibă un tip precizat la compilare (static type). În plus, la executarea programului type-checkerul trebuie să verifice tipul dinamic al variabilei respective și să nu permită executia codului dacă tipul dinamic nu este conform cu tipul static. Runtime type checking trebuie implementat de către compilator, în conformitate cu specificațiile din [4]. Excepțiile trebuie tratate conform specificațiilor de runtime din [5]. O altă particularitate a limbajului este prezența lui SELF\_TYPE, acesta este un tip special de variabilă care este mapat la runtime peste tipuri generice. El permite implementarea unor facilități interesante în COOL (un exemplu sunt listele înlanțuite și un program demonstrativ este prezent în anexe). Din punctul de vedere al compilatorului ceea ce este relevant este sintaxa limbajului COOL care este prezentată în figura 2. Aceasta împreună cu informațiile de runtime din [2] sunt elementele esențiale pe baza cărora este construit compilatorul de COOL.

### 4. Arhitectura compilatorului

Compilatorul COOL este organizat în 5 faze : analiză lexicală, analiză sintactică, analiză semantică, generarea de cod intermediar și optimizarea.

Analiza lexicală presupune recunoașterea atomilor lexicali din textul sursă (codul programului) și este implementată folosind o gramatică lexicală de tip ANTLR. Pentru o prezentare a ANTLR vezi [6]. Textul sursă este transformat într-un șir de atomi lexicali, iar erorile lexicale sunt raportate în această fază – erori de typing, caractere invalide, șiruri neterminate, numere în format invalid etc. Această fază presupune scrierea unei gramatici ANTLR specifică limbajului și este dependentă de limbaj și independentă de arhitectura target. Trebuie menționat că limbajul COOL permite comentarii îmbricate (spre deosebire de C/C++ de exemplu care nu permite acest lucru) și pentru a parsă corect aceste comentarii trebuie folosite reguli recursive (ANTLR nu permite reguli recursive stanga deoarece folosește un praser LL(k) ). Tot în faza lexicală se construiește tabela de simboluri, precum și tabelele de constante. Compilatorul folosește 3 tabele separate : întregi, șiruri de caractere și identificatori. Această alegere este motivată de faptul că trebuie generat cod diferit pentru cele 3 cazuri (obiectele sunt reprezentate diferit în memorie)

Analiza sintactică presupune transformarea fluxului de atomi lexicali într-un arbore AST corespunzător programului. Acest arbore conține elementele sintactice corespunzătoare programului. În plus arborele poate conține noduri virtuale folosite mai departe în analiză semantică. Unii atomi din fluxul lexical nu se regăsesc în arbore (de exemplu “;” sau cuvântul cheie “in” din construcția let). În această fază sunt raportate erorile sintactice – declararea unui identificator fără tip, else fără if etc. Analiza sintactică este realizată cu ajutorul unei gramatici de tip tree (tree-grammar). Aceasta este dependentă de limbaj și independentă de mașină. În general se practică scrierea unei singure gramatici care combină analiza lexicală cu cea sintactică.

Analiza semantică presupune adnotarea arborelui AST generat de analiza sintactică cu informații de tip. Tot în această fază se fac unele verificări asupra structurii programului. Limbajul COOL impune verificări de tip asupra tuturor identificatorilor adăugați în tabela de simboluri. Datorită specificațiilor limbajului sunt necesare verificări de tip asupra tuturor expresiilor - în COOL o construcție de genul :

```
1 + if a > b then 2 else c
```

este perfect validă, cu toate că de exemplu c ar putea fi de un tip care nu poate fi adunat cu 1 (presupunând că 1 este tratat în context ca un Int, c nu extinde clasa Int). În acest caz deși la runtime se poate alege varianta corespunzătoare a > b și rezultatul expresiei să fie 3, un type checker trebuie să respingă această expresie datorită incompatibilității (construcția if nu este conformă cu tipul rezultatului – pentru mai multe detalii despre relația de conformitate se poate consulta [4]). În cadrul analizei semantice se face de asemenea verificarea structurii claselor care compun programul, precum și alte verificări specifice – de exemplu existența unei clase main și a metodei main corespunzătoare – specific COOL.

Generarea codului intermediar constă în traversarea arborelui AST și generarea de cod pentru fiecare

dintre subexpresiile asociate cu nodurile arborelui. Codul rezultat se numeste cod intermediar si foloseste de obicei instructiuni ale arhitecturii target a compilatorului si pseudoinstructiuni. Codul rezultat foloseste un fisier de registri abstracti (pot fi orcati). Alocarea de registrii se face intr-o faza de optimizare in pasul 5. Pentru generarea codului se asociaza <perechi sablon , cost sablon>, unde costul este exprimat in numar de instructiuni. In general compilatoarele mai avansate folosesc o metrica asociata sabloanelor care tine cont si de hazardurile structurale (RAW) cat si de arhitectura procesorului (posibilitati de paralelizare a instructiunilor la runtime, etc.). Cu setul de sabloane si costurile asociate se incearca traversarea arborelui astfel incat costul total al generarii de cod sa fie minim. Daca optimizarea este globala vorbim de un algoritm de programare dinamica, cum (un exemplu este Iburg – pentru gramatici ale limbajului C). Compilatorul COOL foloseste o abordare mai simpla, analizand optimul local (abordare de tip greedy) pentru a optimiza evaluarea seriala a unor expresii.

Datorita mediului de runtime COOL, la evaluarea expresiilor sunt generate obiecte noi, acestea fiind temporar utilizate (in expresii care contin apeluri de functii de exemplu), avand un life-time scurt. Deoarece tot codul COOL este managed code, compilatorul poate cere invocarea GC on demand, setand mai multi parametrii ai acestuia (el este invocat automat si de catre mediul de runtime). In plus compilatorul COOL optimizeaza codul prin reutilizarea unor anumite obiecte in cadrul evaluarii expresiilor prin schimbarea adreselor la care pointeaza referintele operatorilor implicati in evaluarea expresiei. Rezultatul fazei de generare a codului este un program (text) in limbaj intermediar + cod MIPS. Dintre sectiunile corespunzatoare unui program COOL in aceasta faza sunt complete :

- tabela de constante string
- tabela de constante int
- tabela de metode pentru fiecare obiect (Object Dispatch Table)
- tabela de obiecte (Object Table)
- codul pentru clasele de baza folosite in program(Object, String, etc.)

Faza de optimizare este poate cea mai importanta din cadrul unui compilator. In acest domeniu se fac cercetari la ora actuala, si desi exista multi algoritmi de optimizare, in cazul limbajelor complexe (mai ales cele orientate obiect ex Java, C#) exista multe imbunatatiri care s-ar putea aduce. Compilatorul COOL contine o faza de optimizare relativ simplista (comparativ cu compilatoarele comerciale). Principala simplificare este lipsa analizei interprocedurale – adica optimizarea se realizeaza la nivelul metodelor (nu la nivelul intregului cod), netinand seama de informatii externe acestora (de ex. ce se intampla cu variabilele locale ale metodei la un apel al unei alte metode). In cadrul unei metode optimizarile se fac folosind algoritmul Kildall. Acesta presupune o analiza iterativa de flux a codului, functiile si ecuatiile de flux fiind specifice fiecarui tip de optimizare. Compilatorul COOL calculeaza MFP pentru fluxul asociat codului unei metode aplicand in ordine urmatoarele optimizari :

- propagarea copiilor (Copy Propagation) presupune inlocuirea folosirii unei variabile cu (singura) definitie a ei care ajunge in acel loc, daca acea definitie este o instructiune de atribuire(copiere)
- propagarea constantelor (Constant Folding) presupune evaluarea expresiilor care contin constante si inlocuirea acestora in textul sursa la momentul compilarii
- eliminarea codului care nu poate fi executat (Unreachable Code) – caz particular al Dead Code Elimination – presupune validarea tuturor cailor de executie posibile si eliminarea celor care nu se pot executa niciodata indiferent de intrari
- mutarea codului invariant (Loop Code Motion) – presupune mutarea codului invariant dintr-o bucla intr-o sectiune speciala in afara buclei (preheader) pentru a reduce numarul de instructiuni/ bucla

Dupa aplicarea optimizarii tuturor metodelor, se face alocarea registrelor. Compilatorul COOL contine un alocator de registre simplu, care este construit conform specificatiilor COOL si nu contine nici o secventa de optimizare, alegand registrii in mod greedy. Aceasta abordare este rapida si pentru majoritatea programelor este si suficienta datorita numarului mare de registre al arhitecturii MIPS.

## 5. Concluzii

Compilatorul COOL ofera o viteza de compilare comparabila cu cea a compilatoarelor comerciale. Optimizarile sunt simple si sunt eficiente doar in cadrul programelor mici, in care fluxul interprocedural nu este semnificativ. In cadrul proiectelor mari lipsa analizei inter-procedurale duce la minimizarea eficientei optimizarilor locale per metoda.

Organizarea obiectelor si imbunatatirea layoutului in memorie, pentru reducerea dimensiunii codului si transformarea unor apeluri de metoda in salturi locale (posibilitatea de inlining din compilator a unor metode) este o alta imbunatatire majora care ar putea fi adusa compilatorului.

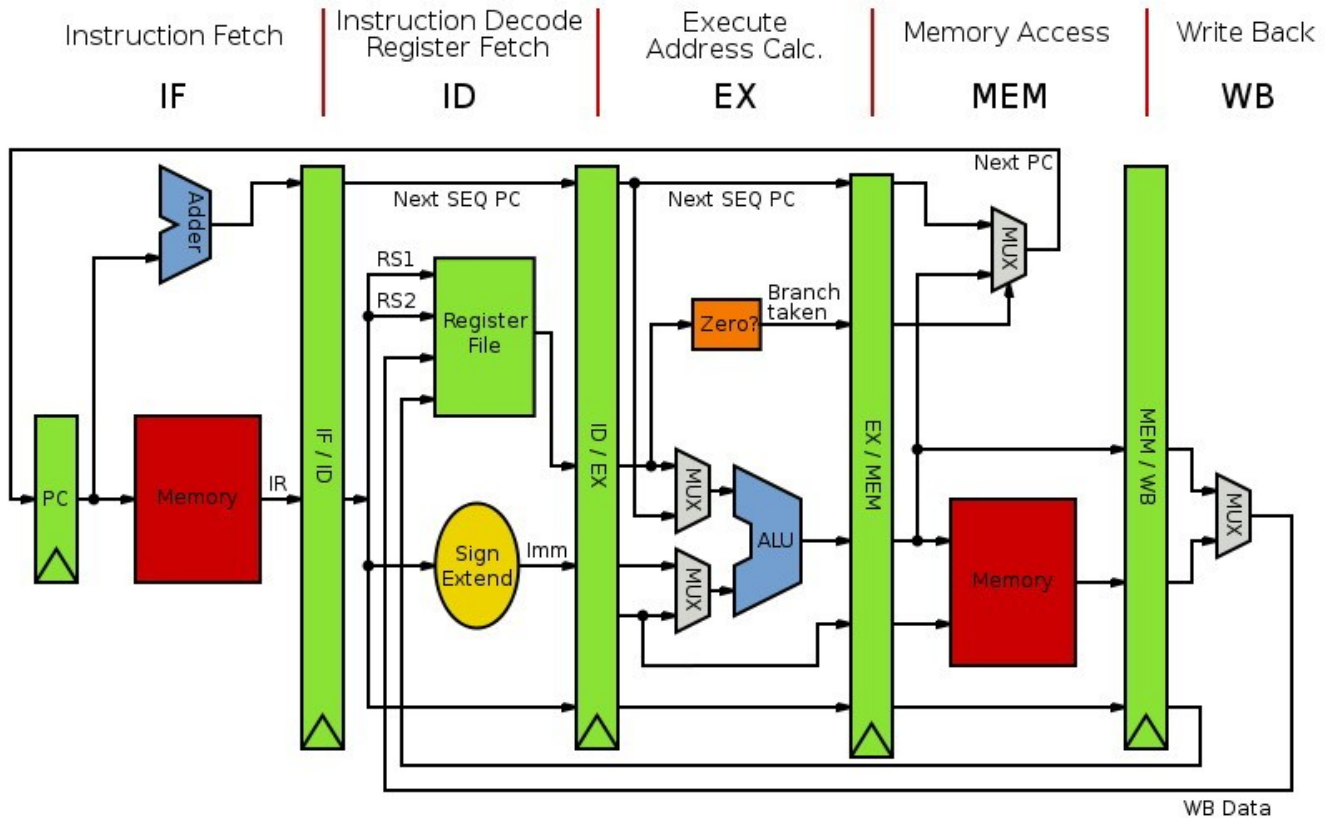
In al trilea rand includerea suportului pentru generarea instructiunilor din seturile extinse MIPS ar extinde functionalitatea compilatorului si posibilitatile de utilizare in aplicatii.

Tablelul 1. Registrele MIPS

| Nu<br>me<br>regi<br>stru | Numar<br>registru | Utilizare                                    | Salvat de<br>apelat |
|--------------------------|-------------------|--|---------------------|
| \$zero                   | \$0               | legat la 0                                   | N/A                 |
| \$at                     | \$1               | temporar                                     | nu                  |
| \$v0<br>-<br>v1          | \$2-\$3           | evaluare expresii returnare rezultat functie | nu                  |
| \$a0<br>-<br>\$a3        | \$4-\$7           | argumentele functiei                         | nu                  |
| \$t0-<br>\$t7            | \$8-\$15          | registrii temporari                          | nu                  |
| \$s0<br>-<br>\$s7        | \$16-\$23         | registrii temporari salvati                  | da                  |
| \$t8-<br>\$t9            | \$24-\$25         | registrii temporari                          | nu                  |
| \$k0<br>-<br>\$k1        | \$26-\$27         | rezervati pentru kernel                      | nu                  |
| \$gp                     | \$28              | global pointer                               | da                  |
| \$sp                     | \$29              | stack pointer                                | da                  |
| \$fp                     | \$30              | frame pointer                                | da                  |
| \$ra                     | \$31              | adresa de return                             | N/A                 |

Figura 1. Arhitectura procesorului MIPS

Figura 2. Sintaxa COOL



```

| isvoid expr
| expr + expr
| expr - expr
| expr * expr
| expr / expr
| ~expr
| expr < expr
| expr <= expr
| expr = expr
| not expr
| (expr)
| ID
| integer
| string
| true
| false

```

## Bibliografie

1. Tim Langens, Jef Neefs, Elio Struyf, Luc Verstrepen - *MIPS*, 2000
2. MIPS Technologies, Inc. - *MIPS32™ Architecture For Programmers Volume II: The MIPS32™ Instruction Set*, Nr. MD00086, r. 0.95, 12 martie, 2001
3. MIPS Technologies, Inc. - *MIPS32 4K™ Processor Core Family*, Nr. MD00016, r. 01.17, 25 septembrie, 2002
4. Alex Aiken, John Boyland - *CoolAid: The Cool 2009 Reference Manual*, 2009
5. Alex Aiken - *The Cool Runtime System*, 2002
6. R. Mark Volkmann - *ANTLR 3*, 2008 Object Computing, Inc. (OCI)

**Danciu Alin - Mircea**  
**Universitatea Politehnica Bucuresti**  
**Facultatea de Automatica si Calculatoare**  
**Sectia Calculatoare, C3**  
alin.danciu@gmail.com

## Aplicație geometrică în Matlab

**Autori:** Draguț Ana Maria, Florea Cristian, Lunceanu Simina, **Universitatea „Lucian Blaga”** din Sibiu  
**Prof. Univ. Dr. Eugen Drăghici, Universitatea „Lucian Blaga”** din Sibiu, Departam. de Matematică

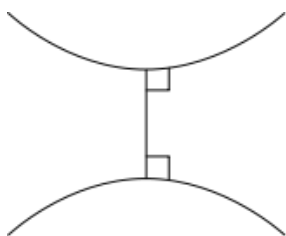
### 1. Introducere

*" În domeniul științei o revoluție este o prefacere adâncă,  
o temeinică transformare pricinuită de un fapt esențial nou,  
de o lege care generează noi și vaste progrese,  
de o descoperire deschizătoare de drumuri noi  
ce lasă să se întrevadă perspective largi, bogate,  
care adesea până atunci nu puteau fi bănuite."  
Simion Stoilov*

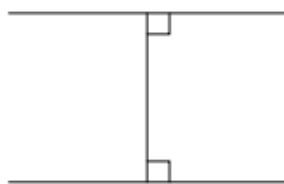
Pășim acum pe un tărâm necunoscut de unii, neuzit de alții, ignorat de cei rămași...dar explorat de noi...trei amici...ghidați de un cunoscător al locurilor...dar nu numai...un cercetător pasionat de aceste tărâmurii...De aceea dorim să vă alăturati și dumneavoastră drumeției: nu contează vârsta, ocupația, trecutul, familia, naționalitatea, statutul social...nu ținem cont de nici unul din aceste criterii...sunteți bineveniți să ne urmați...și poate, cine știe?...poate vă veți surprinde absorbiți de cele ce urmează...poate facem și noi o revoluție precum amintea mai sus domnul Simion...de ce nu?

Teorema a fost descoperită în urma cercetărilor, încercărilor și perseverenței domnului Steiner făcute în faimoasa epocă de aur a geometriei, care a condus la progrese ca geometria proiectivă și geometriile neeuclidiene. Geometria neeuclidiană este o ramură a geometriei care diferă de geometria euclidiană printr-o altă axiomă de paralelism.

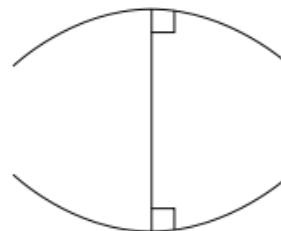
În geometria neeuclidiană *hiperbolică* numită de obicei geometria lui Lobacevski, printr-un punct dat putem să ducem două paralele la o dreaptă dată. În geometria neeuclidiană *eliptică* nu există drepte paralele.



Hyperbolic



Euclidean



Elliptic

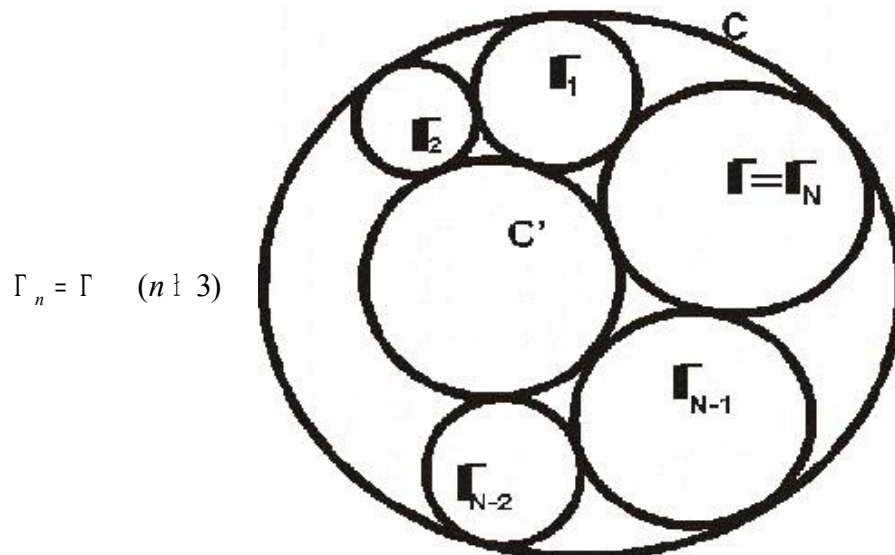
S-a demonstrat că geometriile neeuclidiene sunt necontradictorii și s-au construit și modele în spațiul euclidian pe care ele le verifică. Crearea acestor geometrii neeuclidiene a dovedit faptul că în mod logic sunt posibile mai multe sisteme geometrice.

Geometria neeuclidiană este folosită pentru formularea teoriei generalizate a relativității.

## 2. Formularea Problemei

Fie  $C$  și  $C'$  două cercuri în plan, cu  $C'$  în interiorul lui  $C$ . Ducem un cerc  $\Gamma$  tangent la amândouă cercurile  $C$  și  $C'$ . Apoi ducem cercul  $\Gamma_1$ , tangent la  $C$  și  $C'$  dar și la  $\Gamma$ . Continuăm această construcție de  $n$  ori, ultimul cerc  $\Gamma_n$  fiind tangent la  $C$ ,  $C'$  și de asemenea la  $\Gamma_{n-1}$ . Dacă se întâmplă ca:

Fig.1



$$\Gamma_n = \Gamma \quad (n \geq 3)$$

(1)

după o revoluție în jurul cercului  $C'$ , atunci aceasta se va întâmpla indiferent de cercul inițial  $\Gamma$  care a fost ales.

Încercând să răspundem la întrebarea dacă teorema noastră poate fi redusă la cazul cercurilor concentrice  $C$  și  $C'$ , ar trebui să transformăm fig.1 astfel încât cercurile  $C$  și  $C'$  să devină concentrice. Ce fel de transformări ar trebui folosite în acest scop?

Vom încerca să scriem o demonstrație a teoremei în felul următor: dat fiind faptul că figura (1) este compusă din cercuri, trebuie să folosim aplicații punctuale care transformă cercuri în cercuri. Astfel de transformări sunt liniare fracționare sau transformări Möbius, transformări de forma:

$$(2) \quad w = (Az + B)/(Cz + D) \quad (\text{unde } A, B, C, D \text{ sunt constante complexe}).$$

Acestea sunt tratate în orice fel de introducere la teoria funcțiilor de variabilă complexă  $z = x + iy$ . Să notăm  $D$  și  $D'$  discurile circulare mărginite de  $C$  și  $C'$ .

Presupunem că  $C$  și  $C'$  nu sunt concentrice și, fără a micșora generalitatea că  $D$  este discul unitate:

$$(3) \quad D: |z| \leq 1$$

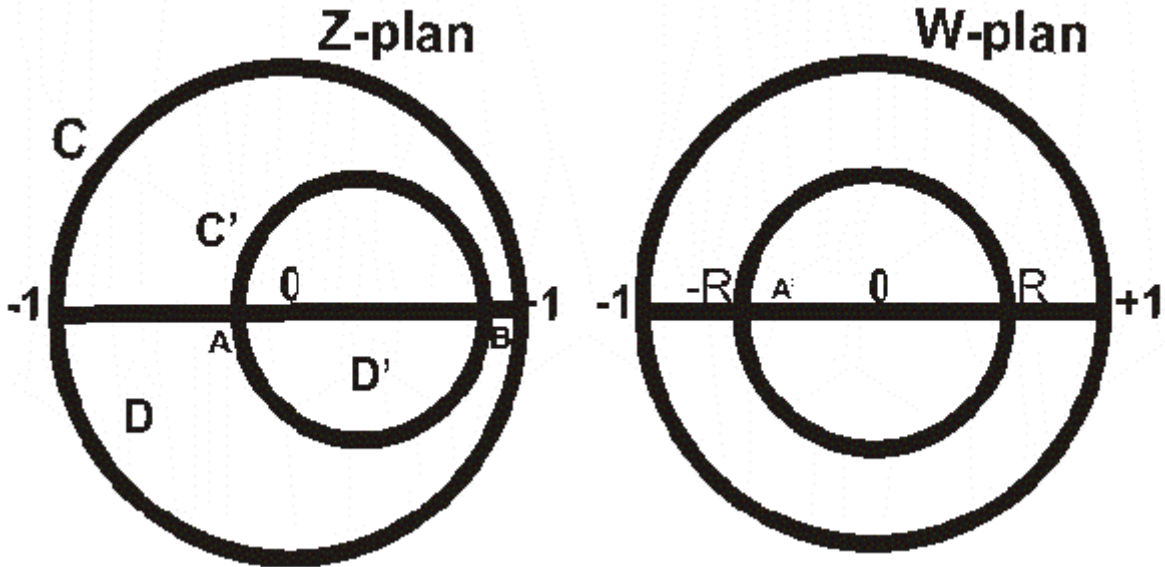
Mai mult, printr-o rotație convenabilă  $w = e^{i\alpha} z$ , putem presupune că centrul lui  $D'$  este pe segmentul  $0 < x < 1$ . În sfârșit, fie intersecția:

$$(4) \quad [\alpha \beta] = D' \cap \{y = 0\}, (\alpha + \beta > 0),$$

un diametru al lui  $D'$ , ca în figura următoare:

Fig.2





Există o aplicație (2) care transformă D pe el însuși și astfel încât D' să fie aplicat în cercul (concentric):

$$(5) \quad |z| \leq r, (0 < r < 1) ?$$

Răspunsul afirmativ se obține după cum urmează. Observăm în primul rând că aplicația:

$$(6) \quad w = (z + a)/(az + 1) \quad (-1 < a < 1)$$

aplică pe D pe el însuși, astfel încât  $z = -1, 0, 1$ , se transformă în  $w = -1, a, 1$ . Vrem să arătăm că putem găsi  $a$  astfel încât  $\alpha$  și  $\beta$  să fie transformate în punctele  $-r$  și  $r$ . Folosind (6), aceasta revine la ecuația:

$$(\alpha + a)/(a\alpha + 1) = (-\beta + a)/(a\beta + 1),$$

care este echivalentă cu ecuația de gradul al-doilea:

$$(7) \quad a^2 + 2a(1 + \alpha\beta)/(\alpha + \beta) + 1 = 0$$

Această ecuație are rădăcini reale, pentru că discriminantul ei

$$\left( \frac{1 + \alpha\beta}{\alpha + \beta} \right)^2 - 1 = \frac{1}{(\alpha + \beta)^2} (1 - \alpha^2 - \beta^2 + \alpha^2\beta^2) = \frac{1}{(\alpha + \beta)^2} (1 - \alpha^2)(1 - \beta^2) > 0,$$

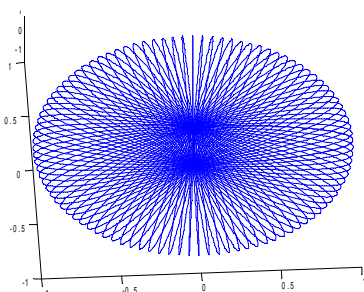
este pozitiv. Cum produsul celor două rădăcini ale lui (7) este 1, tragem concluzia că există un singur  $a$  astfel încât

$$(8) \quad -1 < a < 0,$$

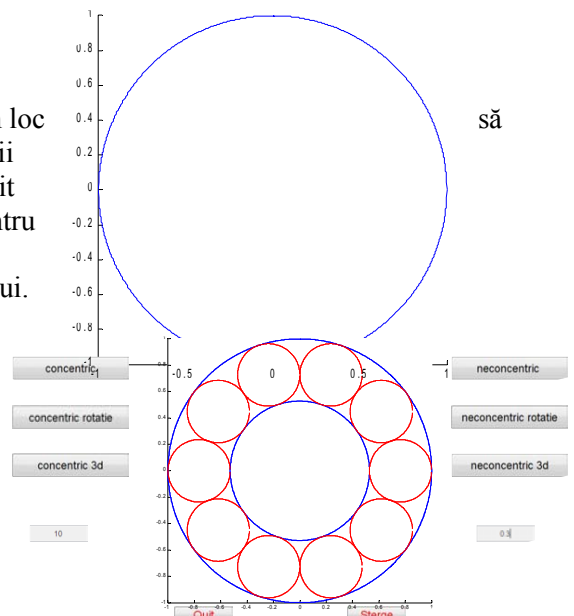
ultima inegalitate (8), fiind o consecință a faptului că cele două rădăcini ale lui (7) sunt negative. Cu această valoare a lui  $a$  se vede că (6) aplică pe D' în discul (5) și teorema este demonstrată.

## 2.1 Realizarea programului în Matlab

Idea realizării programului a fost una relativ simplă. În loc creăm cazul cu cercuri neconcentrice și cu ajutorul transformării omografice să ajungem la cazul cercurilor concentrice, am privit problema invers, datorită simplității calculelor matematice. Pentru cazurile concentrice am realizat 5 programe: primul program desenează un cerc și are ca date de intrare raza și centrul cercului.



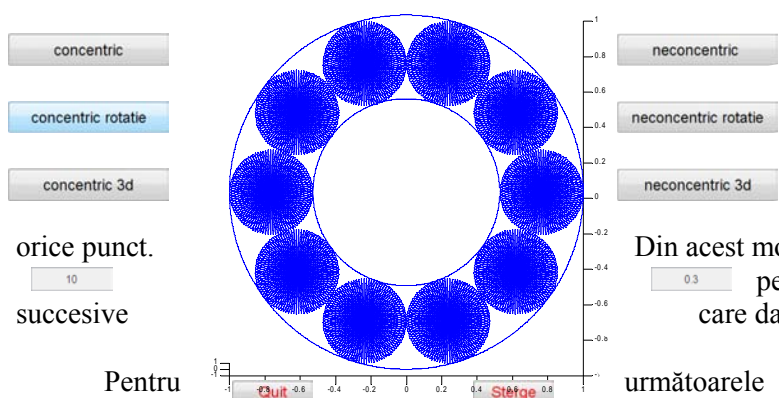
Cel de-al doilea program desenează o sferă, cu ajutorul coordonatelor polare și a funcției plot3



(program necesar pentru desenarea 3D ). Al treilea program desenează un cerc în care ia alt cerc concentric, iar apoi construiește cercuri tangente celor două și cercului precedent desenat. O observație importantă este aceea că între raza cercului mare, raza cercului mic și numărul de cercuri construite există o relație care poate fi dedusă datorită faptului că între oricare două cercuri construite avem același unghi (din acest motiv noi introducem doar numărul de cercuri care vrem să fie construite, raza cercului mare fiind considerată pentru simplificarea calculelor egală cu 1). Inițial am desenat primul cerc tangent, știind faptul că punctele de tangență și toate centrele sunt coliniare formând astfel segmente de anumite proporții.

După ce primul cerc a fost desenat pentru următoarele cercuri aveam nevoie doar de centrele lor, raza fiind aceeași cu a primului cerc. Centrul fiecărui cerc a fost determinat astfel: centrul cercului inițial a fost transformat într-un număr complex unde partea reală reprezenta punctul de pe abscisă, iar partea imaginară reprezenta punctul de pe ordonată. Acest număr complex a fost înmulțit cu  $e^{ika}$ , unde  $k$  este numărul cercurilor pe care vrem să-l desenăm, iar  $a$  unghiul constant dintre oricare două cercuri, și astfel s-a realizat o rotație. Având centrul și raza, cu ajutorul primului program am desenat toate cercurile.

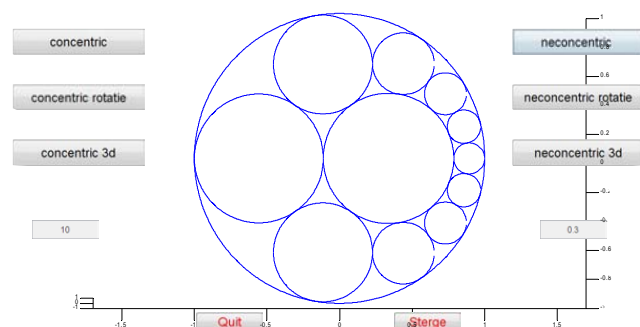
Al patrulea program desenează același lucru, însă în spațiu, folosindu-se de programul al doilea.



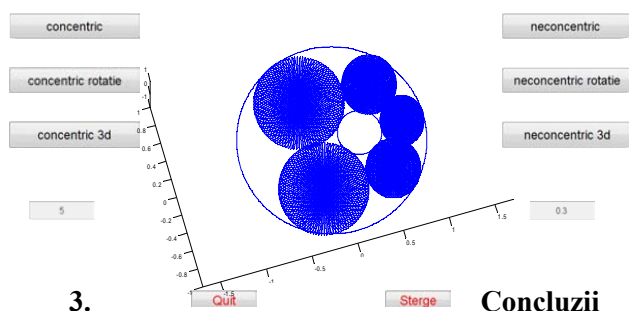
orice punct.  
10  
succesive

Pentru aplicații ne-am folosit de funcția omografică inversă, implementată sub forma unui program. Aceasta avea nevoie de un alt parametru de intrare care să indice cu cât este deplasat cercul în interior. Astfel fiecare program în care am utilizat cercuri concentrice „a fost trecut” prin funcția omografică inversă, rezultând desene similare, dar în care cercul din interior este deplasat.

Al cincilea program desenează aceste cercuri în mișcare, folosindu-se exact de teorema lui Steiner, care spunea că dacă cercurile pot fi construite începând dintr-un anumit punct, atunci ele pot fi construite din Din acest motiv am făcut ca punctul să se plimbe „fin” pe cercul mare, rezultând construcții care dau senzația de mișcare.



Totuși aceste treceri au ridicat anumite probleme, ca de exemplu: un punct de pe cerc este transformat într-un punct de pe același cerc (despre centru nu putem spune nimic, deci trebuie să construim cercuri având doar puncte de pe el); numerele complexe pierd noțiunea de centru (în cazul în care scriem un cerc sub forma  $|z|=r$ ); în unele calcule matematice a trebuit, prin anumite condiții, să evităm împărțirea la 0.



3.

Concluzii

Deocamdata, din câte știm, inginerii și proiectanții nu s-au folosit de această teoremă pentru a realiza ceva în practică, deci nu putem vorbi de



## Aplicație geometrică în Matlab

aplicații ale teoremei lui Steiner în aceste domenii, deoarece, așa cum am precizat și mai sus, această teoremă face parte din multitudinea de teoreme matematice, riguros demonstrate, dar rămase doar la stadiul de teorie. Însă nu putem spune același lucru și în cazul artiștilor, care de această dată au fost mai inspirați și chiar dacă poate nu s-au folosit de bazele teoremei lui Steiner, noi am găsit similitudini, între operele acestora și teorema lui Steiner. Unul dintre acești artiști este Alphonse Mucha, unul dintre cei care au conturat conceptul Art Nouveau, punându-și în valoare talentul într-o minunată operă de artă al cărei nume a rămas mult timp în memoria admiratorilor: Dancel.

Totuși, noi ne-am gândit că această construcție ar putea fi baza construirii unui rulment cu axul deplasat, caruia i s-ar putea găsi o întrebuințare ingenioasă și folositoare.

Acum, ajunși la sfârșitul călătoriei noastre, după ce am vizitat atâtea domenii precum: matematica, arta, ingineria, și am cercetat ideile altor matematicieni, e timpul să dăm frâu liber imaginației și să inventăm noi ceva util generației noastre și celor ce vor veni după noi...

### Bibliografie

1. Isaac J. Schoenberg, Priveliști matematice, Editura Tehnică, București, 1989
2. <http://www.suberic.net/%7Edmm/graphics/crescent/crescent.html>
3. [http://ro.wikipedia.org/wiki/Geometrii\\_neuclidiene](http://ro.wikipedia.org/wiki/Geometrii_neuclidiene)
4. [epsilon.ro/index2.php?option=com\\_docman&task=doc\\_view&gid=53&Itemid=18](http://epsilon.ro/index2.php?option=com_docman&task=doc_view&gid=53&Itemid=18) -

**Drăguț Ana Maria, Florea Cristian, Lunceanu Simina**

**Universitatea „Lucian Blaga” din Sibiu**

**Facultatea de Științe**

**Specializarea Matematică-Informatică (program de licență)**

[anamariadragut@ymail.com](mailto:anamariadragut@ymail.com), [florea\\_cristy2006@yahoo.com](mailto:florea_cristy2006@yahoo.com), [shiminutza@yahoo.com](mailto:shiminutza@yahoo.com)

### Anexa

functia cerc

```
function [x]=cerc(r,a,b)
```

```
axis square
```

```
x=a-r:0.001:a+r;
```

```
y=sqrt(r^2-(x-a).^2)+b;
```

```
hold on
```

```
plot(x,y)
```

```
%x=a-r:0.1:a+r;
```

```
y=-sqrt(r^2-(x-a).^2)+b;
```

```
plot(x,y)
```

```
hold off
```

functia care realizeaza transformarea

```
function [x]=transformare( r, a, b, c)
```

```
z=a+i*b;
```

```

za=(z-r-c)/(1-(z-r)*c);
zb=(z+r-c)/(1-(z+r)*c);
zc=(z+i*r-c)/(1-(z+i*r)*c);
if (imag(za)-imag(zb))==0
    m2=0;
else
m2=(real(zb)-real(za))/(imag(za)-imag(zb));
end

if (imag(zc)-imag(zb))==0
    m3=0;
else
m3=(real(zb)-real(zc))/(imag(zc)-imag(zb));
end

AA=[ -m2, 1; -m3,1];
BB=[ (-1)*m2*real((za+zb)/2)+imag((za+zb)/2); (-1)*m3*real((zc+zb)/2)+imag((zc+zb)/2)];
XX=AA^(-1)*BB;
zz=XX(1,1)+i*XX(2,1);
r1=abs(za-zz);
cerc(r1,XX(1,1),XX(2,1));

```

functia sfera

```

function[x]=sfera(r,a,b)
hold on
i=0;
j=0;
k=0;
p1=linspace(0,pi,100);
p2=linspace(0,2*pi,100);
[n,m]=size(p1);
[n1,m1]=size(p2);
for p3=1:m
for p4=1:m
i=i+1;
j=j+1;
k=k+1;
x(i)=r*cos(p2(p3))*sin(p1(p4))+a;
y(j)=r*sin(p1(p4))*sin(p2(p3))+b;
z(k)=cos(p1(p4));
end
end
plot3(x,y,z)

```

hold off

functia cercuri concentrice

```

function [x]=cercuri(n)
if n<3
    disp(' numar prea mic de cercuri dorite')
    return
end

```

## Aplicație geometrică în Matlab

```
t=sin(pi/n);
r1=(1-t)/(1+t);
r=1;
a=0;
b=0;
a1=0;
b1=0;
p1=1;
p2=0;
hold on
cerc(r,a,b);
cerc(r1,a1,b1);

if((p1-a)^2+(p2-b)^2~r^2)
    disp('punctul p nu este pe cerc')
    return
end

r3=(sqrt((p1-a1)^2+(p2-b1)^2)-r1)/2;
m=r3/(r3+r1) ;
a3=(p1+m*a1)/(m+1);
b3=(p2+m*b1)/(m+1);

z=a3+i*b3;
l=2*pi/n;

for k=0:n-1
    p=i*k*l;
    s=z*exp(p);
    cerc1(r3,real(s),imag(s));
end

functia cercuri concentrice miscare

function [x]=cercuri(n)
contor=1;
numar=0;
p1=1;
p2=0;
if n<2
    disp(' numar prea mic de cercuri dorite')
    return
end

t=sin(pi/n);
r1=(1-t)/(1+t);
r=1;
a=0;
b=0;
a1=0;
b1=0;
```

```

while numar<3
    if p1<-1
        contor=0;
        numar=numar+1;
    end
    if p1>=1
        contor=1;
    end
    cla;

    cerc(r,a,b);
    cerc(r1,a1,b1);
    r3=(sqrt((p1-a1)^2+(p2-b1)^2)-r1)/2;
    m=r3/(r3+r1) ;
    a3=(p1+m*a1)/(m+1);
    b3=(p2+m*b1)/(m+1);

    cerc1(r3,a3,b3);

    z=a3+i*b3;
    l=2*pi/n;
    o=1;
    hold on;
    for k=0:n-1
        p=i*k*l;
        s=z*exp(p);
        cerc1(r3,real(s),imag(s));
        axis('square','equal','on');
    end
    hold off;
    pause(0.01);

    if contor==1
        p1=p1-0.05;
        p2=sqrt(1-p1^2);
    else
        p1=p1+0.05;
        p2=-sqrt(1-p1^2);
    end

    if p1== -1
        p1=1;
    end
end

cercuri concentrice 3d

function [x]=cercuriconcentrice3(n)

if n<3
    disp(' numar prea mic de cercuri dorite')
    return
end

```

## Aplicație geometrică în Matlab

```
t=sin(pi/n);
r1=(1-t)/(1+t);
r=1;
a=0;
b=0;
a1=0;
b1=0;
p1=1;
p2=0;
hold on
cerc(r,a,b);
cerc(r1,a1,b1);

r3=(sqrt((p1-a1)^2+(p2-b1)^2)-r1)/2;
m=r3/(r3+r1) ;
a3=(p1+m*a1)/(m+1);
b3=(p2+m*b1)/(m+1);

z=a3+i*b3;
l=2*pi/n;

for k=0:n-1
    p=i*k*l;
    s=z*exp(p);
    sfera(r3,real(s),imag(s));
end
```

cercuri neconcentrice

```
function [x]=cercuritest(n,l)
contor=1;
numar=0;
p1=1;
p2=0;
if n<3
    disp(' numar prea mic de cercuri dorite')
    return
end
%while numar<3
    if p1<-1
        contor=0;
        numar=numar+1;
    end
    if p1>=1
        contor=1
    end
end
cla;

t=sin(pi/n);
r1=(1-t)/(1+t);
l1=-r1+l;
l2=r1+l;
```

```
q1=(-1)*(1+i1*12)/(11+i2)+sqrt(((1+i1*12)/(11+i2))^2-1));
q2=(-1)*(1+i1*12)/(11+i2)-sqrt(((1+i1*12)/(11+i2))^2-1));
```

```
if(q1<0)
    if(q1>-1)
        q3=q1;
    else
        q3=q2;
    end
else
    q3=q2;
end
```

```
r=1;
a=0;
b=0;
a1=0;
b1=0;
```

```
cerc(r,a,b);
transformare(r1,a1,b1,q3);
if((p1-a)^2+(p2-b)^2~r^2)
    %disp('punctul p nu este pe cerc')
    %return
end
```

```
r3=(sqrt((p1-a1)^2+(p2-b1)^2)-r1)/2;
m=r3/(r3+r1) ;
a3=(p1+m*a1)/(m+1);
b3=(p2+m*b1)/(m+1);
```

```
z=a3+i*b3;
l=2*pi/n;
o=1;
hold on;
for k=0:n-1
    p=i*k*l;
    s=z*exp(p);
    transformare(r3,real(s),imag(s),q3);
    axis('square','equal','on');
end
hold off;
pause(0.01);
```

```
if contor==1
    p1=p1-0.01;
    p2=sqrt(1-p1^2);
else
    p1=p1+0.01;
    p2=-sqrt(1-p1^2);
end
```



## Aplicație geometrică în Matlab

```
if p1== -1
    p1=1;
end
```

cercuri neconcentrice miscare

```
function [x]=cercuri(n,l)
```

```
contor=1;
numar=0;
p1=1;
p2=0;
t=sin(pi/n);
r1=(1-t)/(1+t);
r=1;
a=0;
b=0;
a1=0;
b1=0;
l1=-r1+l;
l2=r1+l;
```

```
q1=(-1)*(1+l1*l2)/(l1+l2)+sqrt(((1+l1*l2)/(l1+l2))^2-1);
q2=(-1)*(1+l1*l2)/(l1+l2)-sqrt(((1+l1*l2)/(l1+l2))^2-1);
```

```
if(q1<0)
    if(q1>-1)
        q3=q1;
    else
        q3=q2;
    end
else
    q3=q2;
end
if n<3
    disp(' numar prea mic de cercuri dorite')
    return
end
while numar<2
    if p1<-1
        contor=0;
        numar=numar+1;
    end
    if p1>=1
        contor=1
    end
end
cla;
```

```
cerc(r,a,b);
transformare(r1,a1,b1,q3);
r3=(sqrt((p1-a1)^2+(p2-b1)^2)-r1)/2;
m=r3/(r3+r1) ;
a3=(p1+m*a1)/(m+1);
b3=(p2+m*b1)/(m+1);
```

```
z=a3+i*b3;
l=2*pi/n;
o=1;
hold on;
    for k=0:n-1
        p=i*k*l;
        s=z*exp(p);

        transformare(r3,real(s),imag(s),q3);
        axis('square','equal','on');
        end
hold off;
pause(0.01);
```

```
if contor==1
    p1=p1-0.05;
    p2=sqrt(1-p1^2);
else
    p1=p1+0.05;
    p2=-sqrt(1-p1^2);
end
```

```
if p1==-1
    p1=1;
end
end
```

cercuri neconcentrice 3d

```
function [x]=cercuriconcentrice3(n)

if n<3
    disp(' numar prea mic de cercuri dorite')
    return
end

t=sin(pi/n);
r1=(1-t)/(1+t);
r=1;
a=0;
b=0;
a1=0;
b1=0;
```

## Aplicație geometrică în Matlab

```
p1=1;
p2=0;
hold on
cerc(r,a,b);
cerc(r1,a1,b1);

r3=(sqrt((p1-a1)^2+(p2-b1)^2)-r1)/2;
m=r3/(r3+r1) ;
a3=(p1+m*a1)/(m+1);
b3=(p2+m*b1)/(m+1);

z=a3+i*b3;
l=2*pi/n;

for k=0:n-1
    p=i*k*l;
    s=z*exp(p);
    sfera(r3,real(s),imag(s));
end

interfata

abutton=uicontrol('style','pushbutton',...
    'string','concentric', ...
    'fontsize',18, ...
    'Units','normalized',...
    'Position',[0.02 0.8 0.2 0.07],...
    'callback', 'contor=1;');

exbutton=uicontrol('style','pushbutton',...
    'string','Sterge', ...
    'fontsize',18, ...
    'Units','normalized',...
    'foregroundColor','r','position',[0.6 0.05 0.1 0.05], ...
    'callback','cla');

quitbutton=uicontrol('style','pushbutton',...
    'string','Quit', ...
    'fontsize',18, ...
    'foregroundColor','r','Units','normalized',...
    'position',[0.3 0.05 0.1 0.05], ...
    'callback','stop=1;close;');

bbutton=uicontrol('style','pushbutton',...
    'string','concentric rotatie', ...
    'fontsize',18, ...
    'Units','normalized',...
    'Position',[0.02 0.65 0.2 0.07],...
    'callback', 'contor=2;');

cbutton=uicontrol('style','pushbutton',...
    'string','concentric 3d', ...
    'fontsize',18, ...
    'Units','normalized',...
```

```

        'Position',[0.02 0.5 0.2 0.07],...
        'callback','contor=3;');

dbutton=icontrol('style','pushbutton',...
    'string','neconcentric ', ...
    'fontsize',18, ...
    'Units','normalized',...
    'Position',[0.78 0.8 0.2 0.07],...
    'callback','contor=4;');
ebutton=icontrol('style','pushbutton',...
    'string','neconcentric rotatie', ...
    'fontsize',18, ...
    'Units','normalized',...
    'Position',[0.78 0.65 0.2 0.07],...
    'callback','contor=5;');
cbutton=icontrol('style','pushbutton',...
    'string','neconcentric 3d', ...
    'fontsize',18, ...
    'Units','normalized',...
    'Position',[0.78 0.5 0.2 0.07],...
    'callback','contor=6;');

xrange=icontrol('style','edit',...
    'string','nr cercuri', ...
    'fontsize',12, ...
    'units','normalized',...
    'position',[0.05,0.3,0.1,0.05], ...
    'callback','contor=1;...'
);
yrange=icontrol('style','edit',...
    'string','deplasare', ...
    'fontsize',12, ...
    'units','normalized',...
    'position',[0.823,0.3,0.1,0.05], ...
    'callback','contor=1;...'
);

stop=0;
contor=0;

while (stop==0)
    drawnow;

    if contor==1
        contor=0;
        x=str2num(get(xrange,'string'));
        cercuriconcentrice(x);
    drawnow;
    end

    if contor==2
        contor=0;

```

## Aplicație geometrică în Matlab

```
    x=str2num(get(xrange,'string'));  
    cercuriconcentrice1(x);  
drawnow;  
end
```

```
if contor==3  
    contor=0;  
    x=str2num(get(xrange,'string'));  
    cercuriconcentrice3(x);  
drawnow;  
end
```

```
if contor==4  
    contor=0;  
    y=str2num(get(yrange,'string'));  
    x=str2num(get(xrange,'string'));  
    cercurineconcentrice(x,y);  
    drawnow;  
end
```

```
if contor==5  
    contor=0;  
    y=str2num(get(yrange,'string'));  
    x=str2num(get(xrange,'string'));  
    cercurineconcentrice1(x,y);  
end
```

```
if contor==6  
    contor=0;  
    y=str2num(get(yrange,'string'));  
    x=str2num(get(xrange,'string'));  
    cercurineconcentrice3(x,y);  
end  
end
```

## Platformă de e-learning pentru învățământ la distanță

**Autor:** Epure Ancuta-Raluca, **Universitatea „Lucian Blaga”** din Sibiu

**Coordonator:** Hunyadi Daniel, **Universitatea „Lucian Blaga”** din Sibiu, Informatică

*Abstract: Nowadays, computer became the object key in every kind of activities, including learning and valuation process. This e-Learning system is used more and more not only in schools, but in companies in trainings and it take a form of the distance education. A lot of universities have a platform for e-learning.*

*A platform for e-learning is in fact a web site; the access to information is online, the Internet becoming not just a search tool and information, but also a learning tool. This kind of site has a lot of advantages: hosts more information which can be accessed from everywhere using a computer connected to the internet, not require real-space, not require synchronization of the student, so that they can organize their learning according to their spare time, and evaluation is effective and interactive.*

*Keywords: Platform for e-learning, distance education, Internet, web site.*

### 1 Introducere

În sens restrâns, e-learning reprezintă un tip de educație la distanță, ca experiență planificată de predare-învățare organizată de o instituție ce furnizează mediat materiale într-o ordine secvențială și logică pentru a fi asimilate de studenți în maniera proprie. Mediarea se realizează prin noile tehnologii ale informației și comunicării - în special prin Internet. Internetul constituie atât mediul de distribuție al materialelor, cât și canalul de comunicare între actorii implicați. Funcțional deocamdată doar la nivelul învățământului superior și în educația adulților, sistemul de instruire prin Internet replică și adaptează componentele demersului didactic tradițional/ față în față: planificare, conținut specific și metodologie, interacțiune, suport și evaluare.

De asemenea, folosirea internă a unui sistem de e-Learning de către companiile cu o mare nevoie de instruire a personalului nu poate fi decât un câștig pentru aceștia și pentru societate în general și de acest lucru își vor da seama tot mai mulți manageri.

Perioada 2006-2010 este descrisă ca fiind de formare a **Comunității Digitale**. Învățământul bazat pe proiecte devine din ce în ce mai pregnant o formă „obișnuită” de promovare didactică în școli.

Pentru a forma noile generații astfel încât să fie capabile să facă față problematicei lumii contemporane, sunt necesare acele schimbări în paradigma învățării care favorizează trecerea de la învățarea disciplinară, atomizată, la cea orientată către dezvoltarea unui nou mod de gândire, integrator, ancorat în actualitatea socioculturală complexă, autonom, creativ, deschis. Procesul învățării transcende educația formală și depinde de interacțiuni realizate cu o multitudine de surse externe situate în zona proximei dezvoltări individuale și în orizontul motivațional personal. Omul și lumea interacționează continuu, realitatea este văzută din perspective multiple, informația ne parvine prin diverse canale, astfel încât, pentru a avea un răspuns adecvat, cunoștințele noi trebuie produse, și nu reproduse.

Aspectele cheie ale competențelor de instruire asistată de calculator sunt mai degrabă de ordin pedagogic decât tehnic. Rolul noilor tehnologii în învățare, și al softului educațional în special, variază de la oferirea accesului la informații și cunoaștere la facilitarea înțelegerii fenomenelor, instruire individualizată etc.

Statisticile realizate în perioada 26.01.2007 – 14.06.2007 la Centrul pentru Dezvoltare și Inovare în Educație Elearning.România arată că cel mai important rol în dezvoltarea Elearning în România îl vor avea Utilizatorii, aceștia obținând 39% din cele 116 voturi, urmați de Firmele dezvoltatoare de soft cu 33% și Politicile Uniunii Europene cu 22%, Ministerul Educației și Cercetării obținând doar 6%. Acest rezultat este într-un fel logic deoarece utilizatorii sunt cei care obțin cele mai multe beneficii din folosirea acestor platforme de Elearning.

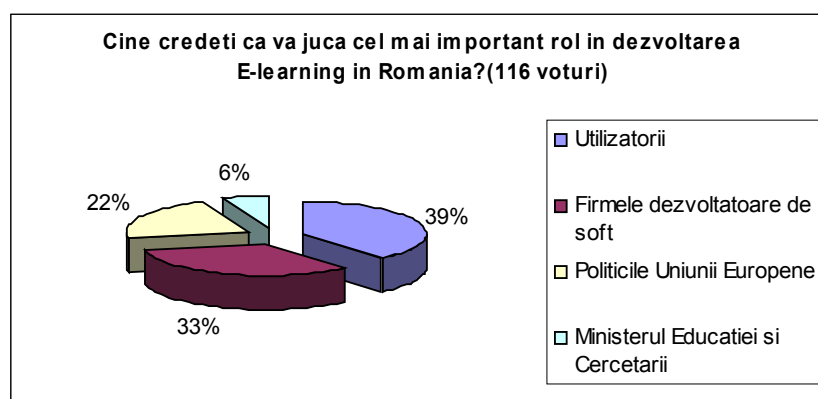


Fig. 1 Statistici e-Learning

## 2 Formularea Problemei

Învățământul deschis și la distanță (IDD) utilizează metode noi pentru a determina o flexibilitate crescută a procesului de învățământ în ceea ce privește spațiul, timpul, alegerea conținutului sau a resurselor pedagogice. Promovarea IDD se justifică prin potențialul acestuia de a realiza unele obiective generale ale învățământului actual cum sunt: educația permanentă și deschiderea spre societate.

Învățământ deschis (ca acces, durată și ritm) este o noțiune preponderent strategică, punerea în aplicare se realizează prin decizie la nivel de politică educațională. IDD-ul (distanța este între cadrul didactic și student în spațiu și/sau în timp) este o noțiune predominant tehnică, implementarea ei aparținând deciziei instituționale.

Principalele caracteristici ale educației la distanță ar fi: separarea dintre student și profesor în timp și spațiu, utilizarea tehnologiei în scopuri educaționale, o relație pedagogică individualizată, caracterul activ și constructiv al învățării, orientarea pe problemă în învățarea la distanță.

Probleme pe care le poate rezolva educația la distanță sunt: distanța fizică, probleme de timp și de programare, spații disponibile limitate, înscrieri limitate sau dispersate, număr limitat de cadre didactice disponibile. Toate aceste, probleme sunt rezolvate, datorită utilizării Internetului și a spațiului virtual.

Ca avantaje ale învățării cu materiale pe suport informatic amintim: flexibilitatea crescută a materialului, și posibilitatea de reactualizare permanentă și cu costuri minime, integrarea unei varietăți de medii de învățare, posibilitatea de acces imediat la o multitudine de alte resurse disponibile pe Internet, o motivare crescută și o angajare mult mai activă a studentului în învățare.

Există bine-nțeleș și limite ale IDD, și anume: necesitatea de familiarizare permanentă cu noile facilități pe care le aduce tehnologia, costurile ridicate ale implementării ei, starea de izolare a studentului care nu permite menținerea unui nivel corespunzător al motivației pe termen lung, dificultatea de organizare a studiului și de rezervare a timpului necesar pentru acesta, dificultatea în dezvoltarea unor deprinderi speciale, supraîncărcarea cu informații.

În lucrarea de față se dorește prezentarea, prin cadrul realizării unei aplicații web, a unei platforme de e-learning ca suport pentru învățământ la distanță, accesul la informație făcându-se on-line, Internetul devenind astfel nu doar un instrument de căutare și informare, ci și unul de învățare, scopul lucrării fiind bine-nțeleș unul pedagogic. Astfel, accesul la baza de date a platformei se face printr-o simplă accesare a site-ului, autentificarea utilizatorului făcându-se prin log-in pe baza de user și parolă.

Un astfel de site, este foarte avantajos: găzduiește multă informație care poate fi accesată non stop, nu necesită un spațiu real, nu necesită sincronizare din partea cursanților, astfel că aceștia își pot organiza învățarea în funcție de timpul liber, iar evaluarea este interactivă și eficientă. Toate aceste avantaje și multe altele, vor fi prezentate prin intermediul lucrării de față.

Principalele sub-obiective în crearea aplicației au fost: crearea unei interfețe cât mai prietenoasă pentru accesul ușor și rapid la informații, personalizată în funcție de utilizator; realizarea a 3 nivele de utilizatori: student, profesor, administrator cu drepturi de acces diferite.

Pe lângă aceste cerințe de bază, s-a dorit realizarea unei biblioteci virtuale pe care studenții și profesorii să o poată accesa ori de câte ori e nevoie, și a unui forum de discuții pentru studenți și profesori.

## Realizarea unei platforme de e-learning pentru învățământ la distanță

Site-ul este dedicat doar studenților și profesorilor din cadrul ULB Sibiu, Facultatea de Științe, pentru fiecare dintre aceștia existând un cont curent. Prin log-are, fiecare utilizator în parte va obține acces la informațiile personale.

### 3 Rezultate principale



Fig. 2 Interfața site-ului

Un punct cheie în realizarea aplicației a fost crearea unei baze de date cât mai bine pusă la punct. Pentru acest lucru am folosit serverul de baze de date MySQL, care este totodată și cel mai popular sistem de management pentru baze de date relaționale. Este de menționat faptul că este Open Source. MySQL Server a fost creat pentru a lucra cu baze de date mai rapid decât soluțiile existente deja și este folosit de ani buni în medii foarte solicitante.

Într-o bază de date relațională datele sunt stocate în mai multe tabele separate, fiind îmbunătățite astfel, viteza și flexibilitatea. Tabelele pot fi legate, între ele, prin relații definite de noi, fiind astfel posibil să combinăm la cerere datele din mai multe tabele.

Pentru crearea paginilor web am folosit limbajul PHP.

Limbajul PHP este una dintre cele mai interesante tehnologii existente în prezent. Deoarece îmbină caracteristici dintre cele mai complexe cu simplitatea în utilizare, a devenit rapid un instrument de frunte pentru dezvoltarea aplicațiilor în web fiind un limbaj de programare comod.

Ca și alte limbaje de scripting pentru web, PHP permite furnizarea unui conținut web dinamic, adică un conținut web care se modifică automat, după necesități, de la un minut la altul sau de la o zi la alta. Conținutul web este un element important în susținerea traficului unui site web; de regulă vizitatorii nu mai revin la o pagină web care conține aceleași informații ca și cele prezentate anterior.

Mai mult, spre deosebire de alte limbaje de scripting, PHP rulează pe serverul web, nu în browserul web. În consecință, PHP poate obține accesul la fișiere, baze de date și alte resurse. Ca server web am folosit serverul de la Apache, care și el este open source.

În această combinație, PHP, MySQL și Apache, am putut oferi aplicației o performanță ridicată, portabilitate, flexibilitate, accesibilitate, dar și o securitate crescută.

Securizarea bazei de date a fost făcută prin controlul accesului – unde fiecărui utilizator informațional i s-a atribuit un set de acțiuni permise, pe care le poate efectua în raport cu anumite obiecte; și prin controlul autenticității – realizează dacă utilizatorul care încearcă să efectueze o acțiune sau alta este anume acela pe care îl reprezintă.



Lucrând cu useri și parole, este nevoie de o securizare a acestora. PHP pune la dispoziția programatorului funcția md5() pentru criptarea informațiilor "prețioase". md5 este o metoda de criptare folosind un algoritm special care furnizează ca rezultat un număr hexazecimal de 32 caractere.

O altă problemă importantă când avem un site cu multe pagini este atunci când dorim să facem anumite schimbări în elementele paginii: fondul, grafica sau fontul textelor din pagini.

Prin utilizarea CSS (Cascading Style Sheets), în traducere "*foi de stil în cascada*", acest lucru nu mai este o problemă, realizându-se relativ ușor, prin schimbarea sau adăugarea unor elemente în codul CSS, ne fiind nevoie să lucrăm la fiecare pagină sau la fiecare element din pagină.

CSS se ocupă în general cu aspectul și controlul grafic al elementelor din pagină, cum ar fi: textul, imaginile, fondul, culorile. CSS folosește stiluri, acestea înglobează, sub un anumit nume, atribute de formatare care se aplică asupra unui element individual din pagina sau la nivelul întregului document. CSS funcționează cu HTML, însă nu este HTML. El extinde funcționalitățile HTML, permițând redefinirea etichetelor HTML existente.

Prin utilizarea CSS aspectul documentului pe ansamblu, sau a unui element individual din interiorul său, poate fi controlat mult mai ușor. Stilurile pot fi aplicate asupra unui element, a unui document sau chiar asupra unui întreg site web.

Un dezavantaj ar fi că unele navigatoare nu sunt compatibile CSS, astfel că documentele HTML sunt afișate ca și cum CSS n-ar exista, dar cele mai cunoscute și utilizate browsere, cum ar fi: Mozilla Firefox, Internet Explorer, Opera, și altele, sunt compatibile CSS. Pentru a rezolva și această problemă, la deschiderea paginii principale, se face o verificare a tipului de browser pe care utilizatorul îl folosește.

### 3.1 Modulul pentru studenți

Cerințele referitoare la modulul pentru studenți au fost:

- accesarea materialelor didactice prin vizualizarea în browser sau posibilitate de download,
- rezolvarea de teste grilă,
- vizualizare anunțuri și informații utile.

Materialele didactice pot fi accesate în orice moment de un student log-at la platforma de e-Learning. Studentul are posibilitatea de a selecta dintr-o listă de materii materia dorită, apoi cursul pe care dorește să-l citească. Lista de materii este populată automat la intrarea în pagină cu materiile corespunzătoare specializării și anului de studiu al studentului log-at., iar la alegerea unei materii se va popula și lista de cursuri cu materialele disponibile în baza de date. Acum studentul are posibilitatea atât de a vizualiza cursurile în browser dar și de a descărca aceste materiale.

Pe lângă posibilitatea de a vizualiza cursuri, studentul are posibilitatea de a-și verifica cunoștințele prin rezolvare de teste grilă, alegerea unui test pentru rezolvare făcându-se similar alegerii unui curs pentru citit. Dacă s-a ales un test pentru rezolvare, acesta va fi încărcat din baza de date întrebare cu întrebare, studentul având posibilitatea de a merge la următoarea întrebare, întrebarea precedentă sau să termine testul pentru a vedea rezultatul.

În orice moment un student log-at are posibilitatea de a vedea rezultatele obținute la examenele anterioare. Dacă studentul log-at este în an superior, el poate vedea și rezultatele obținute în anii anteriori, alegând anul dorit dintr-o listă.

### 3.2 Modulul pentru profesori

Cerințele referitoare la modulul pentru profesori au fost:

- posibilitatea de upload cursuri și materiale de învățare și evaluare,
- crearea de teste grilă cu modificare sau ștergere întrebări,
- vizualizare rezultate studenți.

Fiecare profesor are posibilitatea de a adăuga materiale pentru materiile predate. În primă fază, profesorul trebuie să aleagă una dintre materiile predate, materie pentru care se dorește adăugarea de

## Realizarea unei platforme de e-learning pentru învățământ la distanță

materiale; apoi are următoarele opțiuni: adăugarea prin upload a unui curs, adăugarea de întrebări la un test grilă existent în baza de date, sau crearea unui nou test. Dacă s-a ales opțiunea de upload curs, tot ce mai trebuie făcut este alegerea cursului dorit de pe calculatorul propriu și încărcarea lui pe server.

Un test grilă este dat ca o succesiune de întrebări memorate în baza de date, astfel că pentru crearea unui test este nevoie de adăugarea lui întrebare cu întrebare. Întrebările vor fi memorate în baza de date primind un id unic, dar și id-ul testului din care fac parte. În orice moment profesorul poate vedea un anumit test, și dacă este cazul să modifice sau să șteargă întrebări.

Asemenea studenților, profesorul are posibilitatea de a vizualiza materialele disponibile, dar și să șteargă materiale dacă acestea nu mai sunt folosite.

Orice profesor are dreptul de a vedea rezultatele obținute de studenții săi la materiile pe care acesta le predă. Pentru a vizualiza rezultatele, profesorul trebuie să aleagă materia dorită din lista de materii. Pentru o filtrare a rezultatelor se pot completa și câmpurile pentru specializare, grupă sau chiar numele unui anumit student.

### 3.3 Modulul pentru administrator

Administratorul este cel care gestionează baza de date, astfel, el este singurul care poate crea sau șterge conturi pentru profesori, studenți sau pentru alți administratori. Există 2 categorii de administratori și anume administratorul-creator, care se ocupă nu doar de crearea conturilor dar și de întreținerea site-ului, și administratorul-secretară, care are dreptul de a introduce date personale ale studenților și note.

Crearea de conturi pentru profesori și studenți poate fi făcută doar de către administratorul-creator. Inițial contul va fi numele complet al utilizatorului iar parola CNP-ul acestuia. Atât profesorii cât și studenții pot modifica în orice moment această parolă. De asemenea administratorul este cel care șterge conturi, de exemplu pentru studenții care au absolvit de cel puțin 6 luni, conturile acestora nu mai sunt necesare. Tot administratorul-creator se ocupă și de biblioteca virtuală. El poate adăuga cărți în bibliotecă prin upload.

Administratorul-secretară este cel care administrează baza de date referitoare la datele personale ale studenților și la notele obținute de aceștia. La începutul anului universitar se face o reactualizare a bazei de date. De asemenea orice modificare apărută în timpul anului, și actualizarea cataloagelor sunt făcute tot de administratorul-secretară.

### 3.4 Biblioteca virtuală

Biblioteca virtuală conține o varietate de cărți pentru toate specializările, dar pentru a putea folosi cărți din bibliotecă trebuie să fi utilizator al platformei, adică student sau profesor. Pentru a putea intra în bibliotecă este obligatorie autentificarea. Pentru căutarea unei cărți, se alege mai întâi specializarea dorită și scrierea în câmpul de căutare a unui cuvânt cheie. Acest cuvânt cheie poate fi orice cuvânt ce are legătură cu cartea dorită, căutarea cărții în baza de date făcându-se atât în câmpul de titlu, cât și în câmpurile autor sau descriere. Astfel că la apăsarea butonului caută se va afișa o listă cu toate cărțile din bibliotecă ce conțin în titlu, autor sau descriere cuvântul căutat. Din lista de cărți se alege cartea căutată, și se deschide o pagină html în care se poate citi conținutul acesteia.

De asemenea biblioteca conține și înregistrări video ale unor lecții mai importante predate de profesori, astfel ca înțelegerea lor este mult mai ușoară.

## 4. Concluzii și dezvoltări viitoare

Pe lângă obiectivele propuse, se poate adăuga un chat on-line prin care studenții să poată socializa cu alți colegi și realizarea de consultații on-line oferite de profesori în sprijinul cursanților, astfel că odată pe săptămână studenții pot intra în contact direct cu profesorii pentru mai multe informații.

O noutate pentru platformele de e-learning ar fi realizarea unei clase virtuale, creând astfel o asemănare mai puternică cu mediul real de învățământ.

### Bibliografie

1. PHP 5 Power Programming, Andi Gutmans, Stig Sæther Bakken, and Derick Rethans
2. <http://www.marplo.net/php-mysql>

3. <http://didu.ulbsibiu.ro>
4. <http://academiaonline.ro>

**Epure Ancuța-Raluca**  
**Universitatea „Lucian Blaga” din Sibiu**  
**Facultatea de Științe**  
**Specializarea Informatică, An III**  
[epure\\_ancuta\\_raluca@yahoo.com](mailto:epure_ancuta_raluca@yahoo.com)

## Sistem de management și gestiune a fișierelor

**Autor:** Iulia - Maria Lazăr, Universitatea „Lucian Blaga” din Sibiu

**Prof. Coordonator:** Daniel Ioan Hunyadi, Facultatea de Științe, Specializarea Informatică

*Abstract: A file manager is an indispensable application in terms of handling data, especially when it offers a rich user interface and complete functionality. Since its logic derives from document management solutions, it follows that common operations that a person would do with real papers are implemented. The file manager facilitates the decision of file organization providing key features such as a vast view on data classification, fast access to files and folders and specific management actions.*

*Cuvinte cheie: file management, document management solutions, object-oriented technology, software reliability, software portability, organization*

### 1 Introducere

De la apariția primei ierarhizări a fișierelor a apărut nevoia unei afișări cât mai structurate a acestora. Gruparea fișierelor în directoare este realizată prin schimbarea indexului selectat cu informații din FAT (File allocation table) sau MFT (Master File Table) în funcție de sistemul de operare utilizat. În acest index locația fizică a fișierelor pe un mediu este memorată alături de poziția sa în ierarhie. Diferite metode de afișare și management au fost gândite în vederea afișării și gestionării lui.

Ca și concept un file manager este o fereastră care arată conținutul unui director, fie el localizat pe un mediu de stocare extern, fie un director ales de utilizator din cadrul suportului fizic propriu. Acest concept a fost prima dată introdus de John Socha prin intermediul binecunoscutului Norton Commander pentru DOS în anul 1986. Prescurtat NC, acesta a fost un manager prototip implementat ca o interfață text cu posibilitatea efectuării de acțiuni în linie comandă. NC a fost dat la o parte și a fost înlocuit de Windows explorer o dată cu apariția sistemului de operare Windows 95. Cu trecerea anilor au apărut alte sisteme adaptate noilor tehnologii și noilor cereri aduse de creșterea accelerată în volum a informațiilor.

Caracteristicile cheie ale unui astfel de manager se axează pe afișarea, manipularea și organizarea datelor.

Convențional, fereastra de vizualizare e împărțită în trei secțiuni:

- Un panel de navigare a directoarelor bazat pe ierarhizarea relațiilor rădăcină-frunză
- Un al doilea panel de navigare a directoarelor, însă de această dată frunzele vor putea fi și fișiere-fiu
- O fereastră pentru linie comandă.

Utilizatorul este îndrumat mai departe pentru o mai bună gestionare a datelor prin intermediul unei interfețe intuitive spre folosirea operațiilor comune de copiere, mutare și ștergere a datelor ajungând astfel la tehnică de bază de organizare luând în considerare modalitatea cea mai probabilă de utilizare a fișierelor.

Această lucrare realizează un înlocuitor al Windows Explorer-ului cu surprinzătoare noi caracteristici: noi iconițe pentru partiții și My Computer, un design minimalist și intuitiv, afișarea dimensiunii reale a fișierelor exprimată în multipli de octeți, deschiderea shortcut-urilor în interiorul aceleiași aplicații atingând un nou grad de independență, interacțiune cu componentele hardware în bara de monitorizare a fișierului sau directorului curent, portabilitatea asigurată de setup-ul de instalare, animarea aplicației cu efecte speciale de transformare a opacității specifice în transparență totală până la închidere la apăsarea butonului de închidere a aplicației și ajutorarea procesului de navigare prin adăugarea efectelor de urmărire a cursorului la simpla parcurgere a unei ierarhii de frunze al unui director.

Un alt plus este dat de posibilitatea personalizării aspectului. Setup-ul este prevăzut cu un folder denumit „Skins” în care utilizatorul poate avea acces la schimbarea iconițelor.

Aplicația este gândită să ofere o gestionare ordonată a conținutului suportului fizic atât utilizatorului avansat cât și începătorului.

## 2 Formularea Problemei

Pe măsura trecerii timpului au apărut alternative la Windows Explorer din necesitatea de a obține o navigare îmbunătățită în indexul fișierelor și o interfață vizuală cât mai intuitivă.

O dată cu apariția acestor alternative, concurența a devenit din ce în ce mai intensă iar costurile sistemelor au crescut. Lucrarea propune o aplicație îmbunătățită, gratuită, scrisă în limbajul de programare C#.

## 3 Rezultate principale

C# este conceput pentru dezvoltarea de aplicații software care pot profita de resursele puse la dispoziție de sistemul de operare. Este un limbaj de programare orientat pe obiect și orientat pe componente. Programarea se bazează pe interacțiunea “obiectelor” cu aplicațiile și programele corelate.

Fiecare fișier este văzut ca o clasă, deci ca un obiect, beneficiind de metode și atribute specifice.

Containerul claselor, metodelor și proprietăților necesare manipulării acestor obiecte este System.IO (Input/Output). Menționez aici câteva din clasele folosite cu precădere: Directory, File, DirectoryInfo și FileInfo. Acestea se ocupă cu crearea, ștergerea și mutarea fișierelor și directorilor. Ele obțin informații specifice despre fișiere utilizând proprietățile definite în interiorul aceluiași clase.

Clasa de bază a DirectoryInfo și FileInfo este FileSystemInfo, o clasă abstractă, fapt ce duce la imposibilitatea instanțierii, însă un set de proprietăți pun la dispoziție atributele asociate unui fișier precum: data creării, ultima accesare, calea fizică întreagă, numele și altele.

Aplicația este populată cu obiecte pe baza adresei lor fizice. Am adăugat obiectelor funcționalitate în plus prin intermediul polimorfismului, specific programării orientate pe obiecte, pentru facilitarea accesului la informații.

Prin polimorfism se înțelege transmiterea unui obiect în locul unui alt obiect, în cazul meu, un obiect mai complet din punctul de vedere al informațiilor necesare realizării proiectului. În funcție de componenta pe care utilizatorul se poziționează, obiectul poate avea fie detalii despre ultima cale accesată, fie despre informațiile oferite de clasele mai sus amintite.

### 3.1 Aspectul aplicației

Sistemul de management și gestiune a fișierelor este realizat într-un proiect de tip winform în mediul de programare Visual Studio 2008. Windows Forms este numele dat de Microsoft unei interfețe de programare GUI (Graphical User Interface). Fereastra de lucru este împărțită în trei zone de lucru principale:

- Panelul de navigare, poziționat în partea stângă a ferestrei, este zona unde se încarcă ierarhia de directorii ținându-se cont de rădăcini, sau părinți, și frunze, sau copii. Pentru realizarea acestei zone am folosit controlul „Tree View” pentru o vedere arborescentă.
- Al doilea panel de navigare, situat în partea dreaptă a ferestrei, se ocupă cu popularea tuturor directorilor și fișierelor existente pe suportul fizic. Controlul responsabil cu acest panel este un „List View” ce urmează să ofere sprijin aplicației pentru efectuarea unor operații specifice.
- Zona de inserare a comenzilor de către utilizator, denumită bara de adrese cu controlul asociat „Navigation Bar”, este poziționată în partea superioară a sistemului.

Alături de aceste zone, am implementat un User Control, denumit „File Status Control”, care permite o monitorizare rapidă și eficientă a sistemului și a obiectului folosit curent. Poziționarea lui în partea de jos a aplicației a fost prevăzută pentru a evita crearea confuziilor în cazul unui utilizator începător.

## Sistem de management si gestiune a fisierelor

Aplicația are opțiunea de afișare a unor meniuri contextuale la efectuarea unui click pe butonul drept al mouse-ului.

Cursorul mouse-ului va lua forma curosului de așteptare de fiecare dată când programul execută o operație mai greoaie.

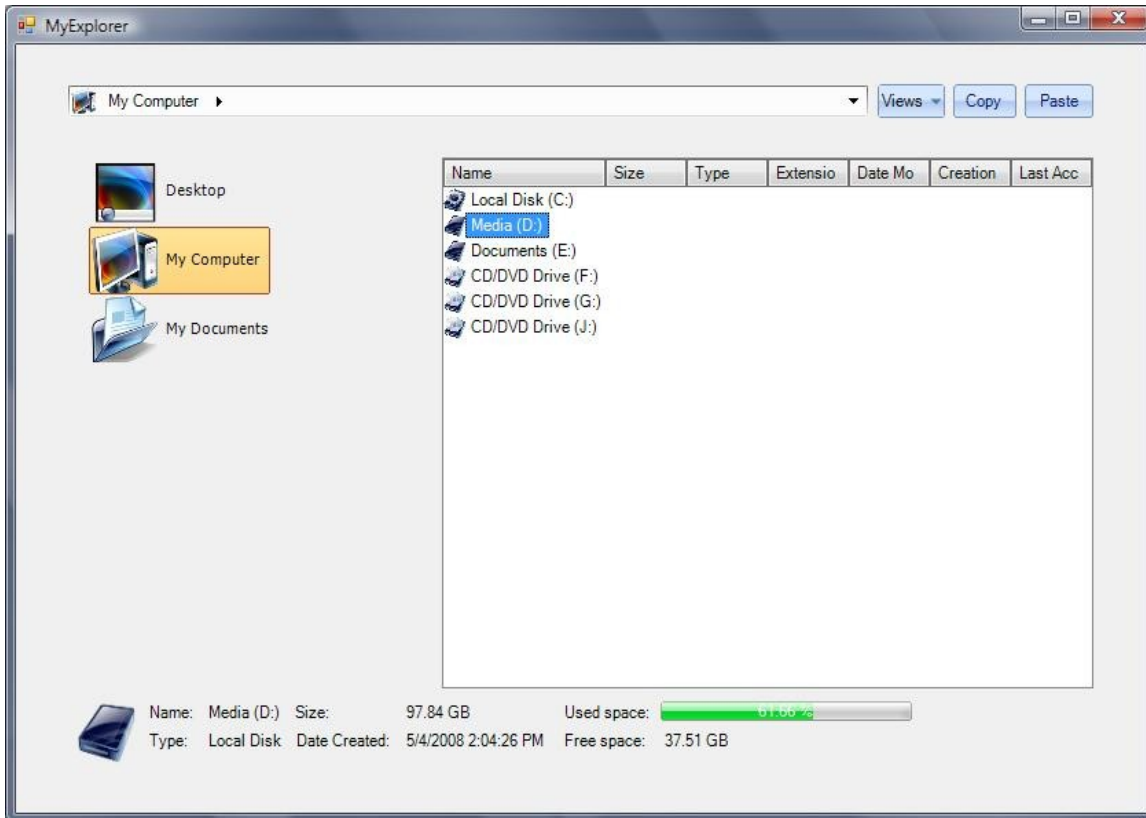


Fig.1 – Aplicația în stare inițială

Butonul de vizualizare în diferite stiluri a iconițelor este implementat cu succes și permite schimbarea modalității în orice moment dorit. Tipurile de vizualizare disponibile sunt: Tiles, List, Details, Thumbnails și Icons.

Alte elemente care vor apărea pe suprafața aplicației vor fi butoanele de copy și paste a căror funcționalitate este cea de copiere respectiv lipire a unuia sau mai multor fișiere sau directoare în felul prezentat de Figura 1.

### 3.1.1 Tree View (Arborele)

Tree View-ul este implementat alături de o clasă manager care se ocupă de popularea și navigarea în interiorul său. Popularea se face pe principiul ierarhiei arborescente, adică de la cel mai important nod numit rădăcină la cel mai adânc fiu. Primele noduri adăugate sunt: Desktop, My Computer și My Documents.

Exemplificare prin cod sursă:

```
CustomTreeNode desktop = new CustomTreeNode();
    desktop.Name = desktopNodeName;
    desktop.Text = desktopNodeText;
CustomTreeNode myComputer = new CustomTreeNode();
    myComputer.Name = computerNodeName;
    myComputer.Text = computerNodeText;
CustomTreeNode myDocuments = new CustomTreeNode();
    myDocuments.Name = documentsNodeName;
    myDocuments.Text = documentsNodeText;
```

Apariția clasei CustomTreeNode se datorează polimorfismului mai sus prezentat. Clasa CustomTreeNode derivă din clasa de bază care se ocupă cu nodurile standard ale Tree View-ului căreia îi

adaugă un membru de tip FileSystemInfo pentru a putea realiza în următoarea etapă, la popularea copiilor celor trei rădăcini, asocierea noilor noduri și crearea lor pe baza adresei fizice.

Popularea nodurilor rădăcină se va face în funcție de nodul selectat. Comportamentul este diferit pentru fiecare din cele trei cazuri, dat fiind faptul că My Computer nu are adresă fizică, este un folder virtual, iar copii lui imediați sunt de tip partiții logice.

### 3.1.2 List View (Lista)

List View este controlul responsabil cu vizualizarea în diferite stiluri a fișierelor și directoarelor. Va fi populat în funcție de o cale primită de o metodă a manager-ului său. O dată primită calea, metoda realizează o parcurgere a obiectelor găsite în indexul respectiv și le va afișa. De această dată obiectele controlului poartă denumirea de itemi dar ca o similaritate față de implementarea de mai sus, am păstrat ideea polimorfismului adăugând informația suplimentară vis-a-vis de itemul standard și anume: fiecare item va avea un FileInfo asociat ce furnizează detalii pentru facilitarea manipulării.

### 3.1.3 Navigation Bar (Bara de adrese)

Bara de adrese face parte din cele trei zone specifice unei astfel de aplicații. Aspectul este asemănător cu cea a Windows Explorer-ului din Windows Vista.

Navigarea se face ierarhic, continuând pe principiul arborelui. Este prevăzută cu metode de salvare a locațiilor cel mai des vizitate astfel încât personalizarea să aibă loc constant. La ieșirea din aplicație informațiile se stochează într-un fișier de tip xml ce va fi apoi citit la fiecare pornire a aplicației.

### 3.1.4 File Status Control (User Control)

File Status Control este un user control destinat monitorizării informațiilor despre fișierul sau directorul curent. Bazat pe etichete și imagini va avea comportament diferit pentru partițiile sistemului, caz în care va arăta detalii legate de procentul de spațiu liber sau afișarea unei etichete de comunicare hardware pentru exercitarea funcției de „eject” în cazul cd/dvdrom-ului, pentru fișiere sau pentru directoare.

## 3.2 Principiul de funcționare

Structura internă a aplicației este bazată pe sincronizarea componentelor în cadrul aplicației. În momentul schimbării locației dintr-o zonă, celelalte trebuie să urmeze navigarea automat.

Această problemă a fost soluționată prin crearea unui eveniment user-defined. În limbajul de programare C#, un eveniment are ca membru privat un delegate ce realizează corelarea declanșatorului evenimentului cu handlerul de tratare.

Fiecare control principal are în clasa manager asociată un handler de tratare a evenimentului de schimbare a locației curente.

Mecanismul este următorul:

- Are loc identificarea declanșatorului
- Fiecare handler va fi parcurs efectuându-se operația de sincronizare.

O serie de instrucțiuni de verificare apar din necesitatea identificării operațiilor corespunzătoare și pentru evitarea repetiției. În funcție de componenta declanșatoare, măsuri de parcurgere sau populare în toate componentele vor fi luate.

Procesul este prezentat în Figura 2.

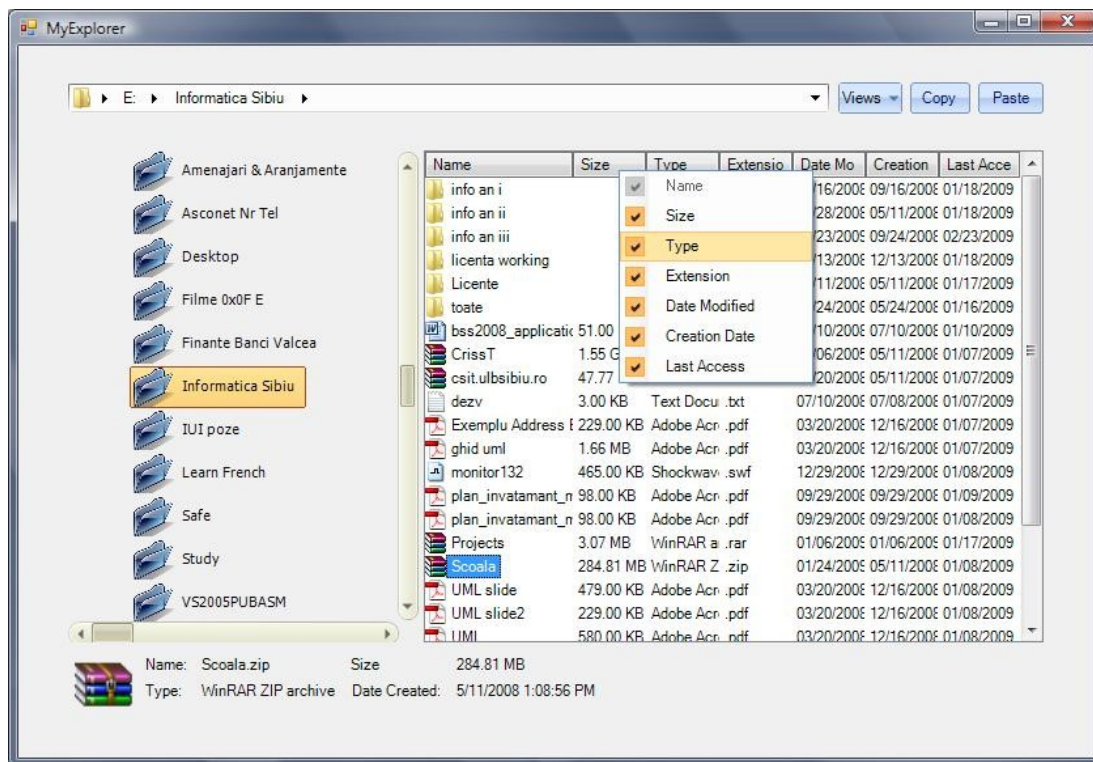


Fig.2 – Procesul de sincronizare

#### 4. Concluzii și dezvoltari viitoare

Conceput ca un sistem compact, una din extensiile propuse pe viitor este implementarea unei metode de copiere si mutare a fișierelor la viteză maximă pe principiul structurii de date de tip coadă. Utilizatorul poate realiza mai multe acțiuni de copiere în același timp. Sistemul va așeza pe o coadă fiecare acțiune ordonat și le va efectua pe rând.

O altă extensie este o unealtă integrată de vizualizare a imaginilor. Prin implementare se va ajunge la independența față de contextul de rulare.

Am reușit să creez o aplicație ce oferă utilizatorului o interfață grafică bogată cu funcționalitate completă, pentru o mai buna gestionare a volumului de informații existente pe suportul fizic.

#### Bibliografie

1. Daniel Solis, *Illustrated C# 2008*, Editura Apress, 2008
2. Andrew Troelsen, *Pro C# 2008 and the .NET 3.5 Platform*, Editura Apress, 2008
3. Chris Sells, *Windows Forms Programming in C#*, Editura Addison Wesley, 2003
4. [http://en.wikipedia.org/wiki/Norton\\_Commander#Screenshots](http://en.wikipedia.org/wiki/Norton_Commander#Screenshots)
5. <http://msdn.microsoft.com/en-us/library/system.io.directoryinfo.aspx>

**Iulia-Maria Lazăr**  
**Universitatea „Lucian Blaga” din Sibiu**  
**Facultatea de Științe**  
**Specializarea Informatică**  
[iulia\\_je@yahoo.com](mailto:iulia_je@yahoo.com)



## Image deblurring/Degradarea imaginii

**Autori:** Lazar Valentina, **Universitatea** "Lucian Blaga", Sibiu

**Coordonator:** Lector Univ. Drd. Ralf Fabian, **Universitatea** "Lucian Blaga", Sibiu

*Abstract: The problem of image deblurring in the presence of salt and pepper noise is considered. Standard image deconvolution algorithms, that are designed for Gaussian noise, do not perform well in the case. Median type filtering is a common method for salt and pepper noise removal. Deblurring an image that has been preprocessed by median-type filtering is however difficult, due to the amplification (in the deconvolution stage) of median-induced distortion. A unified variational approach to salt and pepper noise removal and image deblurring is presented. An objective functional that represents the goals of deblurring, noise-robustness and compliance with the piecewise-smooth image model is formulated. Promising experimental results are shown for blur model.*

*Key-Words: degradarea, operatorul de distorsiune, functia de transfer optic(OTF), punctual de raspandire a functiei (PSF), deconvwnr, deconvdreg, deconvlucy, deconvblind.*

### 1. Introducere

Prelucrarea și analiza imaginilor (numite adeseori prescurtat doar prelucrarea imaginilor) s-a născut datorită ideii și necesității de a înlocui observatorul uman printr-o mașină. Este important de precizat că analiza imaginilor a mers mai departe decât simpla înlocuire a observatorului uman, deoarece au apărut soluții inovatoare pentru probleme cu care acesta nu mai fusese confruntat - ca în cazul imaginilor non-vizibile (imagini acustice, ultrasonore, radar). După cum se remarcă în prelucrarea imaginilor înglobează posibilitatea de a dezvolta mașina totală de viziune, capabilă să realizeze funcțiile vizuale ale oricărui viitoritar (desigur, după realizarea a importante dezvoltări teoretice și tehnologice).

În continuare vom dezbate câteva subiecte legate de cauzele și metodele de combatere a degradării imaginilor.

### 2. Cauze ale Blurring

Printre **cauzele blurring** care pot determina degradarea unei imagini putem enumera următoarele:

- tulburările atmosferice care creează condiții improprii pentru captarea imaginilor (Ex.: ploaia sau vântul care ridică praful între dispozitivul de captare și imagine);
- lipsa luminii care duce la reducerea numărului de fotoni ce pot fi captați în timpul achiziției imaginii și astfel poate duce la apariția unor umbre sau la degradarea totală a imaginii;
- mișcarea în timpul captării.

Pentru a înțelege modul de funcționare vom prezenta **modelul Deblurring** ce analizează estomparea sau degradarea unei imagini, bazându-se pe următoarea ecuație:  $\mathbf{g} = \mathbf{H}\mathbf{F} + \mathbf{n}$  unde:

- $\mathbf{g}$  – estomparea imaginii;
- $\mathbf{H}$  – operatorul de denaturare, numit și punctul de răspandire al funcției (point spread function=PSF);
- $\mathbf{F}$  – originalul (imaginea reală);
- $\mathbf{n}$  – perturbarea introdusă în timpul achiziției imaginii (acesta este factorul ce corupe imaginea);

In urma prezentarii acestui model putem trage urmatoarea **concluzie**: imaginea adevarata **f** nu exista. Aceasta imagine reprezinta imaginea perfecta ce ar trebui sa fie daca aceasta s-ar fi facut in conditii propice de achicitie.

Dupa cum reiese din cele spuse anterior se observa ca un punct important in derularea acestui model il reprezinta cunoasterea **PSF**.

In spatiu, PSF descrie modul in care un sistem optic propaga un punct de lumina. Astfel PSF este invers transformarii Fourier prin functia de transfer optic(OTF). Denaturarea actiunii operatorului asupra imaginii creaza aceste distorsiuni.

In cadrul acestui model, este fundamentala cunoasterea gradului de estompare a imaginii cu PSF care descrie exact distorsiunea.

**Nota** :calitatea imaginii este in principal determinata de cunoasterea PSF.

### 3. Functii Deblurring

Pentru a corecta eventualele degradari ale imaginilor este necesara folosirea unor **functii Matlab**.

Toate aceste functii accepta un PSF si imaginile denaturate ca argumente ale lor.

Dintre acestea vom aduce la cunostiinta urmatoarele:

- **deconvwnr** – pune in aplicare o solutie folosind regula celor mai mici patrate. Astfel sunt oferite informatii cu privire la zgomotul posibil si modalitati de reducere a acestuia.
- **deconvreg** – functioneaza la fel ca functia deconvwnr doar pe un cadu mai restrans, in cazul in care se poate constrange privirea la locul de iesire a imaginii.
- **deconvlucy** – pune in aplicare un accelerator folosit in algoritmul Lucy Richardson. Aceasta functie foloseste multiple repetari prin tehnicile de optimizare din statisticile Poisson.
- **deconvblind** – efectueaza distorsiunile fara acunoaste PSF in prealabil. Se trece ca argument initial un PSF aleator, functia returnand o restaurare a PSF in plus fata de restaurarea imaginii.

In cazul in care dorim sa folosim una din functiile prezentate mai sus trebuie retinute urmatoarele **observatii**:

- Deblurring este un proces iterativ. Este posibila repetarea procesului de mai multe ori, modificarea parametrilor specificati in interiorul functiilor deblurring la fiecare repetare pana la obtinerea unei imaginii cat mai apropiate de cea originala.
- Pentru a evita perturbarile sonore intr-o imagine distorsionata, se poate folosii functia edgetaper la preprocesarea imaginii, inainte de a trece la functiile propriu-zise

### 4. Exemplu Image Deblurring

Pentru a exemplifica cele prezentate teoretic vom da urmatorul exemplu:

**Enunt**: Utilizați deconvwnr si restaurati o imagine .Creați un zgomot, imagini neclare și apoi aplicati filtrul deconvwnr pentru a obtine imaginea corecta.

**Rezolvare:**

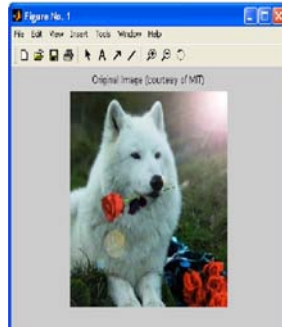
*Pasul 1.* Se afisaza imaginea originala (fig1.):

```
I = im2double (imread ( "abc.jpg "));
```

```
imshow (I);
```

```
titlu ( "imagine originală (curtoazie de MIT) " );
```

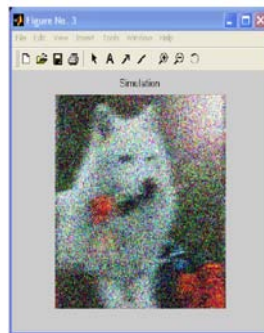
## Image deblurring/Degradarea imaginii



**Figura 1 - Imaginea originala**

*Pasul 2.* Se simuleaza o imagine estompata (fig 2.) prin utilizarea imaginii initiale, a PSF si a doua atribute de afisaj:

**estompează = imfilter (I, PSF, "conv", "circular '');**

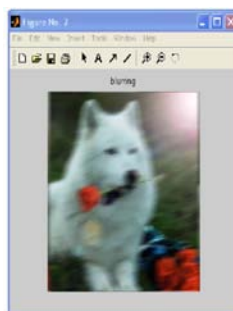


**Figura 2 - Imaginea bruiata**

*Pasul 3.* Se analizeaza zgomotul cu ajutorul urmatoarei functii:

**blurred\_noisy = imnoise(blurred, 'gaussian', ...  
noise\_mean, noise\_var);**

obtinandu-se simularea imaginii perturbate.(fig 3.)



**Figure 3 - Simularea imaginii estompatei**

*Pasul 4.* Se incearca restaurarea presupunand un zgomot foarte mic sau chiar fara zgomot astfel:

**estimated\_nsr = 0;  
wnr2 = deconvwnr(blurred\_noisy, PSF, estimated\_nsr);**

*Pasul 5.* Restaurare prin utilizarea raportului dintre zgomot si semnalul de receptie a imaginii si imaginea initiala nebruiata:

**estimated\_nsr = noise\_var / var (I (:));**

## 5. Concluzii

Dupa prezentarea teoretica si a acestui scurt exemplu putem trage urmatoarele concluzii:

- degradarea imaginilor este o problema tot mai des intalmita iar dorinta de rezolvare a acesteia a determinat cercetarea in amanunt si dezvoltarea unor tehnici cat mai simple de utilizat pentru clientii necunosctori in domeniu;
- aceasta metoda de restaurare poate fi folosita in diverse aplicatii (cresterea gradului de Ex.: restaurarea unor poze neclare datorita conditiilor neprietenice de captare) iar prin imbunatatirii ale tehnicii folosite, poate fi folosita si in cadrul unor institutii(Ex.: in politie, la formarea unui portret robot sau lasporirea claritatii anumitor imagini );

### BIBLIOGRAFIE

- 1) BRAD, Remus, *Procesarea imaginilor și elemente de computer vision*, Editura Universității Lucian Blaga, Sibiu, 2003
- 2) Introducere in Image Deblurring,  
<http://blogs.mathworks.com/steve/2007/08/13/image-deblurring-introduction/>,  
<http://www.mathworks.com/access/helpdesk/help/toolbox/images/f9-6010.html>
- 3) Algoritmi folositi pentru a corecta denaturatile aparute in imagini cat si gradul de eficienta a acestora fata de alti algoritmi:  
<http://www.maths.usyd.edu.au/u/pubs/publist/preprints/2004/donoho-31.html>
- 4)Curs de prelucrare si analiza a imaginilor  
[http://www.studentie.ro/Curs\\_Prelucrarea\\_si\\_analiza\\_imaginilor\\_INTRODUCERE\\_ELEMENTE\\_DE\\_MORFOLOGIE\\_MATEMATICA--citeste-nr588.html](http://www.studentie.ro/Curs_Prelucrarea_si_analiza_imaginilor_INTRODUCERE_ELEMENTE_DE_MORFOLOGIE_MATEMATICA--citeste-nr588.html)

**Valentina Lazar**  
**Universitate „Lucian Blaga”**  
**Facultate de Stiinte**  
**Secția Informatica**  
**E-mail: lyah\_valy@yahoo.ro**

## Sistem de Fișiere Multimedia

**Autori:** Tudor MIU, **Universitatea** București  
Sorin LACRIȚEANU, **Universitatea** Politehnica București,  
**Coordonator:** Alexandru RADOVICI, **Universitatea** Politehnica București

*Abstract:* Prin această lucrare, autorii își propun să expună o modalitate de a facilita accesul la metadate fișierelor multimedia prin organizarea lor într-un sistem de fișiere virtual folosind platforma FUSE (Filesystem in Userspace). Odată implementat sistemul de fișiere, extragerea și creerea metadatelor de către o aplicație este posibilă prin operații specifice unui sistem de fișiere real, prin interfața oferită de sistemul de operare gazdă.

*Cuvinte cheie:* sistem de fișiere, multimedia, FUSE, Linux

### 1 Introducere

În zilele noastre există o cerere foarte mare de sisteme multimedia, cerere care acoperă un spectru larg de doritori și consumatori – de la sisteme de home entertainment, la aplicații integrate în servicii comerciale, la proiecte de cercetare științifică. Industria și cercetarea au făcut posibilă redarea conținutului multimedia de o înaltă calitate și în condiții optime, însă un aspect poate la fel de important ca și cele enumerate mai sus nu a fost exploatat și îmbunătățit în aceeași măsură - relevanța conținutului multimedia. Această relevanță este defapt informație extrasă și prelucrată din metadatele fișierelor multimedia.

Odată cu accesul tot mai puțin îngădit la resurse multimedia ale unui număr tot mai mare de consumatori, mijloacele tehnologice de redare și interfețele de interacțiune cu utilizatorul, ba chiar și sistemele de generare de conținut multimedia devin tot mai sofisticate în încercarea de a ține pasul cu ingerințele pieței. Într-un astfel de mediu deosebit de competitiv pentru cei care participă în dezvoltarea canalelor de livrare multimedia, este poate puțin surprinzător faptul că ceea ce asigură evoluția acest fenomen este uriașul volum de conținut multimedia, sub formă de muzică, filme, poze, etc. aflat în expansiune perpetuă.

Într-un sistem de operare (SO), precum Linux de pildă, informația persistentă este organizată în fișiere care la rândul lor sunt organizate în directoroare. Sistemul de fișiere este astfel format dintr-o structură ierarhică. În această structură se regăsesc și fișierele multimedia, însă această organizare primară oferită este greu de exploatat atât de către un utilizator neinstruit în domeniul calculatoarelor cât și de către un program care, de exemplu, ar încerca să construiască o structură de date pentru un *playlist*. Dificultatea constă atât în modul de organizare generic cât și în faptul că metadatele din fișiere nu sunt direct accesibile, fiind nevoie de biblioteci sau programe speciale. Mai mult decât atât, eroarea umană poate face ca metadatele unui fișier să fie în contradicție cu locul în sistemul de fișiere nativ, de exemplu, un fișier audio în ale cărui metadate artistul este Madonna se găsește în directorul Beatles.

### 2 Formularea Problemei

O experiență multimedia plăcută începe cu existența unei organizări a conținutului multimedia, de exemplu, existența unor playlist-uri pentru diferite ocazii, selectarea unui film pe baza genului, vizionarea pozelor care aparțin unui eveniment, etc. Orice aplicație orientată spre multimedia "serioasă" are funcționalități în acest sens, iar toate aceste aplicații având o caracteristică în comun: criteriile de organizare

sunt cele date de metadatele extrase din fișierele multimedia. Accesul la aceste metadate este realizat diferit, probabil în baza unor biblioteci de funcții diferite de fiecare din aceste aplicații, iar acest proces s-a dovedit a fi atât de dificil de uniformizat, încât subiectul este unul demn de aria cercetărilor științifice.

În această lucrare, autorii și-au propus să construiască un sistem de fișiere virtual care, pe baza metadatelor fișierelor multimedia, să pună la dispoziție o interfață specială de acces la conținutul multimedia în cadrul sistemului de operare. Comportamental, această interfață este identică cu interfața obișnuită a sistemului de fișiere ceea ce duce la apariția unui suport generic pentru organizarea materialelor multimedia și explorarea metadatelor acestora.

Rațiunea acestei abordări este următoarea: de ce fiecare aplicație orientată pe multimedia să-și implementeze funcții de extragere de metadate când această funcționalitate poate fi introdusă direct în sistemul de operare re folosind logica sistemului de fișiere? Dacă în sistemul de fișiere Linux s-au integrat deja interfețe pentru funcționalități comune (dispozitive bloc – CD ROM, DVD ROM, dispozitive caracter – portul serial, tastatura, mouse-ul), atunci de ce să nu se procedeze la fel și cu metadatele fișierelor multimedia, entități chiar mai des utilizate ca CD ROM-ul sau portul serial, de exemplu, de multi utilizatori? De asemenea, de ce să fie necesară folosirea unor programe speciale pentru organizarea bibliotecilor media când un sistem de fișiere ar putea face acest lucru în mod transparent? Astfel o multitudine de programe ar avea acces la aceeași bibliotecă, fără a fi necesari conectori de compatibilitate.

### 3 Platforme Software

În alegerea instrumentelor și a librăriilor pentru dezvoltarea unui sistem de fișiere am avut în vedere următoarele criterii: ușurința de dezvoltare, scalabilitate, viteza de execuție, stabilitate și sistemul de licențiere. Întrucât unele dintre criterii sunt puternic necorelate – viteza de execuție cu stabilitatea sau ușurința de dezvoltare cu sistemul de licențiere (resursele Open Source sunt notorii pentru dificultatea cu care se pot folosi) – opțiunile s-au bazat și pe anumite compromisuri. Pentru stocarea metadatelor într-o bază de date am folosit MySQL, unul din cele mai întâlnite sisteme în proiectele Open Source. Pentru interfațarea cu nucleul sistemului de operare am ales să folosim FUSE care pune la dispoziție un driver și o bibliotecă scrisă în C, ceea ce ne-a restrâns opțiunea limbajului de programare la unul singur. Sistemul de operare gazdă trebuia să fie compatibil POSIX, pentru care FUSE a dat cele mai bune rezultate, așadar opțiunea evidentă a fost Linux.

#### 2.1 C++

Am ales limbajul de programare C++, deoarece toate celelalte componente din proiect au suport excelent pentru acest limbaj. API-ul (Application Programming Interface) de FUSE este în C, iar Linux pune la dispoziție o întreagă suită de dezvoltare pentru C: editoare de text cu syntax highlighting, compilatorul gcc, debugger-ul dbg și numeroase biblioteci pentru majoritatea standardelor.

Deși viteza de rulare a unui program scris în C este, de cele mai multe ori, mai mare decât a acelui program scris în alte limbaje de nivel înalt, nu se poate avea un bun control asupra stabilității datorită faptului că multe erori logice de programare nu pot fi depistate de instrumentele de dezvoltare. Asta conduce la apariția unor bug-uri de la cele critice la cele extrem subtile.

#### 2.2 FUSE (Filesystem in Userspace)

Platforma FUSE permite crearea unui sistem de fișiere în spațiu utilizator (userspace). Prin instalarea unui driver FUSE în kernel-ul de Linux este posibilă redirectionarea apelurilor de sistem cauzate de acțiuni asupra sistemului de fișiere virtual. Dacă în cadrul unui proces se solicită o operație pe un fișier, procesul este suspendat și se trece la tratarea cererii în nucleul SO-ului. Responsabil pentru tratarea operațiilor de fișiere este modulul VFS (Virtual Filesystem Switch) care, după ce face verificări asupra drepturilor de acces, lansează în execuție driver-ul responsabil pentru sistemul de fișiere de unde provine cererea. Dacă apelul inițial a fost făcut pentru sistemul virtual de fișiere, driver-ul FUSE trimite mai departe evenimentul în userspace procesului asociat

unde evenimentul este tratat. În cele din urmă execuția revine kernel-ului care deblochează procesul care a inițiat cererea.

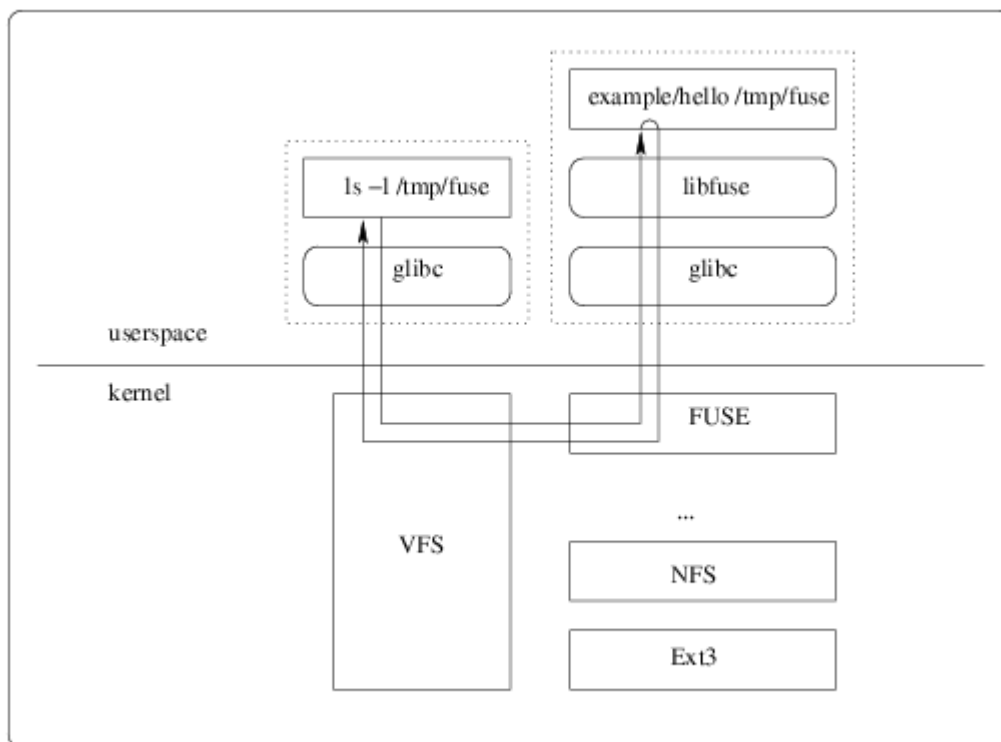


Fig. 1 – Schema de funcționare a platformei FUSE

## 2.2 Linux

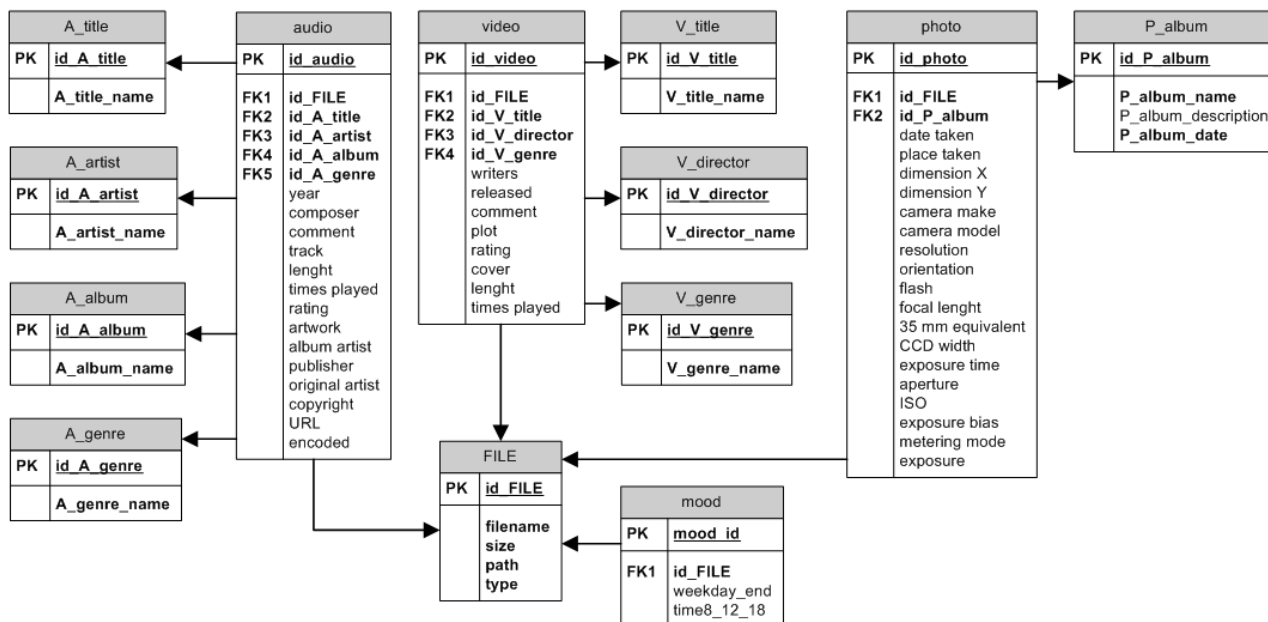
Sistemul de operare (SO) ales pentru dezvoltarea sistemului de fișiere a fost Linux, motivul cel mai important fiind acela că FUSE și-a dovedit stabilitatea sub acest SO, atât pentru kernel-urile 2.4.\* cât și pentru 2.6.\*. Fiind o clonă de UNIX, Linux are un sistem de fișiere care, deși nu este unul intuitiv pentru utilizatorii obișnuiți, are o flexibilitate remarcabilă în ceea ce privește integrarea interfețelor pentru dispozitive și crearea punctelor de montare (mounting points) pentru alte sisteme de fișiere. Prin instalarea driver-ului FUSE, se pune la dispoziție o interfață programabilă de augmentare a sistemului de fișiere în userspace.

La pornirea SO, este lansat în execuție un proces de fundal care scanează toate fișierele de pe harddisk, în căutarea metadatelor din fișierele multimedia. Prin operații de citire, sunt apoi trimise comenzi către daemon-ul MySQL care stochează metadatele. După ce operația se încheie, procesul de scanare este suspendat, urmând să fie repornit îndată ce se montează un alt sistem fișiere, eveniment ce are loc, de exemplu, la introducerea unui stick USB în calculator sau al oricărui alt dispozitiv care stochează fișiere.

## 2.3 Baza de date și MySQL

MySQL are avantajul că este un sistem de gestiune al bazelor de date tranzacțional. De asemenea, prin extensiile pe care le aduce în plus față de standardul SQL ANSI 99, facilitează crearea și gestiunea unor baze de date cu grad ridicat de complexitate și flexibilitate. În plus, suportul pentru arhitecturi client-server îl face atractiv dacă sistemul de fișiere va fi extins la nivel de rețea într-un sistem distribuit.

În cadrul sistemului software, MySQL primește metadatele extrase de procesul de scanare și le organizează într-o bază de date. Structura de bază a sistemului de fișiere construit prin FUSE este dată de relațiile dintre tabele, iar popularea cu fișiere și subdirectoare se realizează în corespondență directă cu conținutul din baza de date.



Baza de date reprezintă back-end-ul sistemului software, entitățile modelate regăsindu-se, prin analogie, în structura ierarhica a sistemului de fișiere.

## 5 Organizarea Sistemului de Fișiere Multimedia

Structura sistemului de fișiere relativă la mounting point-ul din sistemul de fișiere nativ al SO-ului, este următoarea:

```

/audio
/audio/music
/audio/music/artists
/audio/music/albums
/audio/music/genres
/audio/music/playlists
/pics
/video
/video/movies
/video/videoclips
/video/videoclips/artists
/video/genres

```

Popularea acestor directoare cu subdirectoare și fișiere se face după reguli în analogie cu structura bazei de date. De exemplu, pentru fiecare artist se creează un director nou în audio/music/artists, pentru fiecare album se creează un director nou atât în subdirectorul asociat din /audio/music/artists/ cât și în /audio/music/albums, pentru fiecare piesă nouă se ”copiază”



fișierul în /audio/music/albums și /audio/music/genres, iar pentru fiecare playlist, se ”copiază” fișierul în /audio/music/playlists .

Copierea fișierelor nu se face în sensul dat de sistemul nativ de fișiere, ci este guvernat de procesul asociat driver-ului FUSE. Acesta a fost programat să creeze *soft links* (ln -s) către fișierele de pe sistemul de fișiere al mediului de stocare real.

## 6 Avantaje

Avantajele principale al acestei abordări sunt ușurința cu care se pun metadatele la dispoziția aplicațiilor și structurarea fișierelor audio, video și a fotografiilor pentru a fi ușor de accesat de programe multimedia. Multe din operațiile tipice pentru librăriile media nu mai trebuie implementate de la zero, ele putând fi înlocuite prin comenzi sau scripturi shell. De pildă, listarea tuturor melodiilor aparținând genului muzical rock se poate face în felul următor: ls /genres/rock .

## 7. Concluzii și dezvoltări viitoare

Pornind de la necesitatea unei modalități generice de a organiza bibliotecile multimedia, am dezvoltat o platformă generică incorporată în sistemul de fișiere Linux. Un proces pornit în secvența de start-up scanează harddisk-ul și, folosind ca back-end un motor MySQL, extrage metadatele din fișierele multimedia și organizează metadatele într-o bază de date. Driver-ul FUSE folosește interfața sistemului de fișiere și îi extinde funcționalitatea, permițând crearea unui sistem de fișiere virtual în userspace unde, folosind metadatele colectate, modelează o librărie media. Mai departe, acest model servește ca suport de programare pentru aplicații multimedia specifice.

În dezvoltările ulterioare urmărim să extindem sistemul de fișiere de la unul local la unul distribuit pe rețea, folosind aceleași tehnologii . În plus, intenționăm să dezvoltăm propriul nostru sistem Linux orientat pe multimedia, pornind de la zero (Linux From Scratch) în care să includem atât un sistem de fișiere multimedia, cât și propriile programe multimedia, dar și interfață grafică bazată pe Adobe Flex.

## Bibliografie

1. <http://code.google.com/p/simplefuse/>
2. Filesystem in userspace <http://fuse.sourceforge.net/>
3. Linux <http://www.linux.ro/>
4. Linux <http://www.linux.org/>
5. Beekmans, G. ; *Linux From Scratch 6.4*, <http://www.linuxfromscratch.org/lfs/>
6. Sobell, M.; *A Practical Guide to Linux Commands, Editors, and Shell Programming*, Prentice Hall, 2005
7. Sklyarov I. , *Programming Linux Hacker Tools Uncovered*, Independent Pub Group, 2006
8. Matthew N. , Stones R. , *Beginning Linux Programming*, 4ed, Wrox, 2007
9. Kroah-Hartman, G. , *Linux Kernel in a Nutshell*, O'Reilly & Associates, 2006
10. Bovet Daniel P. , *Understanding The Linux Kernel*, O'Reilly & Associates , 2003
11. Mauerer W. , *Professional Linux Kernel Architecture*, Wrox, 2008
12. Tannenbaum, A.; *Sisteme de operare moderne*, Ediția a II-a, Byblos, București, 2004
13. Tannebaum, A.; Woodhull, A, *Operating Systems: Design and Implementation*, Prentice Hall, 2006
14. Love, R. ; *Linux System Programming*, O'Reilly & Associates, 2007
15. MySQL <http://www.mysql.com/>
16. Stephens, J. ; Russell C. ; *Beginning MySQL Database Design and Optimization*, Apress, 2004
17. Site-ul cursului de Sisteme de Operare al Facultății de Automatică și Calculatoare, Universitatea Politehnica Bucuresti <http://cs.pub.ro/~so/>

18. Site-ul cursului de Sisteme de Operare 2 al Facultății de Automatică și Calculatoare, Universitatea Politehnica Bucuresti <http://cs.pub.ro/~ps0/>
19. Site-ul cursului de Baze de Date al Facultății de Automatică și Calculatoare, Universitatea Politehnica Bucuresti <http://www.bazededate.org/>
20. Miu T. , Stanescu O. , Constantin A. , Lacriteanu S. , Grigore R. , Burca D. , Constantinescu T. , Radovici A. , *Media Center oriented Linux Operating System*, CEUR Workshop Proceedings, Toulouse, 2009, pag. 162
21. Radovici A. , Apostol E. A. , *Distributed Hybrid-Storage Partially Mountable File System*, Proceedings of Recent Advances in Artificial Intelligence, Knowledge Engineering and Databases, 8ed, Cambridge 2009, pag. 35

**Tudor Miu**  
**Universitatea Politehnica București**  
**Facultatea de Inginerie în Limbi Străine**  
**Calculatoare și Tehnologia Informației**  
tudor.miu@gmail.com

**Sorin Lacrițeanu**  
**Universitatea Politehnica București**  
**Facultatea de Inginerie în Limbi Străine**  
**Calculatoare și Tehnologia Informației**  
sorin\_lacriteanu@yahoo.fr

**Alexandru Radovici**  
**Universitatea Politehnica București**  
**Facultatea de Inginerie în Limbi Străine**  
**Catedra de Inginerie în Limbi Străine**  
msg4alex@gmail.com

# Matlab - COM Client Support - Exemples of Matlab as an Automation Client

**Author:** NITU RAMONA MARINELA

**Coordinator:** Assoc. Prof. Dana Simian

**University** “Lucian Blaga”, Sibiu, Faculty of Sciences

**Abstract:** *In this paper we propose to show how Matlab COM Client can access applications that support Automation, such as Excel and how we can use Matlab as an Automation Client and the Microsoft Excel as an out-of-process server. To do this we took some examples and after that we changed the strategy proposed in “Matlab – Documentation” according to our project, especially for saving time.*

## 1. COM

The Component Object Model (COM) provides a framework for integrating reusable, binary software components into an application. Because components are implemented with compiled code, the source code can be written in any of the many programming languages that support COM. Upgrades to applications are simplified, as components can simply be swapped without the need to recompile the entire application. In addition, a component's location is transparent to the application, so components can be relocated to a separate process or even a remote system without having to modify the application. Using COM, developers and end users can select application-specific components produced by different vendors and integrate them into a complete application solution. For example, a single application might require database access, mathematical analysis, and presentation-quality business graphs. Using COM, a developer can choose a database-access component by one vendor, a business graph component by another, and integrate these into a mathematical analysis package produced by yet a third. MATLAB supports COM integration on the Microsoft Windows platform.

### 1.1 Objects, Clients, and Servers

A COM object is a software component that conforms to Microsoft's Component Object Model. COM enforces encapsulation of the object, preventing direct access of its data and implementation. COM objects expose Interfaces, which consist of properties, methods and events. A COM client is a program that makes use of COM objects. COM objects that expose functionality for use are called COM servers. COM servers can be in-process or out-of-process. Examples of out-of-process servers are Microsoft Excel and Microsoft Word programs.

### 1.2 Interfaces

The functionality of a component is defined by one or more interfaces. To use a COM component, you must learn about its interfaces, and the methods, properties and events implemented by the component. The component vendor provides this information. There are two standard COM interfaces: *IUnknown* — an interface required by all COM components. All other COM interfaces are derived from *Iunknown*. *IDispatch* — an interface that exposes objects, methods and properties to applications that support Automation (Automation — A server that supports the OLE Automation standard. Automation servers are based on the *IDispatch* interface. Automation servers can be accessed by clients of all types, including scripting clients.)

### 1.3 In-Process and Out-of-Process Servers

You can configure a server three ways. MATLAB supports all of these configurations.  
*In-Process Server*

An in-process server is a component implemented as a dynamic link library (DLL) or ActiveX control that runs in the same process as the client application, sharing the same address space. Communication between client and server is relatively fast and simple.

#### *Local Out-of-Process Server*

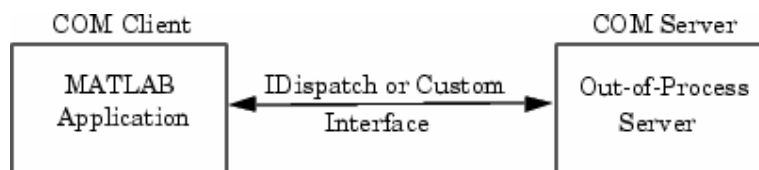
A local out-of-process server is a component implemented as an executable (EXE) file that runs in a separate process from the client application. The client and server processes are on the same computer system. This configuration is somewhat slower due to the overhead required when transferring data across process boundaries.

#### *Remote Out-of-Process Server*

This is a type of out-of-process server; however, the client and server processes are on different systems and communicate over a network. Network communications, in addition to the overhead required for data transfer, can make this configuration slower than the local out-of-process configuration. This configuration runs only on systems that support the Distributed Component Object Model (DCOM).

## 1.4 The MATLAB COM Client

Using MATLAB as a COM client provides two techniques for developing programs in MATLAB: you can include COM components in your MATLAB application (for example, a spreadsheet) and you can access existing applications that expose objects via Automation. In a typical scenario, MATLAB creates ActiveX controls in figure windows, which are manipulated by MATLAB through the controls' properties, methods, and events. This is useful because there exists a wide variety of graphical user interface components implemented as ActiveX controls. For example, Internet Explorer exposes objects that you can include in a figure to display an HTML file. There also are treeviews, spreadsheets and calendars available from a variety of sources. MATLAB COM clients can access applications that support Automation, such as Excel. In this case, MATLAB creates an Automation server in which to run the application and returns a handle to the primary interface for the object created.



## 2. Examples of MATLAB Software as an Automation Client

You can configure MATLAB software to either control or be controlled by other COM components. When MATLAB controls another component, MATLAB is the client, and the other component is the server. When another component controls MATLAB, MATLAB is the server.

### • Example 1

This example uses MATLAB as an Automation client and Microsoft Excel as the server. It provides a good overview of typical functions. In addition, it is a good example of using the Automation interface of another application:

```
% MATLAB Automation Client example; Open Excel, add workbook, change active  
% worksheet, get/put array, save.  
% First, open an Excel Server.  
e = actxserver('excel.application');
```

```

% Insert a new workbook.
eWorkbook = e.Workbooks.Add;
e.Visible = 1;

% Make the first sheet active.
eSheets = e.ActiveWorkbook.Sheets;
eSheet1 = eSheets.get('Item', 1);
eSheet1.Activate;

% Put a MATLAB array into Excel.
A = [1 2; 3 4];
eActivesheetRange = e.Activesheet.get('Range', 'A1:B2');
eActivesheetRange.Value = A;

% Get back a range. It will be a cell array, since the cell range can contain different types of % data.
eRange = e.Activesheet.get('Range', 'A1:B2');
B = eRange.Value;

% Convert to a double matrix. The cell array must contain only scalars.
B = reshape([B{:}], size(B));

% Now, save the workbook.
eWorkbook.SaveAs('myfile.xls');

% Avoid saving the workbook and being prompted to do so
eWorkbook.Saved = 1;
eWorkbook.Close;

% Quit Excel and delete the server.
e.Quit;
e.delete;

```

### **Note !**

Make sure that you always close any workbooks that you add in Excel. This can prevent potential memory leaks.

### **● Exemple 2**

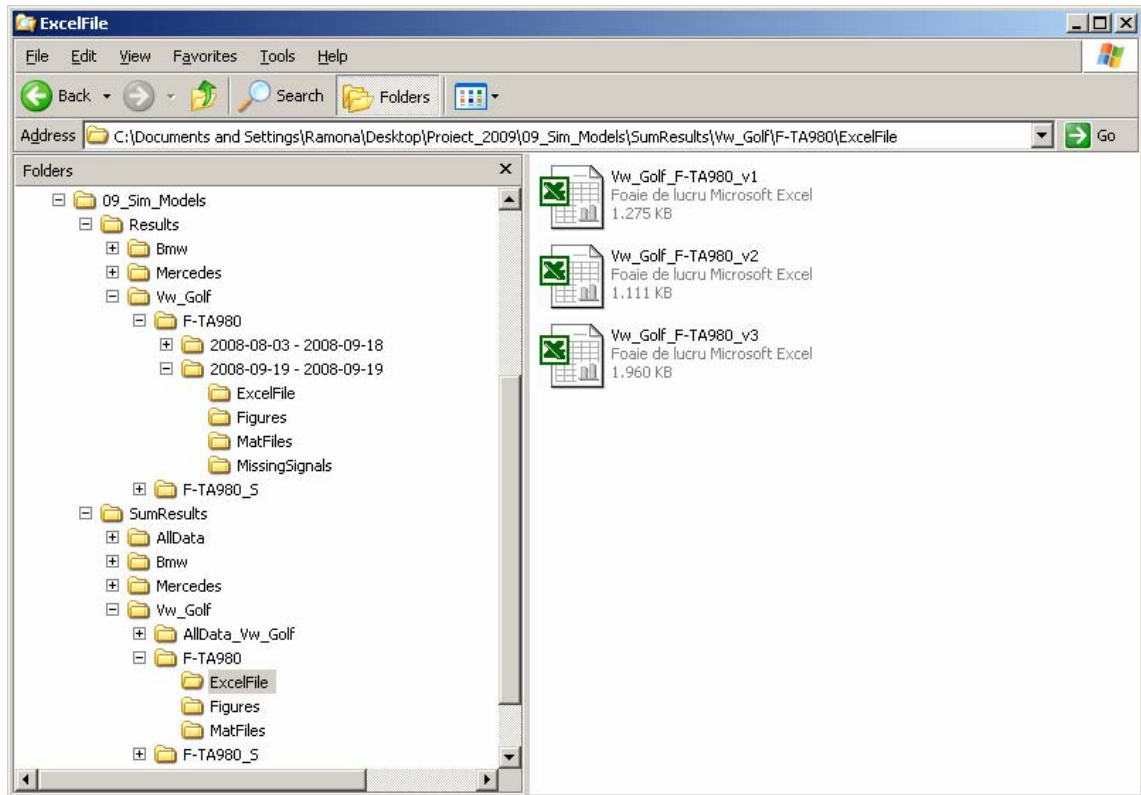
Now I present some parts of my project. I've made a structure for saving data after running the application like this:

```

[path_for_Figures,path_for_Excel,path_for_mat_files,path_for_MisingS
ignals,FolderNameForSet] =
SavingStructure_modif(name_directory,auto_type,
registration_number,save_path);

```

Data are saved into two folders named *Results* and *SumResults* which are part of main folder named *09\_Sim\_Models*.



Here I'm writing the functions for exporting computed data into excel file !

```

% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% %                               WRITING DATA TO EXCEL                               % %
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pat_with_file_name_version =
save_data_statistics_excel(pat_for_Excel,auto_type,registration_number);
name_xls_file = pat_with_file_name_version;

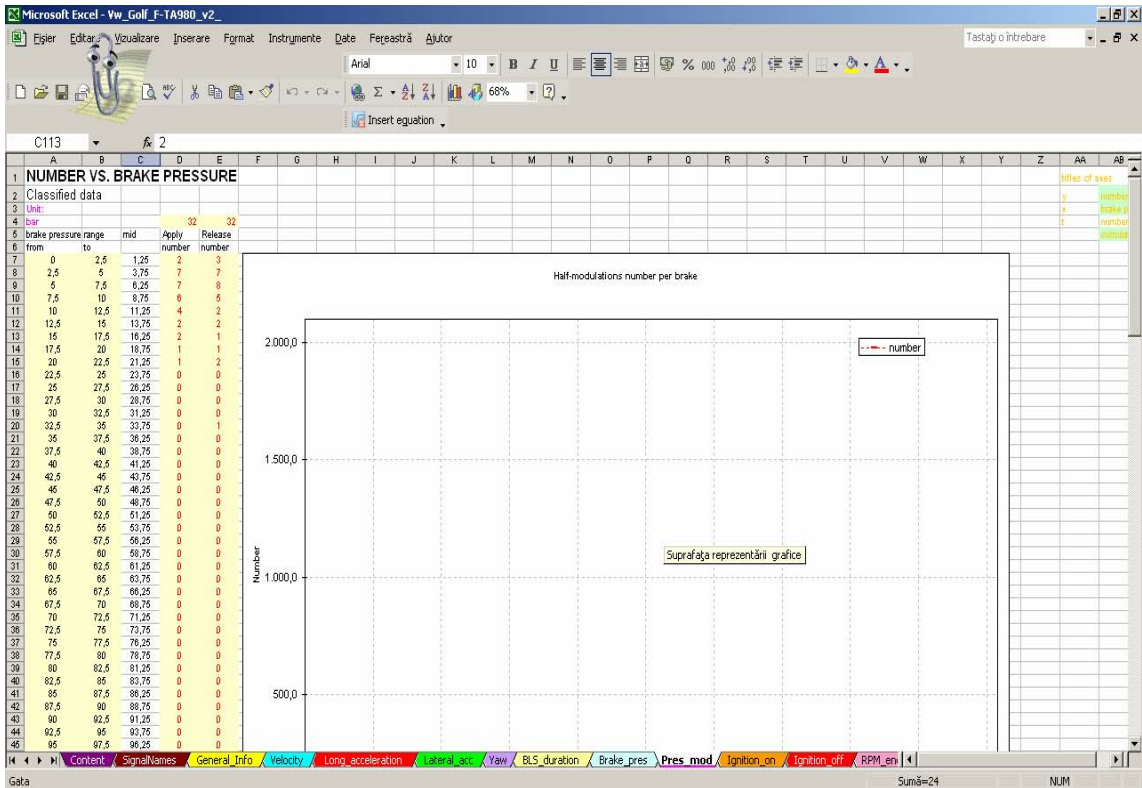
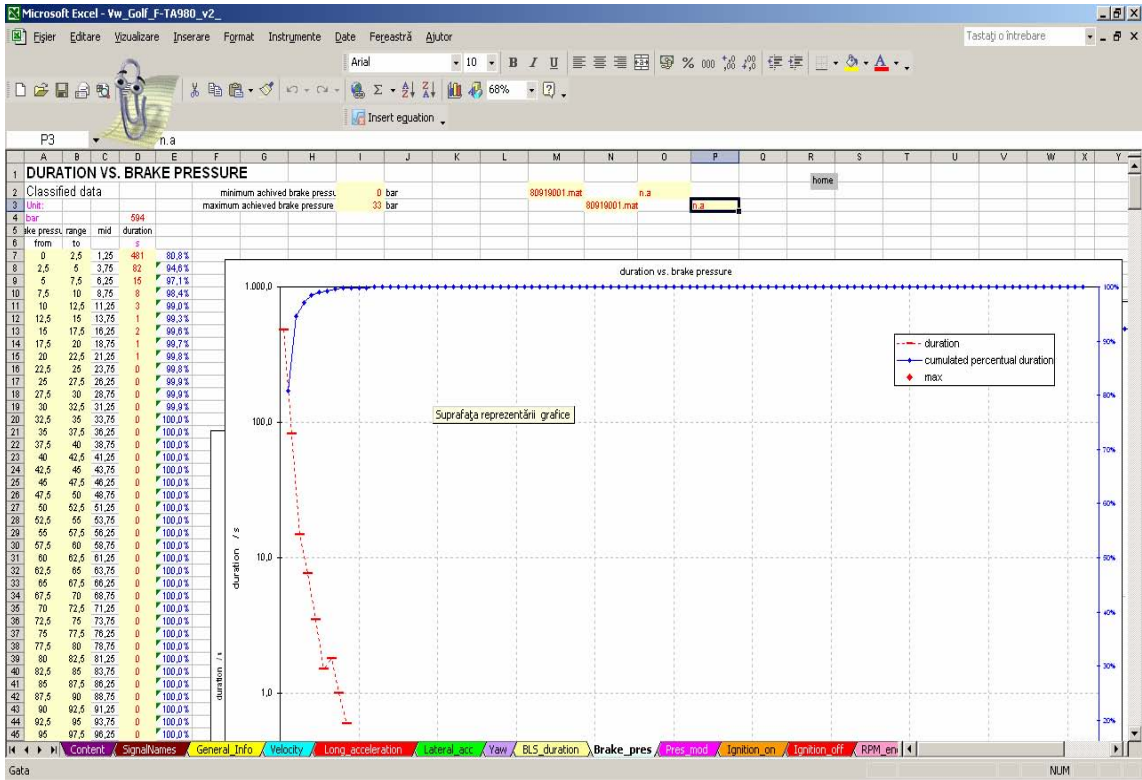
Excel = actxserver ('Excel.Application');
%invoke(Excel.Workbooks,'Open',name_xls_file);
excel_file_wk = Excel.Workbooks;
excelFile = excel_file_wk.Open(name_xls_file);

% % Write brake pressure data into the sheets "Brake_pres" & "Pres_mod"
writel_vectors_excel(Excel,name_xls_file,'Brake_pres',vec_press,bin_duratio
n_bp,min_max_bp,mat_for_min_bp,mat_for_max_bp,time_for_min_bp,...
time_for_max_bp);
if (isempty(delta_p_up) && isempty(delta_p_down) &&
isempty(no_up_modulations) && isempty(no_down_modulations))
    xlswrite2(name_xls_file,{'not available'},'Pres_mod','H1');
else
    xlswrite2(name_xls_file,n_p_up,'Pres_mod','D7');
    xlswrite2(name_xls_file,n_p_down,'Pres_mod','E7');

xlswrite2(name_xls_file,reshape(n_up_mod,length(n_up_mod),1),'Pres_mod','C1
13');
end;

```

Here are the results from Excel File !



```

% % Write data for 3D diagram into the sheet: "Speed_RPM"
write_data_toexcel_for_3d_diag(path_for_Figures,excelFile,Excel,name_xls_file,auto_type,'Speed_RPM',hist6,1:38,1:57,1:4:57,1:2.5:38,-25:20:255,0:500:7400,[1 57],[1 38],0.5,...
    [-37 30],0:200:7400,-25:5:255,'Speed [km/h]','Engine RPM[RPM]',...
    'DURATION VS. VELOCITY AND ENGINE RPM','Total duration','A3','I10',0.9,'B68','A70','B70','B67');

```

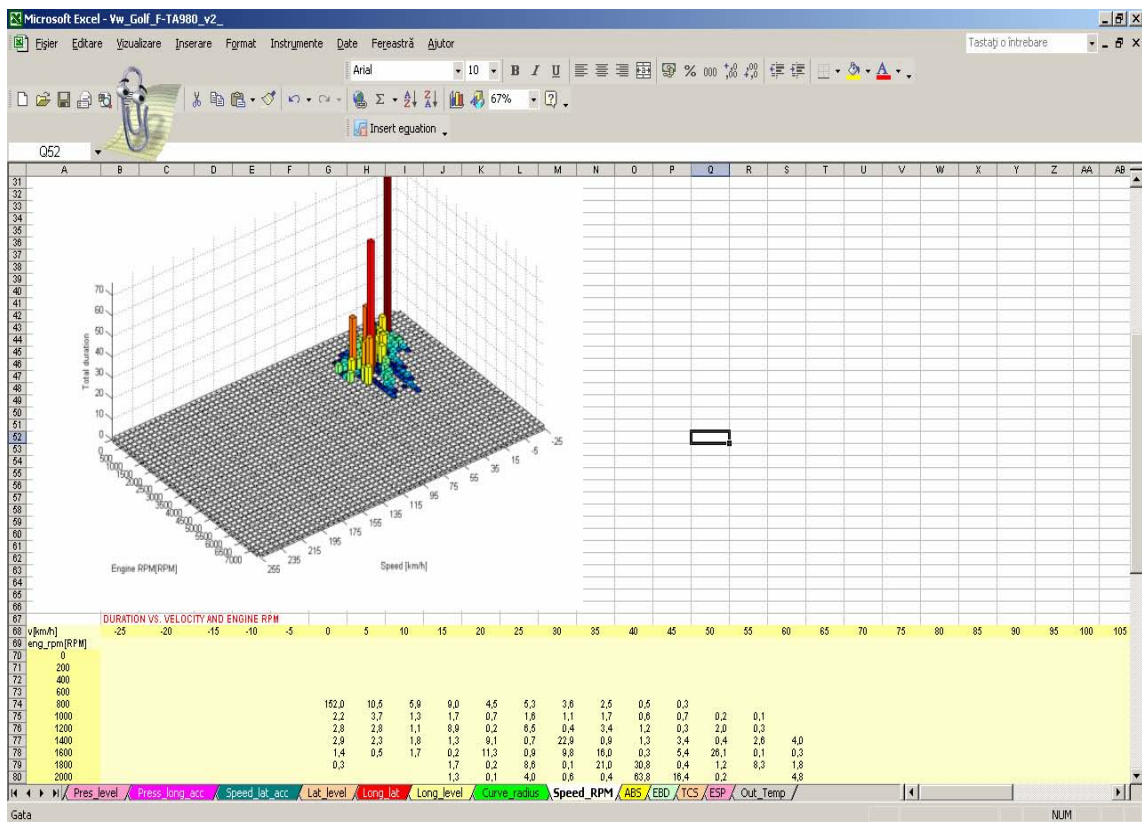
```

invoke(Excel.ActiveWorkbook,'Save');
Excel.Quit
Excel.delete
clear Excel

```

```
winopen(name_xls_file);
```

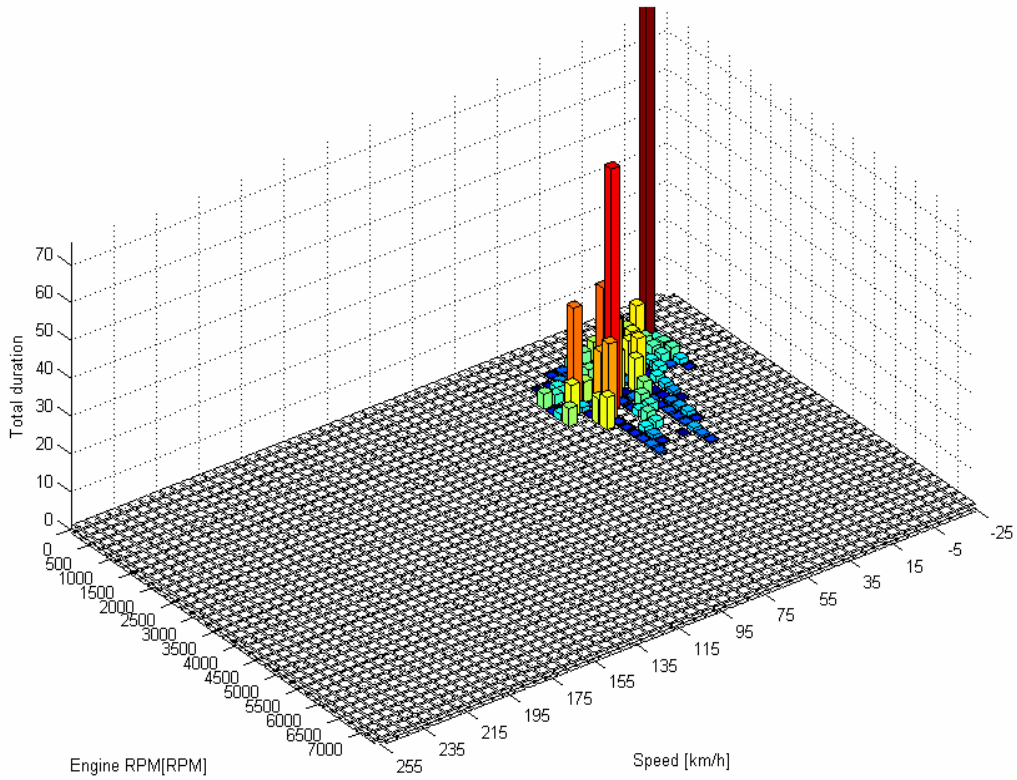
Here are the results from Excel File !



And if you want to understand what it's going on above, see next:



DURATION VS. VELOCITY AND ENGINE RPM



Microsoft Excel - Vw\_Golf\_F-TA980\_v2

U77

DURATION VS. VELOCITY AND ENGINE RPM

| v[km/h]      | -25 | -20 | -15 | -10 | -5 | 0 | 5     | 10   | 15  | 20  | 25   | 30  | 35   | 40   | 45   | 50   | 55   | 60  | 65  | 70 |  |
|--------------|-----|-----|-----|-----|----|---|-------|------|-----|-----|------|-----|------|------|------|------|------|-----|-----|----|--|
| eng_rpm[RPM] |     |     |     |     |    |   |       |      |     |     |      |     |      |      |      |      |      |     |     |    |  |
| 0            |     |     |     |     |    |   |       |      |     |     |      |     |      |      |      |      |      |     |     |    |  |
| 200          |     |     |     |     |    |   |       |      |     |     |      |     |      |      |      |      |      |     |     |    |  |
| 400          |     |     |     |     |    |   |       |      |     |     |      |     |      |      |      |      |      |     |     |    |  |
| 600          |     |     |     |     |    |   |       |      |     |     |      |     |      |      |      |      |      |     |     |    |  |
| 800          |     |     |     |     |    |   | 152,0 | 10,5 | 5,9 | 9,0 | 4,5  | 5,3 | 3,6  | 2,5  | 0,5  | 0,3  |      |     |     |    |  |
| 1000         |     |     |     |     |    |   | 2,2   | 3,7  | 1,3 | 1,7 | 0,7  | 1,6 | 1,1  | 1,7  | 0,6  | 0,7  | 0,2  | 0,1 |     |    |  |
| 1200         |     |     |     |     |    |   | 2,8   | 2,8  | 1,1 | 8,9 | 0,2  | 6,5 | 0,4  | 3,4  | 1,2  | 0,3  | 2,0  | 0,3 |     |    |  |
| 1400         |     |     |     |     |    |   | 2,9   | 2,3  | 1,8 | 1,3 | 9,1  | 0,7 | 22,9 | 0,9  | 1,3  | 3,4  | 0,4  | 2,6 | 4,0 |    |  |
| 1600         |     |     |     |     |    |   | 1,4   | 0,5  | 1,7 | 0,2 | 11,3 | 0,9 | 9,8  | 16,0 | 0,3  | 5,4  | 26,1 | 0,1 | 0,3 |    |  |
| 1800         |     |     |     |     |    |   | 0,3   |      |     | 1,7 | 0,2  | 8,6 | 0,1  | 21,0 | 30,8 | 0,4  | 1,2  | 8,3 | 1,8 |    |  |
| 2000         |     |     |     |     |    |   |       |      |     | 1,3 | 0,1  | 4,0 | 0,6  | 0,4  | 63,8 | 16,4 | 0,2  | 4,8 |     |    |  |
| 2200         |     |     |     |     |    |   |       |      |     | 0,8 | 0,2  | 0,5 | 3,3  | 0,3  |      | 20,1 | 6,7  |     |     |    |  |
| 2400         |     |     |     |     |    |   |       |      |     | 0,1 | 0,6  | 0,3 | 2,3  | 1,8  | 0,3  | 8,5  |      |     |     |    |  |
| 2600         |     |     |     |     |    |   |       |      |     |     | 0,9  | 0,5 |      | 2,8  | 0,3  |      |      |     |     |    |  |
| 2800         |     |     |     |     |    |   |       |      |     |     | 1,1  | 0,4 |      | 1,9  | 1,5  | 0,3  |      |     |     |    |  |
| 3000         |     |     |     |     |    |   |       |      |     |     | 0,3  | 0,5 |      |      | 0,9  | 0,4  |      |     |     |    |  |
| 3200         |     |     |     |     |    |   |       |      |     |     |      | 0,5 | 0,1  |      | 0,3  | 0,6  |      |     |     |    |  |
| 3400         |     |     |     |     |    |   |       |      |     |     |      | 0,6 |      |      | 0,5  |      |      |     |     |    |  |
| 3600         |     |     |     |     |    |   |       |      |     |     |      | 0,4 |      |      |      |      |      |     |     |    |  |
| 3800         |     |     |     |     |    |   |       |      |     |     |      |     |      |      |      |      |      |     |     |    |  |

Gata NUM

### **3 Bibliography**

[http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/matlab\\_external/brd4at8.html](http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/matlab_external/brd4at8.html)

[http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/matlab\\_external/brd05nl.html](http://www.mathworks.com/access/helpdesk/help/techdoc/index.html?/access/helpdesk/help/techdoc/matlab_external/brd05nl.html)

## Metode de procesare a textului pentru sinteză vocală

**Autor:** Stancu Mihai, Facultatea de Științe specializarea matematică-informatică  
**Conf. Univ. Dr. Dana Simian, Universitatea "Lucian Blaga" din Sibiu**

*Abstract: The paper presents different new methods of processing text for speech synthesis applications. The methods we present are suitable for the Romanian language, but can be applied to many other languages as well. In order to illustrate the concepts presented, we propose and implement in Visual C++ a TTS system for the Romanian language, based on the MBROLA (Multiband Resynthesis Overlap-Add) project.*

*Cuvinte cheie: text processing, text-to-speech system, Romanian language*

### 1. Introducere

Cea mai naturală formă de comunicare între oameni este comunicarea orală. De aceea este de mare folos utilizarea vorbirii naturale pentru interacțiunea dintre om și calculator. În acest sens au fost create programe de recunoaștere vocală și sintetizatoare vocale bazate pe text, numite și sisteme Text-to-Speech (TTS).

La ora actuală există numeroase aplicații de sinteză vocală, dintre care menționăm numai câteva exemple: în serviciile de telecomunicații sinteza vocală poate înlocui vocea operatorului, mesajele scurte putând fi ușor redade de către un sintetizator. În transportul public, sinteza vocală este folosită pentru a informa pasagerii despre sosiri/plecări, sau pentru a furniza informații importante. O altă aplicație a sintezei vocale o reprezintă programele care permit utilizarea calculatorului de către persoanele nevăzătoare sau cu deficiențe de vedere. De asemenea, sinteza vocală ar putea fi folosită la învățarea unei limbi, și în educație în general, în diferite aplicații de tip e-learning. Numărul acestor din urmă aplicații este însă unul extrem de mic din cauza calității slabe a sistemelor existente.

Scopul principal al acestei lucrări este de a propune noi metode de procesare a textului pentru sinteza vocală și ilustrarea acestora cu ajutorul unui sistem TTS bazat pe proiectul MBROLA (Multiband Resynthesis Overlap-Add) pentru limba română.

Lucrarea este organizată în modul următor: secțiunea 2 conține o scurtă prezentare a procesului de sinteză vocală; secțiunea 3 descrie pe scurt sistemul MBROLA care stă la baza sistemului TTS implementat pentru ilustrarea conceptelor originale din această lucrare; secțiunea 4 prezintă în detaliu algoritmul elaborat precum și designul și implementarea sistemului TTS pentru limba română. În ultima secțiune prezentăm rezultatele și câteva posibile direcții de dezvoltare viitoare.

### 2. Procesul de sinteză vocală

Metodele de sinteză vocală pot fi împărțite în trei categorii:

1. Sinteza prin articulare: modelarea directă a sistemului de vorbire umană;
2. Sinteza prin formare: modelarea frecvențelor unor semnale de sunet;
3. Sinteza prin concatenare: folosirea unor segmente preînregistrate de vorbire naturală.

Sinteza prin articulare nu este foarte folosită în sintetizatoarele de astăzi pentru că este prea complicată pentru implementări de calitate înaltă. O dată cu dezvoltarea tehnologiei însă, aceasta va putea reprezenta o opțiune mai atractivă în viitor.

În sistemele dezvoltate în prezent cele mai folosite metode sunt sinteza prin articulare și sinteza prin concatenare. Dacă în trecut sinteza prin formare a fost cea care a dominat, astăzi metoda prin concatenare devine din ce în ce mai populară. De aceea prezentăm detaliile acestei metode.

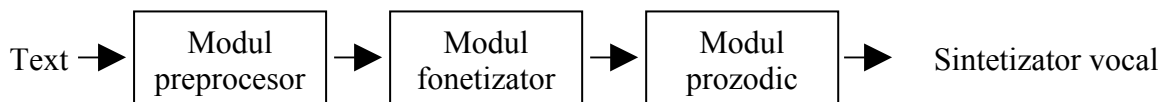
Principala problemă a sintezei prin concatenare este de a găsi lungimea corectă a unităților de vorbire ce vor fi concatenate. Unitățile de tip cuvânt sunt practice atunci când sunt pronunțate izolat. Însă o propoziție formată din astfel de unități nu sună deloc natural. De aceea majoritatea sintetizatoarelor sunt bazate pe unități mai mici precum fonemele, difonurile, demisilabele sau combinații între acestea.

Există diverse metode de sinteză prin concatenare, dintre care menționăm aici doar pe cele mai interesante:

- Metoda microfonică: folosirea unor unități de lungime variabilă preluate din înregistrări de vorbire naturală;
- Metode bazate pe predicție liniară (LP): modelul sursă-filtru, coeficienții de filtrare sunt estimați automat din înregistrări de vorbire naturală;
- Modele sinusoidale: presupunerea că semnalul de vorbire poate fi reprezentat ca o sumă de unde sinusoidale cu amplitudini și frecvențe care variază în funcție de timp.
- Metoda PSOLA: o metodă foarte populară, care permite concatenarea fără întreruperi a unor înregistrări de vorbire naturală și furnizează un control foarte bun al înălțimii și duratei sunetelor; este folosită în sisteme sintetizatoare precum ProVerbe, HADIFIX, MBROLA (sistemul care este folosit în această lucrare), etc.

### 3. Metodele de procesare a textului

Algoritmul funcționează astfel: textul primit ca date de intrare (fișier text sau text introdus de la tastatură) este mai întâi procesat de către modulul preprocesor pentru a identifica abrevierile, numerele arabe sau romane, datele calendaristice, orele, diferite simboluri, etc. pe care le înlocuiește cu textul corespunzător. În al doilea pas textul rezultat este fonetizat de către modulul fonetizator într-o secvență de foneme. Al treilea pas constă în atribuirea unei durate și a unei înălțimi sunetului corespunzător fiecărui fonem, prin intermediul modulului prozodic. În final este apelat sintetizatorul vocal ales pentru a realiza sinteza vocală. Aceste etape sunt reprezentate schematic în figura 1:



*Fig.1: Principiul de funcționare al algoritmului de procesare a textului: modulul preprocesor transformă abrevierile, numerele, etc. în textul corespunzător; modulul fonetizator transformă întregul text în foneme; modulul prozodic atribuie fiecărui fonem o durată și o înălțime a sunetului corespunzătoare; sintetizatorul vocal realizează sinteza vocală pe baza acestor parametri.*

### 4. Sistemul TTS pentru limba română

În această secțiune descriem designul și implementarea sistemului TTS pentru limba română folosit pentru ilustrarea conceptelor prezentate mai sus.

Sistemul TTS pentru limba română este implementat în Visual C++, conține o interfață grafică, dar prezintă și posibilitatea rulării de la linia de comandă.

Aplicația folosește câteva fișiere externe ca surse de informație. Configurația sistemului este memorată într-un fișier ce conține simbolurile SAMPA (Speech Assessment Methods Phonetic Alphabet), viteza de citire, durata pauzei dintre două cuvinte, etc. De asemenea sunt utilizate câteva dicționare precum dicționarul de reguli fonetice, dicționarul de abrevieri și dicționarele de excepții fonetice, respectiv prozodice, cu scopul de a îmbunătăți cât mai mult sinteza vocală.

### 4.1 Modulul preprocesor

Modulul preprocesor folosește dicționarul de abrevieri pentru a înlocui abrevierile sau simbolurile care pot apărea în text prin cuvintele corespunzătoare. De asemenea acest modul transformă numerele arabe sau romane, ora sau datele calendaristice pe care le întâlnește în text în numeralele corespunzătoare prin intermediul algoritmilor clasici.

### 4.2 Modulul fonteizator

Modulul fonetizator este responsabil pentru codificarea textului rezultat în urma preprocesării într-o secvență de foneme. Simbolurile folosite trebuie să fie aceleași cu simbolurile SAMPA pentru difonurile furnizate de baza de date corespunzătoare limbii române pentru sistemul MBROLA. Pentru a putea îndeplini această condiție este folosit dicționarul de reguli fonetice. De asemenea, acest modul mai folosește un dicționar ce conține o serie de cuvinte care prezintă excepții de la regulile de pronunție (majoritatea neologisme), precum și modul corect de pronunție a acestora. Astfel se poate asigura codificarea corectă a fonemelor unui număr cât mai mare de cuvinte. Dacă un cuvânt nu se găsește în acest dicționar de excepții fonetice, atunci se aplică regulile obișnuite pentru reprezentarea sa fonetică. Aceste reguli sunt relativ simple pentru limba română, și de aceea calitatea sintezei vocale pentru această limbă este superioară celor mai multe limbi străine.

### 4.3 Modulul prozodic

Modulul prozodic realizează accentuarea corectă a cuvintelor plurisilabice din text pentru a oferi, pe cât posibil, mai multă naturalețe sintezei vocale. Pentru aceasta este nevoie de aplicarea regulilor limbii române pentru despărțirea în silabe a cuvintelor, iar apoi accentuarea penultimei silabe a fiecărui cuvânt plurisilabic din text. Cuvintele care prezintă excepții de la aceste reguli sunt conținute în dicționarul de excepții prozodice. După determinarea silabei accentuate a unui cuvânt, se mărește înălțimea sunetului corespunzător vocalei din această silabă. De asemenea modulul acordă atenție deosebită diftongilor și triftongilor. Modulul prozodic mai interpretează și semnele de punctuație: în cazul virgulelor, a punctului sau a parantezelor se mărește pauza dintre cuvintele corespunzătoare; în cazul întâlnirii semnului exclamării sau al întrebării la sfârșitul unei fraze, se accentuează diferit ultimul cuvânt al acelei fraze. Nu în ultimul rând, acest modul trebuie să asigure respectarea vitezei de citire a textului și a pauzei medii dintre cuvinte, valori citite din fișierul de configurație, sau stabilite de către utilizator prin intermediul interfeței grafice. Aceasta se realizează prin modificarea corespunzătoare a duratei fiecărui fonem.

## 5. MBROLA

Sintetizatorul vocal ales pentru a realiza sinteza vocală după procesarea textului în limba română de către aplicația noastră este MBROLA.

MBROLA folosește o metodă de sintetizare vocală bazată pe sinteza prin concatenare a difonurilor. De aceea, pentru a putea rula acest sintetizator, este nevoie de o bază de date cu difonuri adaptate formatului MBROLA. Astfel de baze de date au fost create pentru 34 de limbi, printre care și limba română.

Datele de intrare sunt reprezentate de o listă de foneme împreună cu informație prozodică cum ar fi durata fonemelor și descrierea liniară a înălțimii sunetelor ce vor fi generate. De aceea sistemul nu poate fi numit un sintetizator TTS, pentru că nu acceptă text brut ca date de intrare. Rezultatul generat de MBROLA este un fișier audio pe 16 biți având aceeași frecvență cu cea la care au fost înregistrate difonurile bazei de date. Acest fișier audio reprezintă sinteza vocală a fonemelor introduse.

MBROLA este distribuit gratuit pentru aplicații non-comerciale și folosește algoritmul TD-PSOLA pentru sinteza vocală.

## 6. Concluzii și dezvoltări viitoare

În această lucrare am prezentat noi metode de procesare a textului, ilustrate printr-un sistem TTS pentru limba română, dar care pot fi adaptate pentru numeroase alte limbi. Datorită algoritmilor folosiți, sistemul prezintă o foarte mare flexibilitate, iar prin eforturile depuse, vorbirea produsă de program este de foarte bună calitate. Nu în ultimul rând, datorită particularităților limbii române, vocea artificială generată este net superioară multor alte limbi străine.

Posibile îmbunătățiri ce vor fi aduse în versiunile viitoare ale programului nostru sunt:

- crearea unui modul prozodic mult mai performant, bazat pe inteligență artificială (implementarea unei rețele neurale care, o dată antrenată, va putea accentua corespunzător anumite părți de propoziție, va putea asigura un ritm coerent la citirea unei întregi fraze și va putea respecta mult mai bine intonația în cazul semnelor de punctuație);
- îmbunătățirea modului preprocesor pentru a putea trata posibilele erori care pot apărea din cauza prezenței în text a unor caractere necunoscute; o altă îmbunătățire care ar putea fi adusă acestui modul constă în corectarea textului introdus înainte de a fi citit, (funcționalitate asemănătoare cu cea a uneltelor lingvistice din Microsoft Word);
- implementarea unui modul de emoționalitate (vocea artificială generată să poată fi ”bucuroasă”, ”supărată”, ”speriată” sau ”neutră”);
- implementarea unui modul de cântec (care să ofere posibilitatea cântării textului, o dată cu furnizarea unei partituri);
- înregistrarea de noi voci de vârste diferite, de sexe diferite, cu accente diferite, etc.

De asemenea programul ar putea deveni mai folositor prin adăugarea de noi funcționalități precum:

- citirea ceasului, sau a anumitor mesaje preprogramate de către utilizator;
- citirea textului din fișiere de formate diferite (pdf, doc, html, etc.);
- salvarea fișierului audio generat în diferite formate (mp3, wav, ogg, etc).

## Bibliografie

1. Taylor P., *Text-to-Speech*, Cambridge University Press, Cambridge, 2009
2. Dutoit T., *A Comparison of Four Candidate Algorithms in the context of High Quality Text-to-Speech Synthesis*, ICASSP'94, Adelaide, Australia, 1994
3. Burileanu D., *A Parser-based Text Preprocessor for Romanian Language TTS Synthesis*, Eurospeech'99, Budapesta, Ungaria, 1999
4. Buză O., *A Romanian Syllable-Based Text-to-Speech System*, 6th WSEAS International Conference, Insula Corfu, Grecia, 2007
5. Ferencz A., *Romvox – Experiments Regarding Unrestricted Text-to-Speech Synthesis for the Romanian Language*, 6th EWNLG, Duisburg, Germania, 1998
6. <http://tcts.fpms.ac.be/synthesis/mbrola.html>
7. [http://en.wikipedia.org/wiki/Speech\\_synthesis](http://en.wikipedia.org/wiki/Speech_synthesis)

Stancu Mihai  
 Universitatea „Lucian Blaga” din Sibiu  
 Facultatea de Științe  
 Specializarea Matematică - Informatică (anul I)  
 stancu\_meehigh@yahoo.com]

## Mobile Earth

**Autor:** STOICA Lucian, **Universitatea** "Lucian Blaga", Sibiu  
**Coordonator:** prof. univ. dr. SIMIAN Dana, **Universitatea** "Lucian Blaga", Sibiu,  
Departamentul Informatica

*Abstract: This project represents a powerful tridimensional editor that is based on polygons (same as 3D Studio Max or Maya). It has been built with Microsoft Visual C# and functions from OpenGL (Open Graphic Library) to improve the performance and fiability. The objects and maps that are built with this 3D editor can be saved in files with .3DC extension (new format created by the author), and that files can be loaded in other applications, if the programmers know the .3DC format. For security reasons, this format is not public for the moment. Also 3D Studio Max files (.3DS) can be loaded, and edited with this tool. After an object is created this is saved in a Keyhole Markup Language (.KML) file and can be previewed with Google Earth application.*

*In the end I want to present my special thanks to my teacher, Dana Simian, and to my best friend Gabriel Szocs, for helping me with mathematical problems.*

*Cuvinte-cheie: Mobile Earth, Google Earth, OpenGL, C#.*

### 1. Introducere

Mobile Earth este o "unealta" pentru crearea de obiecte si harti tridimensionale, bazat pe lucrul cu poligoane. Proiectul are o interfata cu utilizatorul prietenoasa si interactiva in acelasi timp, oferindu-i acestuia posibilitatea de a crea, cu o foarte mare usurinta, o multitudine de obiecte tridimensionale, care pot fi salvate in fisiere cu extensia **.3DC**, si apoi pot fi importate in alte aplicatii care au ca scop construirea unei lumi sau a unor obiecte tridimensionale. Totodata, cu ajutorul acestui editor 3D se pot incarca obiecte create in 3D Studio Max, iar in viitor se prevede incarcarea obiectelor Maya si MD2. Odata creat un astfel de obiect, acesta se poate salva intr-un fisier cu extensia **.KML** si apoi vizualizat cu ajutorul Google Earth.

**.3DC** este un format nou de fisier, creat de autorul lucrarii, si datorita modificarilor care pot aparea in viitorul apropiat, si din unele motive de securitate, acest format nu este inca facut public.

Proiectul in sine, este o aplicatie creata in mediul de dezvoltare **Microsoft Visual C#**, si foloseste pentru viteza si fiabilitate, functii din biblioteca **OpenGL**, functii care sunt implementate direct pe placa video a calculatorului. Nu exista aplicatie in ziua de azi, care sa nu foloseasca functii **OpenGL** sau **DirectX** pentru grafica tridimensionala.

Acest proiect permite utilizatorului asezarea unor obiecte tridimensionale intr-o lume 3D, obiecte ce pot fi vizualizate mai apoi cu ajutorul Google Earth.

S-a ales dezvoltarea unui asemenea proiect, deoarece in domeniul GPS si al aplicatiilor care folosesc GPS-ul, este o concurenta acerba, iar crearea unui asemenea proiect necesita cunostinte foarte bogate despre lucrul cu hartile digitale si lumea tridimensionala.

In viitor (peste aproximativ o luna) se prevede integrarea acestui proiect pe telefoanele mobile care au ca sistem de operare Windows Mobile, au implementat hardware OpenGL Embedded System, pentru un Fps cat mai mare, unul dintre aceste telefoane fiind Samsung Omnia.

## 2. Prezentarea proiectului.

La inceputul rularii aplicatiei, va aparea o platforma tridimensionala de culoare albastra, formata din mai multe patrate, fiecare asemenea patrat avand latura egala cu 1m, iar in partea dreapta a ecranului un meniu cu diferite optiuni, in functie de actiunea intreprinsa de utilizator, asa cum se poate vedea in Fig. 1.

Din sectiunea "Menu" se poate alege editarea unui obiect, salvarea obiectului intr-un fisier .KML, iesirea din aplicatie, etc. Pentru crearea unui polygon se fixeaza creionul grafic (punctul de culoare galbena din centrul ecranului), si se apasa o data tasta SpaceBar (spatiu), acesti pasi se repeta de inca 3 ori, obtinandu-se in final un poligon. In randurile ce urmeaza sunt prezentate mai detaliat actiunile ce se pot executa pe un astfel de poligon.

Sectiunea "Position" se refera la coordonatele creionului grafic (coordonate care se pot schimba folosind tastele WASD, respectiv LEFT\_ARROW, RIGHT\_ARROW, pe toate cele trei axe (XOY)). Este important de retinut faptul ca din varful creionului grafic se fixeaza toate cele patru puncte, pe rand, pentru obtinerea unui poligon (patrulat).

Sectiunea "Polygon Action", face referinta la actiunile pe care le intreprindem asupra unui poligon (mutare, rotire, setare textura, stergere, etc). Daca, de exemplu, se doreste rotirea poligonului curent pe axa OY atunci se alege din lista de actiuni "Rotate", dupa care se va bifa check box-ul "Y" si se va da click pe unul dintre butoanele "+" sau "-".

Sectiunea "Object Action", este cea mai importanta deoarece ne permite sa efectuam toate actiunile mai sus mentionate, dar asupra unor obiecte. Un obiect nu este altceva decat o succesiune de poligoane, care se retin intr-o lista. Un mare avantaj al acestui proiect este faptul ca se pot incarca obiecte .3DS, se pot face modificari pe aceste obiecte, si apoi pot fi salvate. Daca utilizatorul doreste sa adauge un nou obiect la cele existente deja, atunci va alege din lista de actiuni a obiectelor "Add new object", va selecta tipul obiectului care doreste sa-l adauge (3DC sau 3DS), dupa care va da click pe butonul "Add". De retinut este faptul ca obiectul va aparea in punctul in care este plasat creionul grafic, si punctul de origine al obiectului (necesar la rotirea sa), coincide cu primul punct din primul poligon al obiectului.

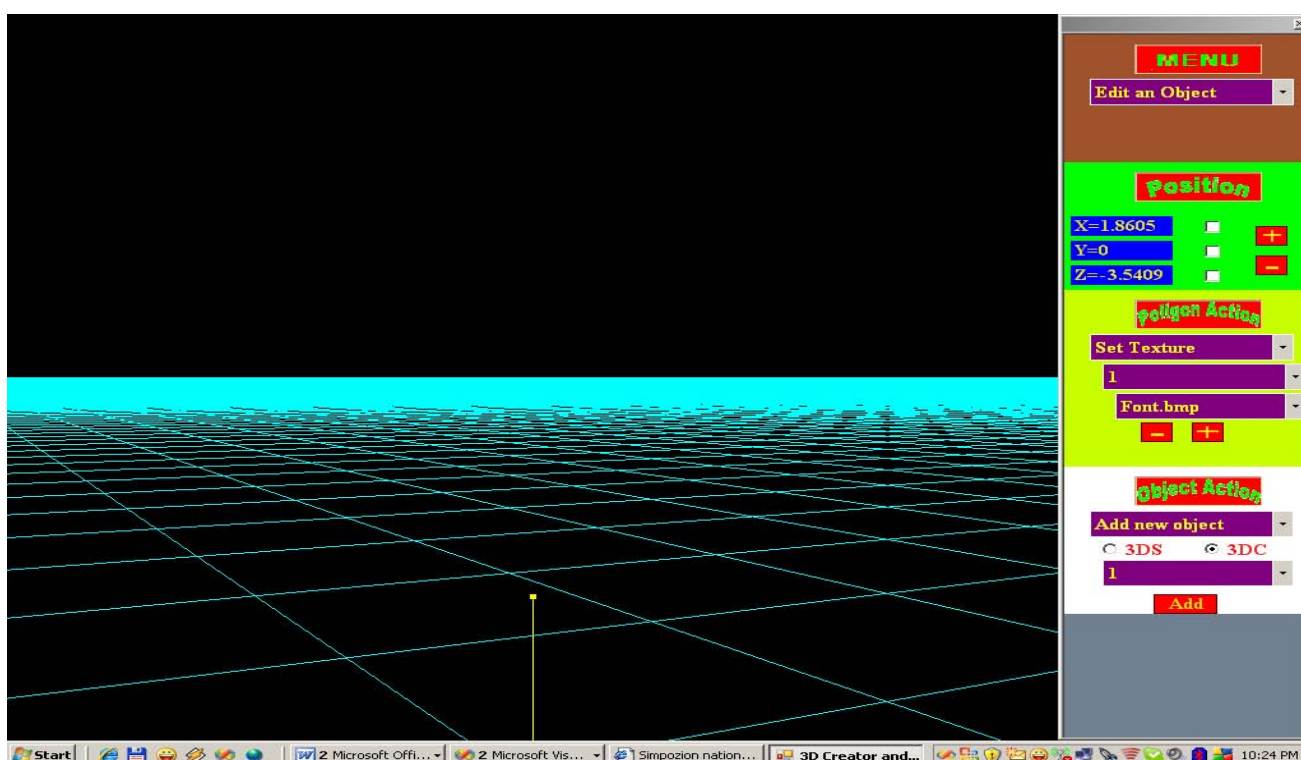


Fig. 1

In Fig. 2 este prezentat un cub, obtinut prin adaugarea repetata de astfel de poligoane, iar in Fig. 3 se poate vedea un obiect 3DS.



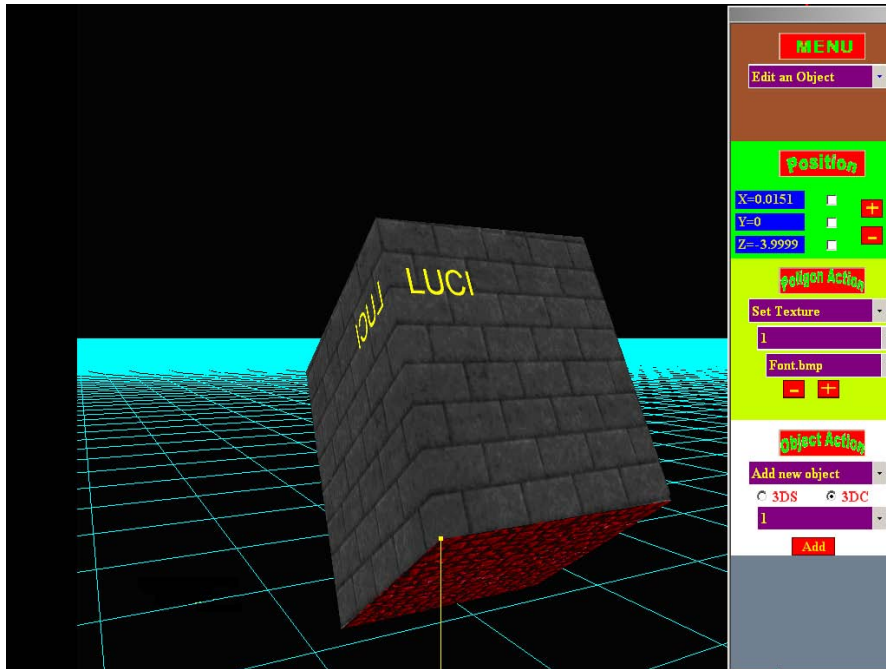


Fig. 2

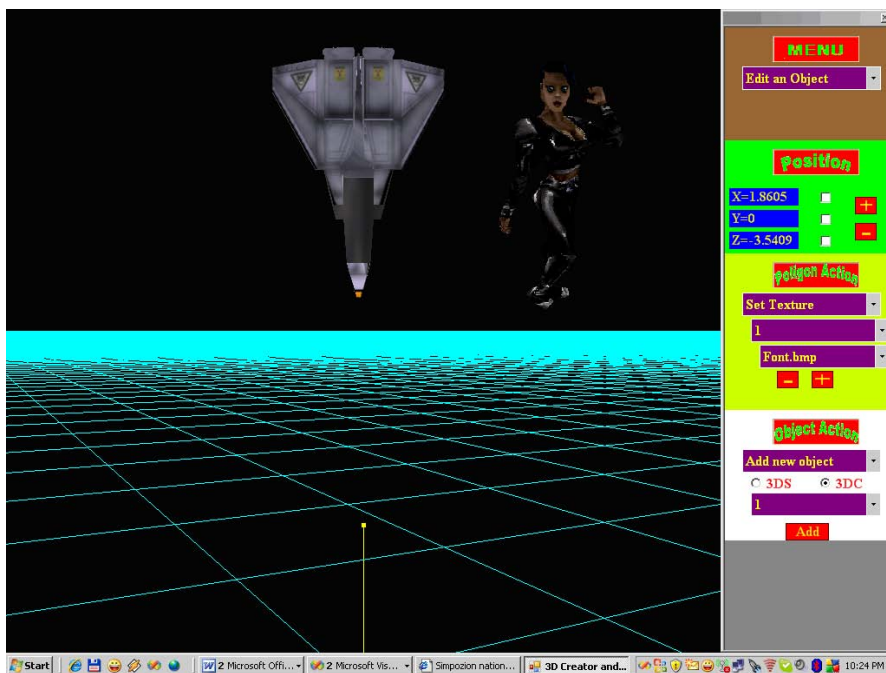


Fig. 3

Odata salvate aceste obiecte in fisiere cu extensia .KML, acestea pot fi vizualizate cu ajutorul Google Earth, ca in Fig. 4, unde se poate vedea Pentagonul.

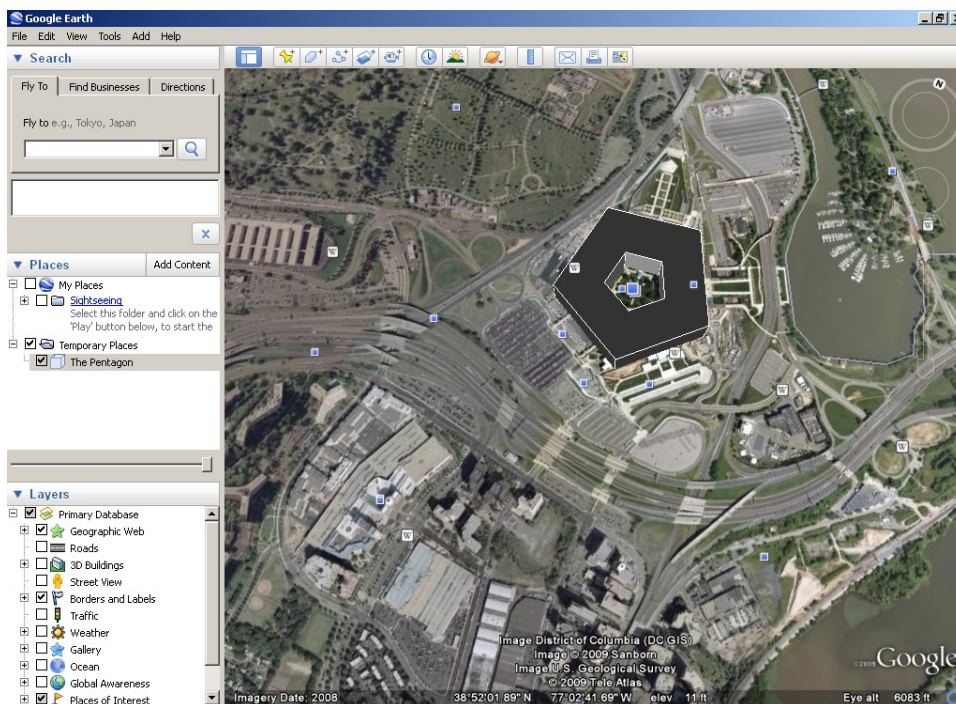


Fig. 4

Pentru plasarea unui obiect -creat cu acest editor- intr-un spatiu Google Earth se poate alege din lista care apartine meniului Object Action, optiunea Add in Google Earth, dupa care va aparea o fereastra ca in Fig. 5, unde in partea superioara se poate recunoaste cu usurinta fereastra Google Earth iar in partea inferioara este o fereastra Mobile Earth, care afiseaza ceea ce "vede" Google Earth, dar ne permite sa vizualizam si obiectele noastre. Se poate vedea mai jos cum s-a plasat un obiect tridimensional (un avion), deasupra centrului istoric al orasului Sibiu, folosind aceasta aplicatie.

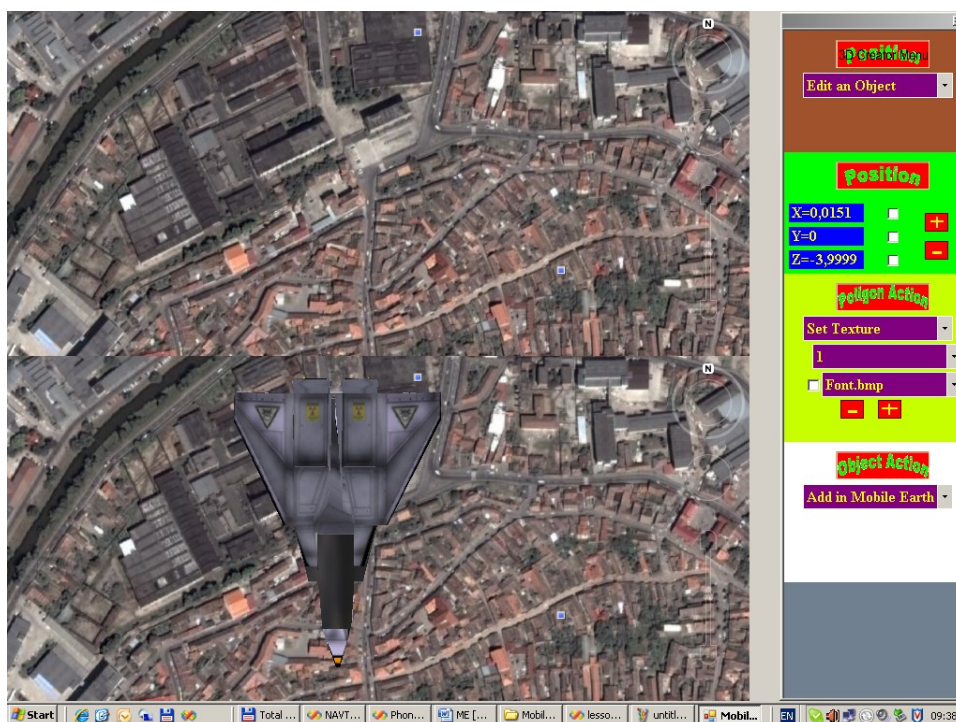


Fig. 5

### 3. Fundamente matematice necesare proiectului.

Este cunoscut faptul ca toate aplicatiile de grafica tridimensionala (in special jocurile 3D), folosesc elemente de matematica superioara, mai exact geometrie in spatiu si trigonometrie, iar acest proiect nu face exceptie. In cele ce urmeaza, sunt explicate prin formule matematice, cum s-au obtinut rotatiile poligoanelor, ciocnirea cu peretii, translatiile fata de un anumit punct, etc.

Pentru ciocnirea cu peretii s-a plecat de la faptul ca daca

$$\begin{vmatrix} x & y & z & 1 \\ x1 & y1 & z1 & 1 \\ x2 & y2 & z2 & 1 \\ x3 & y3 & z3 & 1 \end{vmatrix} = 0$$

unde:

-x,y,z sunt coordonatele in spatiu ale utilizatorului

-x1,y1,z1,x2,y2,z2,x3,y3,z3 sunt cele trei varfuri ale unui poligon

atunci utilizatorul s-a "ciocnit" de un poligon.

Pentru rotirea unui punct in spatiu pe axa OX, in functie de o origine s-a folosit formula:

$$y' = y * \cos(\alpha) - z * \sin(\alpha)$$

$$z' = y * \sin(\alpha) + z * \cos(\alpha)$$

Pentru rotirea unui punct in spatiu pe axa OY, in functie de o origine s-a folosit formula:

$$x' = x * \cos(\alpha) - z * \sin(\alpha)$$

$$z' = x * \sin(\alpha) + z * \cos(\alpha)$$

Pentru rotirea unui punct in spatiu pe axa OZ, in functie de o origine s-a folosit formula:

$$x' = x * \cos(\alpha) - y * \sin(\alpha)$$

$$y' = x * \sin(\alpha) + y * \cos(\alpha)$$

Dar aceste rotatii sunt valabile numai pentru punctele care au ca origine punctul P(0,0,0), si atunci pentru a face posibila rotirea unui punct A(xa,ya,za) fata de un alt punct de origine (sa zicem O(xo,yo,zo)) trebuie sa facem translata A(xa-ox,ya-yo,za-zo), apoi rotim punctul aplicand una dintre formulele mai sus mentionate, dupa care refacem translata originala a punctului A, astfel:

A(xa+ox,ya+yo,za+zo).

Pentru rotirea unui obiect - care nu este altceva decat o succesiune de poligoane-, pe axa OX, se foloseste functia Rotate\_X() care primeste ca parametrii numarul obiectului, punctul in jurul caruia dorim sa rotim acest obiect, si unghiul sub care dorim sa facem rotatia, iar apoi intr-un ciclu `for` rotim toate poligoanele care apartin obiectului respectiv. Procedam similar pentru rotirea obiectului pe axa OY sau OZ.

```
public void Rotate_X(int obj, CPoint3D ptOrigin, double Angle)
{
    for (int i = gl.vobj[obj].Start_Poly; i <= gl.vobj[obj].End_Poly; i++)
        gl.Polygons.Rotate_X(i, ptOrigin, Angle);
}
```

Pentru a afla daca utilizatorul "s-a ciocnit" de un perete am folosit functia determinant(), care primeste ca parametru un poligon si verifica daca determinantul mentionat in capitolul precedent are valoarea apropiata de zero, in acest caz obtinandu-se o coliziune.

```
float determinant(POLIGON unpoligon)
{
    return -1*((unpoligon.x1-sold.x)*(unpoligon.y2-sold.y)*(unpoligon.z3-sold.z)+
(unpoligon.y1-sold.y)*(unpoligon.z2-sold.z)*(unpoligon.x3-sold.x)+
(unpoligon.x2-sold.x)*(unpoligon.y3-sold.y)*(unpoligon.z1-sold.z)-
(unpoligon.z1-sold.z)*(unpoligon.y2-sold.y)*(unpoligon.x3-sold.x)-
(unpoligon.x1-sold.x)*(unpoligon.z2-sold.z)*(unpoligon.y3-sold.y)-
```

```
(unpoligon.x2-sold.x)*(unpoligon.z3-sold.z)*(unpoligon.y1-sold.y));
}
```

In acest capitol am prezentat o infima parte din elementele de matematica necesare realizarii proiectului, din dorinta de a nu descuraja persoanele care se descurca mai greu la acesta parte, cunoscandu-se faptul ca elementele de geometrie si trigonometrie stau la baza oricarei aplicatii de grafica tridimensionala, iar explicarea intregii parti matematice ce face parte din acest proiect ar fi mult prea mare, si nu si-ar avea rostul in aceasta scurta prezentare.

#### 4. Implementarea proiectului

Dupa cum am mentionat la inceputul prezentarii acestui proiect, aplicatia a fost creata cu ajutorul limbajului de programare Microsoft Visual C# si al platformei Microsoft .NET Framework.

S-a ales ca limbaj de programare limbajul C# deoarece este foarte puternic orientat obiect, si aplicatiile dezvoltate intr-un astfel de mediu se pot controla mult mai usor, iar pe partea de interfata are foarte multe elemente ajutatoare, si bineinteles ne ofera accesul la platforma .NET Framework.

Pe telefonul mobil s-a ales platforma Microsoft .NET Compact Framework, deoarece are aproape aceleasi beneficii ca si "fratele ei mai mare", si anume .NET Framework. Grafica tridimensionala de pe telefoane este una de calitate superioara deoarece foloseste functii din biblioteca OpenGL ES (Embeded System), iar pe un telefon cum ar fi Samsung Omnia, care are implementat OpenGL in mod hardware, avantajele sunt evidente.

S-a ales ca aplicatia sa fie implementata pe device-uri care au ca sistem de operare Windows Mobile, deoarece este cel mai raspandit sistem de operare care se gaseste pe telefoanele mobile la ora actuala si probabil si in viitor. Se prevede implementarea proiectului si pe telefoane mobile care au ca sistem de operare Symbian (Nokia, Sony Ericsson). In rest implementarea graficii tridimensionale pe telefonul mobil nu a fost foarte dificila deoarece are la baza aceleasi elemente de grafica 3D dar a trebuit sa se tina cont de faptul ca un telefon mobil oricat de performant ar fi nu poate egala, deocamdata, un calculator.

#### 5. Exemplificare

Dupa ce am creat acest editor 3D, l-am folosit pentru a realiza o aplicatie tridimensionala care prezinta cladirea S.C. Route66, din Brasov, cladire ce se poate vedea cu ajutorul Google Earth, cu conditia de a avea fisierul respectiv pe calculator (KML). Utilizatorul poate sa aleaga dintr-o lista numele unui birou in care doreste sa ajunga, sa spunem de exemplu Directiune, iar proiectul il va conduce virtual pana la acel birou, si il va asista pe utilizator si cu sunete cum ar fi: "mergeti inainte 5 metri, apoi faceti la dreapta, treceti pe langa Secretariat...".

In viitorul apropiat intentionam dezvoltarea si folosirea acestui editor 3D cu scopul de a crea obiecte tridimensionale cat mai detaliate, si totodata se doreste incorporarea unui modul pentru editarea celor mai elementare functii, pentru editarea fisierelor .KML, dupa care se va trece la implementarea aplicatiei pe telefoanele mobile care au ca sistem de operare Windows Mobile. Cu ajutorul acestui proiect se vor putea crea si prezenta cladiri tridimensionale (muzee, scoli, spitale, etc).

In final, as dori sa multumesc pe aceasta cale, Coordonatorului Stiintific Prof. Univ. Dr. Dana Simian, si prietenului meu Gabriel Szocs, pentru ajutorul pe care mi l-au acordat la rezolvarea diferitelor probleme de matematica care au aparut pe parcursul dezvoltarii acestui proiect.

#### Bibliografie

1. Tom McReynolds, David Blythe - Advanced Graphics Programming Techniques Using OpenGL, 26 Aprilie 1998,
2. Computer Graphics Using OpenGL 2D and 3D - Hong Zhang, Y. Daniel Liang – Armstrong, 06 Decembrie 2006,
3. The OpenGL Graphics System: A Specification - Mark Segal, Kurt Akeley, 22 Octombrie 2004
4. An Introduction to Graphics Programming with OpenGL - Toby Howard, 26 Ianuarie 2004
5. [www.google.com](http://www.google.com) – tutoriale si documentatii API.

Mobile Earth

**Stoica Lucian**

**Universitatea „Lucian Blaga” din Sibiu**

**Facultatea de Științe**

**Specializarea Tehnologia Informatiei (program de masterat)**

lucil5ian@yahoo.com

## Construirea de Jocuri Interactive in Matlab

**Autori:** Cosma Stefan, **Universitatea „Lucian Blaga” Sibiu**  
Fireșcu Andreea, **Universitatea „Lucian Blaga” Sibiu**  
Nechifor Vasile Nicusor, **Universitatea „Lucian Blaga” Sibiu**

**Coordonator:** Ionela Maniu, **Universitatea „Lucian Blaga” Sibiu**, Departamentul de Informatica

*Abstract:* Jocurile interactive sunt o parte din rutina zilnică. Ele sunt, de obicei, dezvoltate cu software specializat. Am încercat să dezvoltăm, cu ajutorul Mathworks Matlab, un joc interactiv. Scopul nostru a fost acela de a vedea dacă este posibil dezvoltarea unui astfel de joc interactiv cu ajutorul matematicii. Mathworks Matlab ne-a oferit un mediu care ne-a ajutat în dezvoltarea acestui tip de jocuri interactiv. Acest proiect ilustrează ideea de joc interactiv, cu ajutorul programele bazate pe matematica..

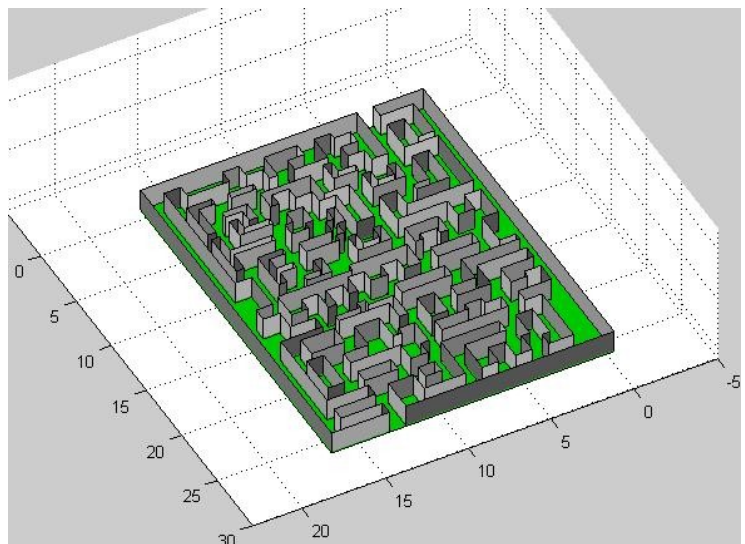
*Cuvinte-cheie:* Jocuri Interactive, Matlab, Realitate Virtuala, Mediu Matematic.

### 1. Introducere

Jocurile interactive, o parte a vieții cotidiene din zilele noastre. Acestea pot fi foarte complicate sau foarte complexe. Pornind de la simplul joc, Minesweeper și până la cele mai avansate jocuri 3D care folosesc cea mai dezvoltată tehnologie în materie de motoare grafice. Acest proiect stă la baza unei idei care a rezultat în urma încercării de a dezvolta un mediu interactiv într-un mediu în care în general nu se pot dezvolta astfel de proiecte. Programul Mathworks Matlab a fost ales pentru construirea unui joc interactiv deoarece din acesta privință nu s-a mai încercat implementarea unei asemenea idei în mediul Matlab. Ideea a fost aleasă deoarece jocurile au un rol foarte important în dezvoltarea caracterului și pentru că un mediu interactiv este mult mai atractiv decât de exemplu, un mediu de formule matematice.

### 2. Formularea Problemei Fig. 1

Jocul creat ar trebui să stimuleze abilitățile utilizatorului de a găsi singur soluții cât mai eficiente pentru rezolvarea unor probleme. Dezvolta creativitatea fără a genera frustrarea jucătorului în caz de nereușită. Conceptul după care a fost creat jocul interactiv se rezumă la: dezvoltarea unui labirint în format 3D și utilizarea săgeților de pe tastatură pentru a naviga în interiorul acestuia. Problema care a rezultat din ideea inițială a fost implementarea grafică a unui astfel de joc interactiv în mediul Matlab.





## 2.4 Generarea Labirintului

Labirintul a fost realizat folosind o metoda foarte simpla de introducere a coordonatelor fiecarui punct. Fiecare punct a fost generat separat prin introducerea fiecărei coordonate in parte. Pereti apoi a fost definiti in urma generarii punctelor de intersectie. Sa incercat introducerea mai multor elemente dar acest lucru ar fi ingreunat labirintul din punct de vedere grafic. Lipsa elementelor, care ar fi ingreunat labirintul din punct de vedere grafic, chiar daca acest lucru ar genera o anumita monotonie din partea utilizatorului, arata de fapt modul simplu care a fost utilizat pentru a genera labirintul.

Dupa generarea labirintului s-a trecut la crearea unui obiect care sa se deplaseze in interiorul labirintului pentru a se observa cum lucreaza jocul.

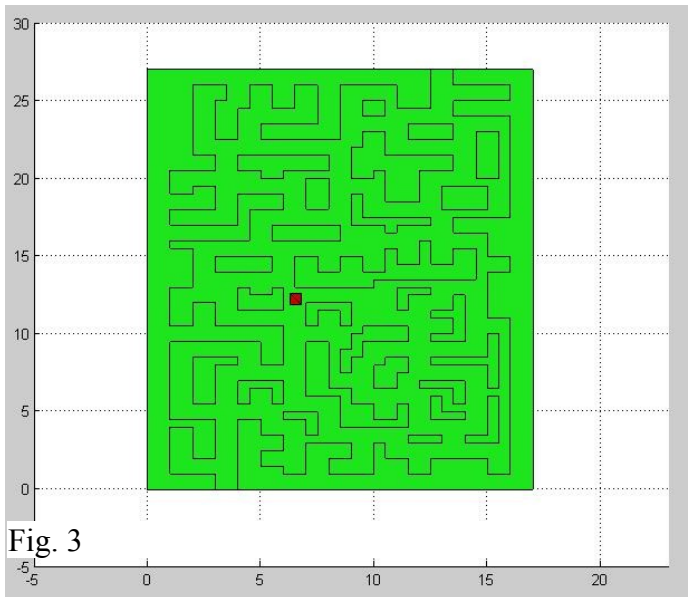


Fig. 3

### 1. Rezultate principale

Principalele rezultate care s-au fost obtinute in urma dezvoltarii unui astfel de joc au fost faptul ca matematica este o parte importanta a dezvoltarii jocurilor, chiar daca in ziua de azi exista programe specializate care numai folosesc direct matematica. Un alt rezultat important ar fi acela ca dezvoltarea de jocuri interactive este posibila in orice mediu, fie el si unul matematic. Scopul acestui joc este de a intelege mai bine cum lucreaza anumite programe cu medii grafice si cum pot fi imbunatatite jocurile din ziua de azi folosind metode matematice simple.

Un astfel de joc precum cel din Fig. 2 este generat in mediul Matlab, acesta ilustreaza cel mai bine conceptul de joc interactiv generat cu ajutorul programului Matlab. Se poate spune ca rezultatul obtinut in urma implementarii acestei idei cu ajutorul programului Matlab este unul foarte satisfactor deoarece se pot obtine o multime de date care pot ajuta la o mai buna dezvoltare de jocuri interactive.

### 2. Concluzii și dezvoltari viitoare

Proiectul de fata este doar o idee a carei implementare s-a incercat in mediul Matlab si sa ajuns pana la un anumit punct. Constructia unui astfel de joc necesita foarte mult timp in mediul Matlab. Ca si dezvoltarii viitoare acestui joc interactive i se pot aduce noi componente sau o interfata noua. Se poate dezvolta sistemul de navigatie dar si sistemul de vizualizare al jocului in sine. Se poate face o dezvoltare majora din punct de vedere al graficii dar si al elementelor care fac parte din joc. Pana la urma mediul Matlab a fost prielnic pentru crearea unui asemenea joc chiar daca nu se compara cu programele specializate in acest domeniu.

In concluzie as dori sa specific faptul ca jocurile interactive pot fi dezvoltate in orice mediu fie el pur matematic sau direct grafic. Utilizarea unui program precum Mathworks Matlab creste probabilitatea de succes a unei incercari de implementare a unui program, fie el joc interactiv sau aplicatie precum un editor de texte. Pana la urma toate aceste concepte pot fi implementate in ralitatea virtuala care reprezinta o ramura

Fig. 4





importanța a informaticii. Acestea oferă modalitatea prin care calculatorul și echipamentele specializate modifică modul în care omul percepe realitatea din mediul natural, prin simularea/modelarea unei alte realități.

### **Bibliografie**

1. Marin V., Mircea P., - *Realitatea Virtuală (Virtual Reality), tehnologie modernă a informaticii aplicate*
2. [www.mathworks.com](http://www.mathworks.com) - tutoriale și demonstrații
3. [www.mathtools.net](http://www.mathtools.net) – exemple de jocuri interactive

**Cosma Stefan**  
**Universitate „Lucian Blaga” Sibiu**  
**Facultate de Științe**  
**Specializarea Informatica (program de licență)**  
E-mail: [steficosma@yahoo.com](mailto:steficosma@yahoo.com)

**Firescu Andreea**  
**Universitatea „Lucian Blaga” din Sibiu**  
**Facultatea de Științe**  
**Specializarea Informatica (program de licență)**  
E-mail: [andreea\\_vl@yahoo.com](mailto:andreea_vl@yahoo.com)

**Nechifor Vasile Nicusor**  
**Universitatea „Lucian Blaga” din Sibiu**  
**Facultatea de Științe**  
**Specializarea Informatica (program de licență)**  
E-mail: [vasilenicusor@yahoo.com](mailto:vasilenicusor@yahoo.com)

## Joc de Strategie

Sopa Christian, Universitatea Lucian Blaga, Facultatea de Inginerie, Sibiu

*Abstract: This project implements most basic functions to create a strategy game. It features a technology tree, the possibility to play against a computer opponent and all algorithms needed for these to work. Graphics, the input, the units' behaviour, their pathfinding, a basic Artificial Intelligence for the computer opponent, a menu, are just some of the problems we need to handle.*

*Cuvinte cheie: strategy game, pathfinding algorithm, graphics*

### 1 Introducere

Programarea de jocuri este un domeniu foarte complex, dându-i programatorului multe posibilitati de abordare a problemelor de care se lovește. Proiectul curent este un joc de strategie, după modelul jocurilor existente pe piață; scopul meu a fost de a studia diverse metode de a rezolva subproblemele acestui tip de joc: Un management al obiectelor / claselor, găsirea celui mai scurt drum între două puncte (într-un mod eficient), implementarea unei inteligențe artificiale, sisteme de afișare grafică, etc.

Fiecare dintre aceste probleme, plus multe altele minore, se pot rezolva în diverse metode (algoritmi), utilizând biblioteci corespunzătoare.

### 2 Formularea Problemei

În această lucrare, s-au considerat următoarele scopuri:

- Realizarea unui engine de bază pentru a fi utilizat în un joc de strategie. Acesta ar trebui să cuprindă:
  - o Initializare (citire date externe, initializare mod grafic, etc)
  - o Bucla principală (tratarea evenimentelor (mouse/keyboard), procesarea informației (unitati, miscare, atacuri, AI)
  - o Output: grafică și sunet, trebuie performantă
- Realizarea unui HUD (unde vom citi date și vom putea da comenzi)
- Realizarea unor algoritmi de găsire a drumului optim, optimizări
- Realizarea unei inteligențe artificiale de bază, pentru a putea fi jucat împotriva calculatorului
- Realizarea unui „technology tree”, după terminologia jocurilor; realizare de clădiri, management resurse, unitati, propriile abilitati.

#### 2.1 Engine

Engine-ul jocului este simplu, structurat obiectual. Contine o funcție de initializare, o buclă principală în care tratăm evenimentele și apoi distrugerea datelor.

## 2.2 HUD

HUD-ul cuprinde un „minimap” cu care ne putem orienta mai usor pe harta, detalii despre unitatea/unitatile selectate si cateva butoane cu care putem da comenzi (de atac, de constructie, etc).

## 2.3 Pathfinding

Pathfinding-ul este realizat in mai multe etape:

- Folosirea algoritmului A\* pentru a gasi drumul cel mai scurt, pe 8 directii (pentru viitor voi implementa HPA\* pentru performante)
- „Smoothing” la drum, pentru a nu ne limita la cele 8 directii date de algoritmul

Dupa prima cautare de path, in cazul impiedicarii de un obiect (alta unitate) se aplica acesti algoritmi, putin modificati, pentru a isi gasi o alta cale optima in situatia noua.

## 2.4 AI

Inteligența artificială este o inteligență scriptată, ca ordine de construcție / momente de atac. Pentru moment, nu este implementată foarte avansat, acest lucru va fi făcut în viitor.

## 2.5 Technology Tree

Multimea unitatilor / cladirilor pe care le are jucatorul la dispozitie va fi intotdeauna in lucru, in functie de idei. Usurinta de a implementa noi unitati provine de la structurarea obiectuala atat a HUD-ului, cat si proiectarea clasei PUnit, care ne ofera flexibilitatea necesara pentru a implementa aceste lucruri.

## 4 Rezultate principale

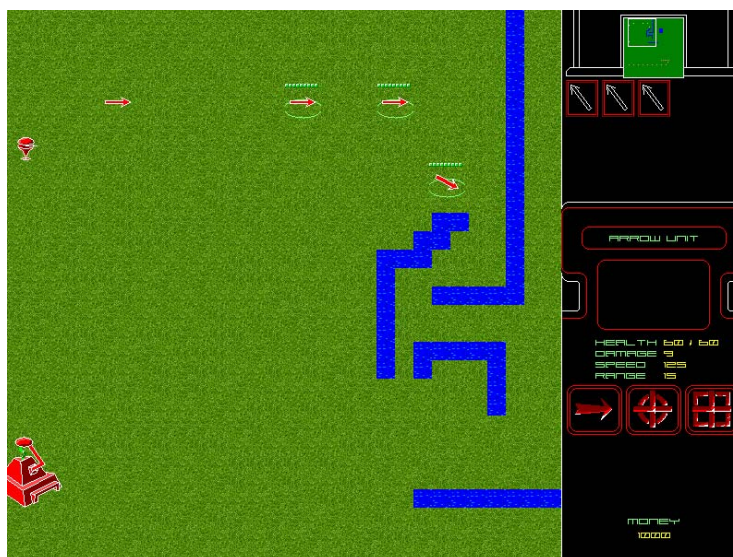


Fig. 1

Dupa cum se observa in screenshot-ul anterior, avem o grafica simpla 2D (se poate modifica oricand pentru ca imaginile sunt bitmap-uri), cu un HUD pentru control in dreapta ecranului, si in restul ecranului vedem scena din joc.

Mentionez ca proiectul este in stadiu de dezvoltare, fiind nevoie de multi algoritmi diversi si multe adaugari pentru a ajunge la scopul final, un joc complet jucabil. Mai multe detalii se pot gasi in planurile pentru dezvoltari viitoare.

## 5. Concluzii și dezvoltari viitoare

Cu ajutorul bibliotecii SDL pentru afisare grafica, sunet si interfata, si un sistem de clase definind obiectele din scena, am realizat engine-un principal al jocului.

A\* si mici optimizari ne-a folosit la a ne ajuta obiectele sa se deplaseze pe harta.

Unele din planurile mele pentru viitor sunt:

- Sa imbunatatesc AI-ul calculatorului
- Utilizarea algoritmului HPA\* pentru pathfinding
- Utilizarea unui engine grafic 3D (bazar ori pe openGL ori DirectX)
- Posibilitatea de a avea mai multi players (controlati de calculator)
- Posibilitatea de a juca in retea

### Bibliografie

1. <http://libsdl.org>
2. <http://aigamedev.com/theory/near-optimal-hierarchical-pathfinding>
3. Diverse eBooks, gasite pe internet.

**Sopa Christian**  
**Universitatea Lucian Blaga, Sibiu**  
**Facultatea de inginerie „Hermann Oberth”**  
**Specializarea Calculatoare, an II**  
chr@gladiusdei.org

**Sesiunea de Comunicări Științifice a Studenților  
"Imaginație, Creativitate, Design, Dezvoltare"  
Sibiu – România, 2009**

## **PARTEA 2**

### **Secțiunea dedicată elevilor Abstracte extinse**

## APLICAREA PROGRAMEI ȘCOLARE ÎN ÎNVĂȚĂMÂNTUL PREUNIVERSITAR

Prof. Preda Georgeta

*Key-Words: programa școlară în învățământul preuniversitar, oracle, lecții AEL*

### 1 Introducere

Informatica a pătruns astăzi în cele mai variate domenii, prin urmare, indiferent de profesia pe care o va alege un tânăr, la viitorul lui loc de muncă – în mileniul III – cu siguranță va avea nevoie de valorificarea mijloacelor informatice pentru rezolvarea unor sarcini de lucru specifice.

Pornind de la faptul că nu există domeniu de activitate unde să nu se prelucreze și să nu se transmită informații atât în cadrul domeniului respectiv cât și spre exteriorul lui, afirmăm că azi informația este foarte prețioasă, ea trebuie stocată, prelucrată și transmisă în condiții care asigură corectitudine și exactitate, deci la nivel profesional.

În funcție de profilul liceului (teoretic, grup școlar) și de specializare (Matematică informatică, Științe ale naturii, Filologie) în aria curriculară "Tehnologii" se studiază trei discipline: Informatică, Tehnologia informației și a comunicațiilor și Educație antreprenorială.

În acest context, disciplinele „Informatică” și „Tehnologia informației și a comunicațiilor”, trebuie să asigure dobândirea unor competențe și cunoștințe de utilizare a calculatorului și a unor programe informatice cu aplicabilitate în mediul de inserție profesională vizat de absolvenții filierelor, profilurilor și specializărilor la care se studiază aceste discipline.

Dezvoltarea deprinderilor moderne de utilizator, respectiv pregătirea elevilor astfel încât să beneficieze de avantajele „științei calculatorului”, pentru realizarea intereselor sale educaționale, reprezintă una din componentele de bază ale formării de specialitate.

Domeniul tehnologiilor informaționale, prin specificul său, este esențial legat de lucrul individual pe un calculator, deci dezvoltă deprinderea de a lucra individual. Pe de altă parte, prin intermediul rețelelor de calculatoare este posibil schimbul de informații între mai mulți utilizatori de calculatoare mult mai eficient decât prin orice altă metodă clasică.

Formarea elevilor în spiritul unei activități desfășurate în grup, în colaborare, se finalizează prin predarea tehnologiei informației și comunicațiilor orientată pe proiecte. Obisnuirea elevilor cu responsabilități, cu răspunderea privind finalizarea propriei munci și asigurarea înlănțuirii unor elemente realizate în paralel, îi va pregăti în mod cât se poate de clar pentru o activitate pe care cu siguranță o vor întâlni în viitor. Educarea elevilor pentru realizarea unor produse utilizabile, dezvoltarea spiritului inventiv și creator apare ca un obiectiv impus de sistemul economic în care trăim și vom trăi și în viitor. Indiferent de conținutul aplicației, ceea ce realizează elevul, trebuie să fie utilizabil; altfel spus, trebuie să aibă toate calitățile unui produs.

Datorită implicației pe care tehnologia informației și comunicațiilor o are azi în toate profesiile, rezultă caracterul ei interdisciplinar. Deci, nu putem vorbi despre tehnologia informației și comunicațiilor pur și simplu. Ea nu poate fi privită ca o disciplină independentă și nu poate fi ținută între bariere create artificial.

Elevii trebuie să înțeleagă conexiunile dintre tehnologia informației și a comunicațiilor și societate, să fie capabili să se adapteze dinamicii schimbărilor determinate de aceste conexiuni.

## 2. Conținuturi și competențe pentru disciplina de informatică- filiera teoretică, profil real

Plan cadru pentru profilul real, informatică

| ARIA CURRICULARĂ / Disciplina | Clasa a IX-a |    |           | Clasa a X-a |    |           | Clasa a XI-a |    |           | Clasa a XII-a |        |                       |
|-------------------------------|--------------|----|-----------|-------------|----|-----------|--------------|----|-----------|---------------|--------|-----------------------|
|                               | T<br>C       | CD | TC+C<br>D | TC          | CD | TC+<br>CD | T<br>C       | CD | TC+<br>CD | TC            | C<br>D | T<br>C<br>+<br>C<br>D |
| TEHNOLOGII                    | 2            | 1  | 3         | 2<br>1      | 1  | 3         |              | 4  | 4         |               | 4      | 4                     |
| TIC                           | 2            |    | 2         |             |    | 1         |              |    |           |               |        |                       |
| Informatică                   |              | 1  | 1         |             | 1  | 1         | -            | 4  | 4         |               | 4      | 4                     |
| Educație antreprenorială      |              |    |           | 1           |    | 1         |              |    |           |               |        |                       |

În Colegiul Național "Gheorghe Lazar" numărul orelor de informatică (din planul cadru + ore la decizia școlii+ extracurricular) sunt :

| Clasa | Nr ore/ săptămâna |
|-------|-------------------|
| 9     | 6                 |
| 10    | 5                 |
| 11    | 11                |
| 12    | 7                 |

Conform programei școlare în vigoare la disciplina de informatică se urmărește obținerea următoarelor competențe și conținuturi

| CLASA | COMPETENȚE   | CONȚINUTURI  |
|-------|--|--|
| 9     | <ol style="list-style-type: none"> <li>Identificarea conexiunilor dintre informatică și societate.</li> <li>Identificarea datelor care intervin într-o problemă și a relațiilor dintre acestea</li> <li>Elaborarea algoritmilor de rezolvare a problemelor</li> <li>Implementarea algoritmilor într-un limbaj de programare</li> <li>Aplicarea algoritmilor fundamentali în prelucrarea datelor</li> </ol> | Algoritmi<br>Structuri de control<br>Tipuri de date (simple și structurate) în limbajul C++<br>Aplicații interdisciplinare |
| 10    | <ol style="list-style-type: none"> <li>Identificarea datelor care intervin într-o problemă și a relațiilor dintre acestea</li> <li>Implementarea algoritmilor într-un limbaj de programare</li> <li>Elaborarea algoritmilor de rezolvare a problemelor</li> <li>Aplicarea algoritmilor fundamentali în prelucrarea datelor</li> <li>Identificarea conexiunilor dintre informatică și societate.</li> </ol> | Tipuri structurate de date<br>Subprograme<br>Recursivitate<br>Aplicații interdisciplinare                                  |
| 11    | <ol style="list-style-type: none"> <li>Identificarea datelor care intervin într-o problemă și aplicarea algoritmilor fundamentali de prelucrare a acestora în teoria grafurilor</li> </ol>   | Grafuri neorientate și orientate<br>Algoritmi de prelucrare grafurilor (Parcurgerea grafurilor, Drum de cost               |

|    |  |   |
|----|--|---|
|    | <p>2. Elaborarea algoritmilor eficienți de rezolvare a problemelor<br/>                 3. Utilizarea și înțelegerea unor metode de rezolvare a problemelor (Metoda Backtracking, Metoda "Divide et impera", metoda "Programarii dinamice")<br/>                 4. Identificarea datelor ce intervin într-o problemă și aplicarea algoritmilor fundamentali de prelucrare a acestora<br/>                 5. Elaborarea algoritmilor de rezolvare a problemelor</p> | <p>minim, Arbore parțial de cost minim, etc.)<br/>                 Structuri alocate dinamic și algoritmi specifici<br/>                 Structuri arborescente și algoritmi specifici<br/>                 Modelul conceptual al problemei de gestiune (Oracle Academy)<br/>                 Tabele, Baze de date<br/>                 Relații între entități<br/><br/>                 Comenzi SQL (structura comenzilor, operații elementare)<br/>                 Realizarea unui proiect folosind Oracle SQL</p> |
| 12 | <p>1. Identificarea datelor care intervin într-o problemă și aplicarea algoritmilor fundamentali de prelucrare a acestora<br/>                 2. Elaborarea algoritmilor de rezolvare a problemelor<br/>                 3. Implementarea algoritmilor într-un limbaj de programare</p>   | <p><b>PLSQL- Oracle Academy</b><br/>                 (Tipuri de date specifice, Structuri de control, Subprograme, Triggere, excepții)<br/> <b>Visual Studio C#:</b><br/>                 Programare orientată pe obiecte (clase, mostenire, polimorfism)<br/>                 Programare vizuală (elemente de interfață utilizator, baze de date)</p>  |

Câteva imagini care evidențiază diversele activități desfășurate:

1. Folosirea SEI (Sistem Educațional Informatizat) - este un program complex inițiat de Ministerul Educației și Cercetării (MEdC), al cărui obiectiv de bază îl reprezintă susținerea procesului de predare-învățare în învățământul preuniversitar cu tehnologii de ultimă oră.

Programul sprijină obiectivele reformei educaționale în conformitate cu planul de acțiune eEurope 2005, demarat de Uniunea Europeană ca parte a inițiativei europene eLearning.

Furnizorul programului SEI este un consorțiu de companii din care fac parte: IBM și HP (pentru echipamente hardware) și SIVICO România (pentru aplicații software).

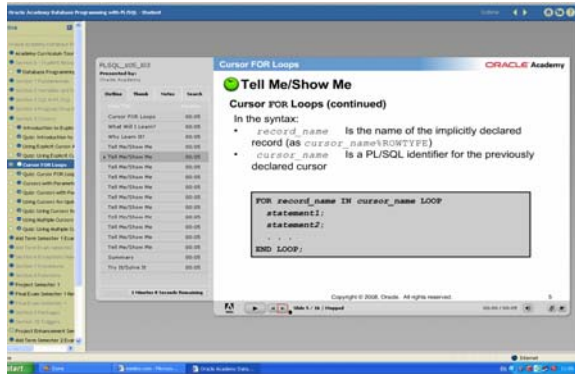
Un beneficiu major al utilizării acestui sistem este încurajarea învățământului inovativ și stimularea creativității profesorilor și elevilor, oferind un cadru general favorabil pentru dezvoltarea proiectelor și participarea beneficiarilor sistemului educațional la dezvoltarea societății informaționale. Prin folosirea de materiale didactice intuitive și interactive calitatea procesului de învățare crește. Elevul are la dispoziție nu numai ilustrații abstracte sau statice ale fenomenelor predate ci și simulări realiste și ilustrații dinamice. Aceste lecții electronice dau elevilor un puternic suport intuitiv, făcând legătura dintre materia predată și lumea reală. Prin eliminarea timpului de „scriere pe tablă”, profesorul economisește timp valoros și se poate concentra asupra predării propriu-zise.

## 2. Platforma Oracle Academy

În România, Academiei Oracle este prezentă încă din 1999. În 2003, programul a fost implementat în mai multe licee din țară. Începând cu ianuarie 2006, el a fost extins la toate școlile.

Materialul este structurat pe lecții, quiz-uri și examene pentru verificarea cunoștințelor (totul on-line). Elevii români au participat în cadrul competiției internaționale de Data Modelling organizată în cadrul Academiei Oracle, cele mai multe premii fiind câștigate de elevii din România.





## 2. BIBLIOGRAFIE

### 2. A. Tratate, monografii, cursuri universitare si alte lucrari de specialitate

### 2. B. Surse Internet

1. <http://www.edu.ro> – 6 aprilie 2009
2. <http://www.oracle.academy.com> – 5 aprilie 2009
3. <http://192.169.1.2:81/ael-> 8 aprilie 2009
4. <http://siveco.ro>- 10 martie 2009

**Preda Georgeta**  
**Colegiul Național Gheorghe Lazăr Sibiu**

**E-mail:** [gtprd@yahoo.com](mailto:gtprd@yahoo.com)

## **Desfășurarea unor activități practice, educative pentru formarea de abilități de comunicare și tehnice în secolul XXI**

Prof. Monica Oancea

*Key-Words: învățare prin proiecte, platforme de învățare, lecție virtuală*

La început de secol 21 învățământul se află în în continuă dezvoltare legată de corelarea activităților școlare de cerințele unei societăți bazată pe tehnologie. Se trece de la mediul de învățare clasic centrat pe profesor la un mediu de învățare centrat pe elev. Există o alternanță între orele clasice și cele bazate pe învățarea prin proiecte. Profesorul are în continuare rolul de organizator al activităților, respectă cerințele programei școlare, coordonează activitatea de predare-evaluare. Ce aduce nou învățarea prin proiecte este faptul că elevul are oportunitatea să interacționeze, să exploreze să gândească critic în timpul experienței de învățare. Se pune accent pe dezvoltarea abilităților secolului XXI:

1. gândire critică
2. creativitate
3. lucru în echipă
4. înțelegere multi-culturală
5. comunicare
6. folosirea tehnologiei
7. autodeterminare

Având în vedere faptul că elevii dispun de tehnologie multimedia și internet la școală și în multe cazuri și acasă, se lucrează deseori pe platforme de învățare :<http://www.thinkquest.org/en/>, <http://www.etwinning.ro/> . Acestea sunt spații virtuale în care elevi și profesori din întreaga lume crează proiecte de învățare, colaborează online și participă la competiții.

Pentru a experimenta învățarea pe bază de proiecte, am conceput o clasă virtuală găzduită pe site-ul <http://wik.is> pentru unitatea de învățare **tipul de date struct C++** la clasa a X-a.

Am avut în vedere trimiteri spre noțiunile teoretice ce pot fi studiate pe internet. S-au creat echipele. S-au stabilit reguli și termene de evaluare parțială și finală. Am postat întrebarea esențială, și tot felul de indicații legate de conținutul proiectului. Prin logarea pe site-ul <http://structuric.wik.is> elevii clasei a X-a au lucrat on-line pe teste grilă, au citit instrucțiuni și au postat proiectele lor pe echipe, au comentat proiectele colegilor și au fost evaluați conform criteriilor postate.

**În continuare țin să subliniez avantajele esențiale pe care le oferă metoda proiectului :**

- Accesul la materiale se poate face într-un mediu virtual ;
- Este favorizată dezvoltarea unei gândiri interdisciplinare ;
- Utilizarea tehnologiei dezvoltă creativitatea, capacitatea de a cerceta o temă dată, spiritul de lucru în echipă, gândirea critică și favorizează schimbul de idei.
- Sunt dezvoltate abilități ale gândirii de nivel superior ;
- Unitățile de învățare bazate pe proiect măresc gradul de implicare al elevilor și responsabilitatea acestora, îmbunătățind stilul propriu de învățare ;
- Metodele utilizate facilitează legătura dintre standardele de conținut și viața reală.
- Proiectul se bazează pe obiective aliniate standardelor specifice;
- Sunt utilizate tipuri de evaluări multiple și continue.
- Obiectivele urmăresc dezvoltarea unor competențe specifice secolului XXI.

Menționez că am folosit această tehnică de predare-evaluare în urma cursului de formare INTELTEACH desfășurat la CCD Sibiu și consider ca a fost foarte apreciată de elevi.

## BIBLIOGRAFIE

### 2.A. Tratat, monografii, cursuri universitare și alte lucrări de specialitate

Curs formare „Instruire pentru societatea cunoașterii”

### 2. B. Surse Internet

1. <http://www.thinkquest.org/en/>
2. <http://wik.is>

**Prof. Monica Oancea**

**Colegiul Național “Gheorghe Lazăr ” Sibiu**

**E-mail:** moni\_oancea@yahoo.com

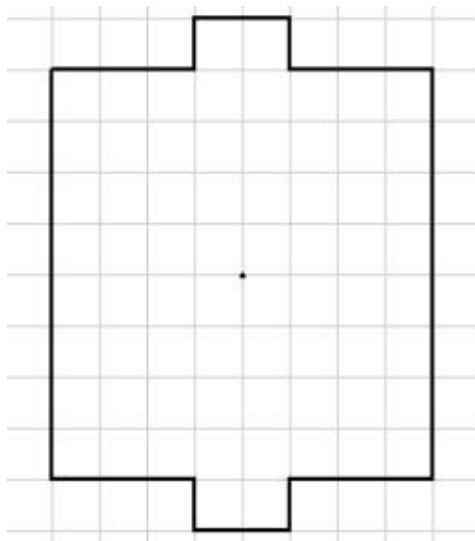
## Fotbal Logic

**Autor:** Badea Victor  
**Coordonator:** Prof. Preda Georgeta

*Cuvinte-cheie:* fotbal, picior, minge, pix, creion, foaie, joc, distracție, c#, c sharp;

### Abstract

Programul este o versiune PC a unui joc Pen&Paper popular în regiunea României și Poloniei .  
Spațiul de joc este prezentat în *Fig.1*, o mutare constând într-o linie trasă între două puncte (intersecții ale liniilor gri) .



*Fig.1: Spațiul de joc*

Completarea matricei se face astfel:

- punctele se reprezintă pe pozițiile de coordonate pare in matrice astfel:

$$teren(i, j) = \begin{cases} 1 & \text{dacă punctul este parcurs} \\ 0 & \text{în caz contrar} \end{cases}$$

- liniile verticale / orizontale se reprezintă pe pozițiile de coordonate impare astfel:

$$teren(i, j) = \begin{cases} 1 & \text{dacă linia verticală / orizontală este parcursă} \\ 0 & \text{în caz contrar} \end{cases}$$

- liniile oblice se reprezintă pe pozițiile de coordonate par / impar astfel:

$$teren(i, j) = \begin{cases} 1 \text{ dacă este parcursă diagonala secundară} \\ 2 \text{ dacă este parcursă diagonala principală} \\ 3 \text{ dacă sunt parcurse ambele diagonale} \\ 0 \text{ în caz contrar} \end{cases}$$

Matricea inițială este reprezentată în tabelul 1.

Tabel 1. Matricea teren[,]

|    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|
| 1  | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  |
| 2  | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  |
| 3  | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  |
| 4  | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0  | 0  | 0  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  |
| 5  | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0  | 0  | 0  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  |
| 6  | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0  | 0  | 0  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  |
| 7  | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  | 3  | 3  |
| 8  | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  | 3  | 3  |
| 9  | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  | 3  | 3  |
| 10 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  | 3  | 3  |
| 11 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  | 3  | 3  |
| 12 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 1  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  | 3  | 3  |
| 13 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  | 3  | 3  |
| 14 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  | 3  | 3  |
| 15 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  | 3  | 3  |
| 16 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  | 3  | 3  |
| 17 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  | 3  | 3  |
| 18 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  | 3  | 3  |
| 19 | 3 | 3 | 3 | 0 | 0 | 0 | 0 | 0 | 0 | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 0  | 3  | 3  | 3  |
| 20 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0  | 0  | 0  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  |
| 21 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0  | 0  | 0  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  |
| 22 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 0  | 0  | 0  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  |
| 23 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  |
| 24 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  |
| 25 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  | 3  |

## 2. BIBLIOGRAFIE

### 2.A. Tratate, monografii, cursuri universitare si alte lucrari de specialitate

1. GĂLĂȚAN, Constantin, *C# pentru liceu*, Editura L&S Info-mat, București, 2008.

### 2. B. Surse Internet

1. <http://msdn.microsoft.com/en-us/vcsharp/default.aspx>

**Badea Victor**

**Colegiul Național Gheorghe Lazăr**

**XII G**

**Matematică-Informatică**

vkt40000@yahoo.com

## Graful hamiltonian, program oprire calculator

**Autori:** Cherciu Dorin-Ionut  
Preda Georgeta

*Abstract:* This is a sample of the format of your full paper. Use Word for Windows (Microsoft) (or equivalent Word Processor with exactly the same "printing results") by tuning -1.3 cm from right and -1.3 cm from left in the Microsoft Word package, or equivalently by keeping 1.8 cm, real distance, from right and 1.8 cm, real distance, from left. Use single space. Use double-column format after the Key-Words. Use 11pt, Times New Roman. For the words Abstract, Key-Words and References use Italics.

*Key-Words:* graf, hamiltonian, ciclu hamiltonian, opreste, shutdown, restart, log off, calculator, computer

### 1. Abstract extins

Primul proiect consta in gasirea tuturor ciclurilor hamiltoniene ale unui graf dat. Ciclu hamiltonian este un ciclu elementar care contine toate nodurile grafului.

Conditii de suficienta : Teorema lui Dirac: Fie  $G$  dat prin perechea  $(A,B)$ . Daca  $G$  are un numar de cel puțin 3 varfuri astfel incat gradul fiecarui nod respecta conditia  $d(x) \geq n/2$ , atunci graful este hamiltonian.

Aceasta se realizeaza prin folosirea unui algoritm backtracking adaptat corespunzator problemei. Implementarea lui s-a realizat in C++ fiind reprezentat cu grafica(Fig.1).

Proiectul al doilea consta in implementarea unui algoritm de stingere, repornire sau delogare a calculatorului dupa un anumit numar de minute sau secunde date de utilizator. Programarea este orientata pe obiecte si este realizata in C#(Fig.2).Proiectele sunt puse intr-o pagina html.

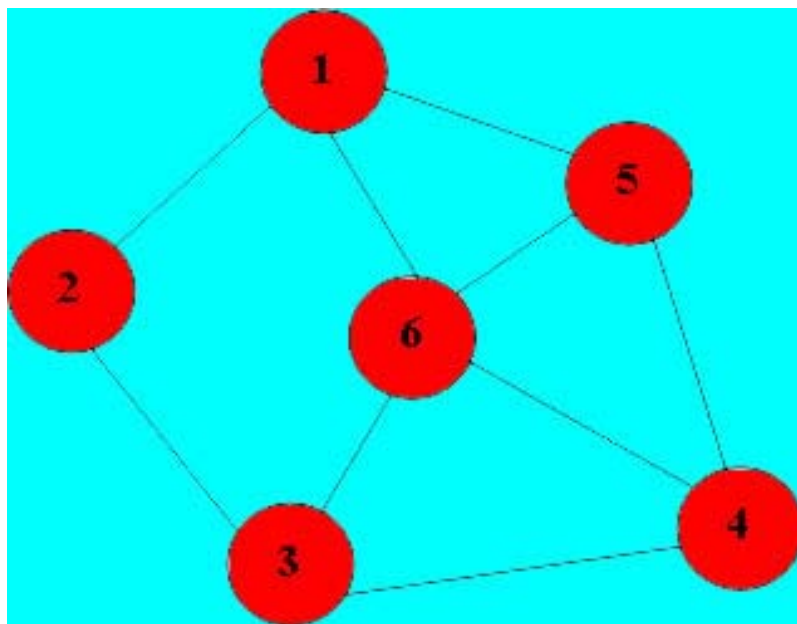


Fig.1

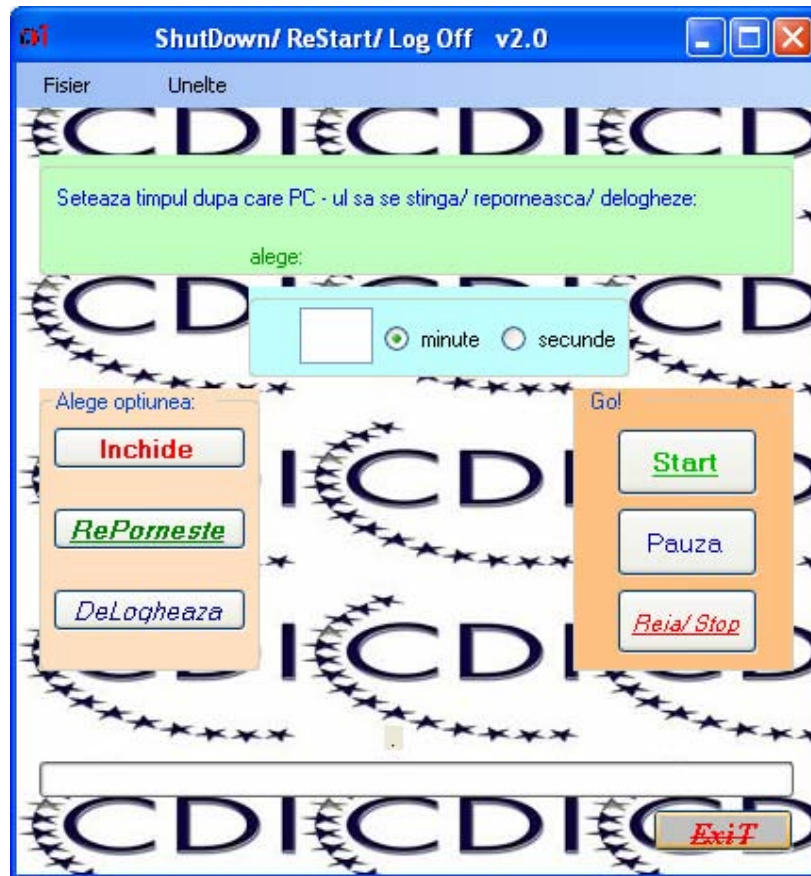


Fig.2

Motivatia a fost de a crea un program propriu implementat dupa o idee proprie, deci configurat dupa placul meu, dar care sper sa placa si altora.

## 2. Bibliografie

### 2.A.

1. Manual C#, Autori Constantin Galatan, Susana Galatan

### 2.B. Surse Internet

1. <http://cppworld.warpfive.net/cls11/grafuri/teorie.html>

**Cherciu Dorin-Ionut**  
**Colegiul National „Gheorghe Lazar” Sibiu**  
**Matematica-Informatica**  
 cdionut1990@yahoo.com

## Data Modeling: Environmental Friendly Technology

**Autori:** Cristian Alexandru, Radomir Mareș  
Prof. Oancea Monica

*Abstract:* In this project there are presented ideas and informational need of two companies, Plank-Corp and Eco-Evolution, that will help make the world a better place. The first company deals with producing O<sub>2</sub> from CO<sub>2</sub> using plankton, the second company is a solar panel retailer which promotes the idea of “green living”.

*Key-Words:* Plank-Corp, Eco-Evolution, carbon footprint, plankton, solar panel, environment, ERD, entities, attributes, relationship, normal form

### Extended abstract

The goal of our project was to document, present and asses the informational needs of two companies and build an ERD (entities relationship diagram) from witch a future database program would be created. Starting from the base ideas of each company a plan of development and business was made:

#### Plank – Corp

In a world where pollution is continuously growing, and the global warming is reaching dangerous heights, the humankind is searching for a way to reduce the quantity of toxic gasses like carbon.

We, here at “Plankton Corp.”, have a well defined goal: to provide every company that effuses a quantity of toxic gasses, much bigger than the law admits the possibility to convert carbon to oxygen. We had the idea to balance the difference between carbon dioxide and oxygen by offering a new kind of natural technology, plankton. Plankton through pphotosynthesis eliminates carbon from the atmosphere and produces oxygen. The factories will use our services because the amount of toxic gasses emitted is limited by law, so if they want to continue their work, the representative agents will start new contracts with our company.

#### Eco-Evolution

We promote the concept of green energy by building an international company that makes retail with solar panels manufactured all over the world.

Such a complex system requires a strong control, provided by our well organized, flexible and efficient database, meeting all our company’s needs.

After assessing the needs of both companies we designed and ERD and integrated it in a detailed presentation of our imaginative business.

### References

1. <http://www.washingtonpost.com/wpdyn/content/article/2007/07/19/AR2007071902553.html>  
- a website which contains information about a similar company
2. <http://en.wikipedia.org/wiki/Plankton>  
- detailed information about what plankton means, and its role in nature
3. <http://www.solarhome.org/solarpanelscenter.html>
4. [http://en.wikipedia.org/wiki/Carbon\\_footprint](http://en.wikipedia.org/wiki/Carbon_footprint)



Cristian Alexandru  
Radomir Mares

**Cristian Alexandru**  
**Colegiul National „Gheorghe Lazar” Sibiu**  
**Matematica – Informatica**  
[cristianalex81@gmail.com](mailto:cristianalex81@gmail.com)

**Radomir Mares**  
**Colegiul National „Gheorghe Lazar” Sibiu**  
**Matematica – Informatica**  
[maresh2all@yahoo.com](mailto:maresh2all@yahoo.com)

## Teoria grafurilor reprezentată grafic în C++

**Autori:** Cristian Alexandru, Diaconescu Bogdan, Radomir Mares  
Prof. Oancea Monica

*Abstract:* În această lucrare sunt prezentate într-un mod interactiv câteva din noțiunile de bază ale grafurilor. Pe lângă partea teoretică pe care o reprezintă algoritmi de bază a grafurilor pentru o mai bună înțelegere a funcționării acestora am reprezentat grafic fiecare pas al algoritmului cu ajutorul mediului de programare C++.

*Key-Words:* grafuri, noduri, muchie, parcurgere, drum, ciclu, componentă conexă, bipartit,

### Abstract extins

Lucrarea constă într-o prezentare PowerPoint a algoritmilor de bază pe grafuri și aplicații C++ corespunzătoare.

S-au avut în vedere noțiunile fundamentale : nod, muchie, grad, lanț, ciclu, componente conexe.

Pe lângă noțiunile elementare am studiat și parcurgerile Depth First, Breadth First și am realizat programe ce simulează în mod grafic parcurgerea grafului.

Pentru testarea proprietății de graf hamiltonian am folosit tehnica de programare backtracking.

Pentru testarea proprietății de graf eulerian am folosit condițiile necesare și suficiente ale teoremei, verificând conexitatea grafului și paritatea gradelor nodurilor.

De asemenea am implementat testarea noțiunii de graf bipartit ce separă mulțimea nodurilor în două mulțimi disjuncte a căror reuniune este mulțimea inițială. Dacă fiecare muchie are extremitățile în cele două mulțimi atunci graful este bipartit.

În toate aceste probleme se regăsește algoritmul de desenare a unui graf citit din fișierul de intrare. Am găsit un mod eficient de reprezentare a grafului cu ajutorul ecuației carteziene a cercului, pe care se dispun omogen toate nodurile grafului. În funcție de cerința problemei am sincronizat algoritmul de rezolvare a problemei cu reprezentarea grafică a rezultatelor. Animația s-a realizat prin desenări și ștergeri succesive.

Lucrarea reprezintă o modalitate de a recapitula noțiunile legate de grafuri într-un mod intuitiv, putând fi folosită ca material auxiliar în înțelegerea și fixarea algoritmilor studiați.

### 2. Bibliografie

1. **CERCHEZ, Emanuela, ȘERBAN, Marinela, Programarea în limbajul C/C++ pentru elevi**, Editura Polirom, 2006
2. **MILOȘESCU, Mariana, Manual pentru clasa a XI-a Informatică Intensiv**, Editura didactică și pedagogică, 2006

**Cristian Alexandru, Diaconescu Bogdan, Radomir Mares**  
**Colegiul Național „Gheorghe Lazăr” Sibiu**  
**Matematică – Informatică**

cristianalex81@gmail.com, d\_bogdi@yahoo.com, maresh2all@yahoo.com

## Fields of Gold

**Autori:** Bonci Eduard, Bulat Flaviu, Făgărășan Luana, Iaru Claudiu  
Coordonator: Preda Georgeta

*Cuvinte-cheie:* afacere, client, furnizor, servicii, meteorologie, agricultură,

### Abstract

În cadrul unui curs de baze de date, am participat la un concurs de idei de afaceri, ce avea ca scop crearea și explicarea diagramei bazei de date a unei firme ce se bazează pe date meteorologice și explicarea conceptului de operare al acesteia.

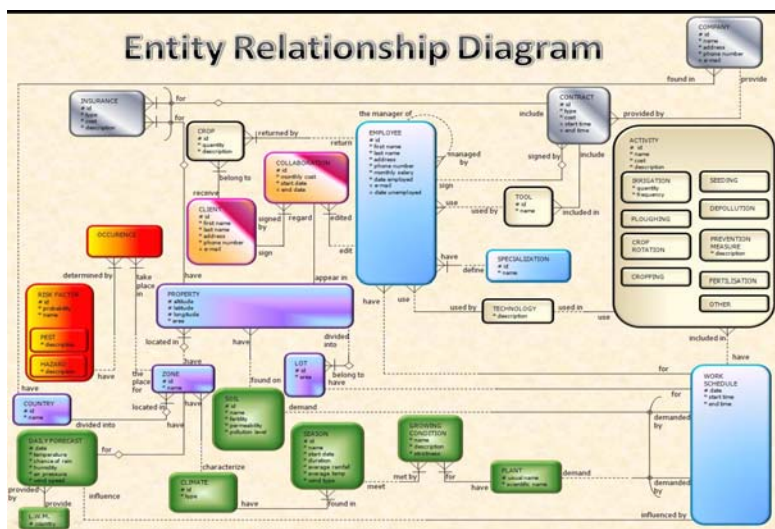


Fig. 1: Diagrama bazei de date

Într-o sesiune de brainstorming am identificat necesitățile afacerii și am creat această diagramă, colorată în funcție de categoriile de entități. Firma imaginată de echipa noastră se ocupă cu cultivarea terenurilor clienților. Conceptul este simplu, dar eficient, luând în seamă datele meteo, în funcție de care sunt programate lucrări agricole pentru fiecare teren. O parte din recoltă este reținută de firmă ca plată, iar restul este restituit proprietarului. Obiectivul principal a fost implementarea cu succes a datelor meteorologice într-o afacere profitabilă, iar acest lucru a fost realizat, folosind date diverse, necesare funcționării eficiente a afacerii.

**Iaru Claudiu**  
Colegiul Național „Gheorghe Lazăr”  
Profil: Matematică - Informatică  
iclam90n@yahoo.com