REALIZAREA UNEI APLICAȚII DE REPREZENTARE GRAFICĂ A GRAFURILOR NEORIENTATE ȘI COLORARE PRIN METODA DE NUMĂRARE PÒLYA

THE DEVELOPMENT OF AN APPLICATION THAT GRAPHICALLY REPRESENTS UNDIRECTED GRAPHS AND COLORS NODES USING PÒLYA'S COUNTING METHOD

MIHALACHE Andrei,
Facultatea: Ingineria și Managementul Sistemelor Tehnologice,
Specializarea: Modelarea și Simularea Sistemelor Mecanice Mobile,
Anul de studii: I, e-mail: andrei@mdata.ro

Conducător științific: Șl.dr.ing. **Ileana DUGĂEȘESCU**, Departamentul TMR

REZUMAT: Graphs are used successfully to model several types of specific problems like circuit analysis, finding the shortest route, optimizing production processes, analyzing project planning, criticism of literary texts, social networks, applications in chemistry and economics, but the objective of this paper is the modeling of mobile structures. This article uses conceptual and logical models to present theoretical aspects and presents a client-server application that can be accessed from any browser using PHP and JavaScript programming languages with data stored in a Maria DB database. The reason for choosing this implementation is determined by the interpreter that runs without previous compilation and do not emphasize on the physical model of data representation.

CUVINTE CHEIE: grafuri neorientate, reprezentare grafică, colorare

1. Introducere

În contextul ingineriei software definim un *model de date* ca fiind un model abstract care descrie modul în care datele sunt reprezentate și accesate. Într-un astfel de model sunt definite în mod formal datele și legăturile dintre acestea și se prezintă în mod explicit semnificația datelor, rezultând date structurate [2].

Un ***model*** este definit ca o abstractizare simplificată a unei realitățiii complexe, punând în evidență elementele esențiale și ignorând detaliile.

2. Stadiul actual

Termenul de ***dată*** se referă la fapte care caracterizează obiecte sau evenimente ce pot fi înregistrate și stocate într-un sistem informatic și au semnificație și înțeles în mediul utilizatorilor.

Modelarea datelor constituie o metodă utilizată în definirea și analiza cerințelor datelor necesare în procesele de lucru ce sunt desfășurate în organizații. Atunci când aceste date sunt salvate, de cele mai multe ori este aleasă o bază de date relațională datorită performanțelor și accesibilității, întrucât datele sunt mai ușor de înțeles.

Modelul de date reprezintă organizarea logică a structurilor de date corespunzătoare obiectelor din realitate care prezintă un interes pentru aplicație. De asemenea acest model include și constrângerile din lumea reală și relațiile dintre obiecte.

Conform [1] modelele de date fac parte din unul din cele trei tipuri de modele de date: modele conceptuale, modele logice sau modele fizice.

Modelele *conceptuale* descriu semantica domeniului și conțin entități ale claselor care reprezintă elemente de interes pentru domeniu și afirmații despre asocierile existente între aceste entități. Acestea sunt reprezentate prin schema conceptuală.

Modelele *logice* descriu structura logică a datelor și pot conține descriptori ai colecțiilor, atributelor, marcatorilor XML etc. Modelele logice sunt reprezentate conform cerințelor impuse de o anumită tehnologie de implementare sub forma unei scheme logice.

Modelele *fizice* descriu modul în care sunt stocate datele. Aceste modele conțin fișiere, partiții, spații de tabelă, indecși etc. și se reprezintă prin schema internă sau fizică.

Vom aborda tema folosind doar modelele *conceptual* și *logic*, explicând secvențele de cod folosite în programarea aplicației.

3. Definirea grafurilor

Un **graf** este o pereche ordonată de mulțimi $G=(X, U)$, unde Mulțimea $X=\{x_1, x_2, ..., x_n\}$ este o mulțime nevidă și finită de elemente denumite ***noduri*** sau ***vârfurile*** grafului, iar Mulțimea U este o mulțime de perechi de vârfuri din graf [4].

În cazul grafurilor neorientate, perechile de vârfuri din mulțimea X sunt neordonate de forma $[x_i, x_j]$, unde $i \neq j$ și x_i, $x_j \in X$ și sunt denumite ***muchii***. Vârfurile x_i și x_j se numesc ***extremitățile muchiei*** $[x_i, x_j]$ și sunt etichetate în aplicație cu numere naturale de la 0 la $n-1$ (unde cu n vom nota numărul de vârfuri din graf) [4].

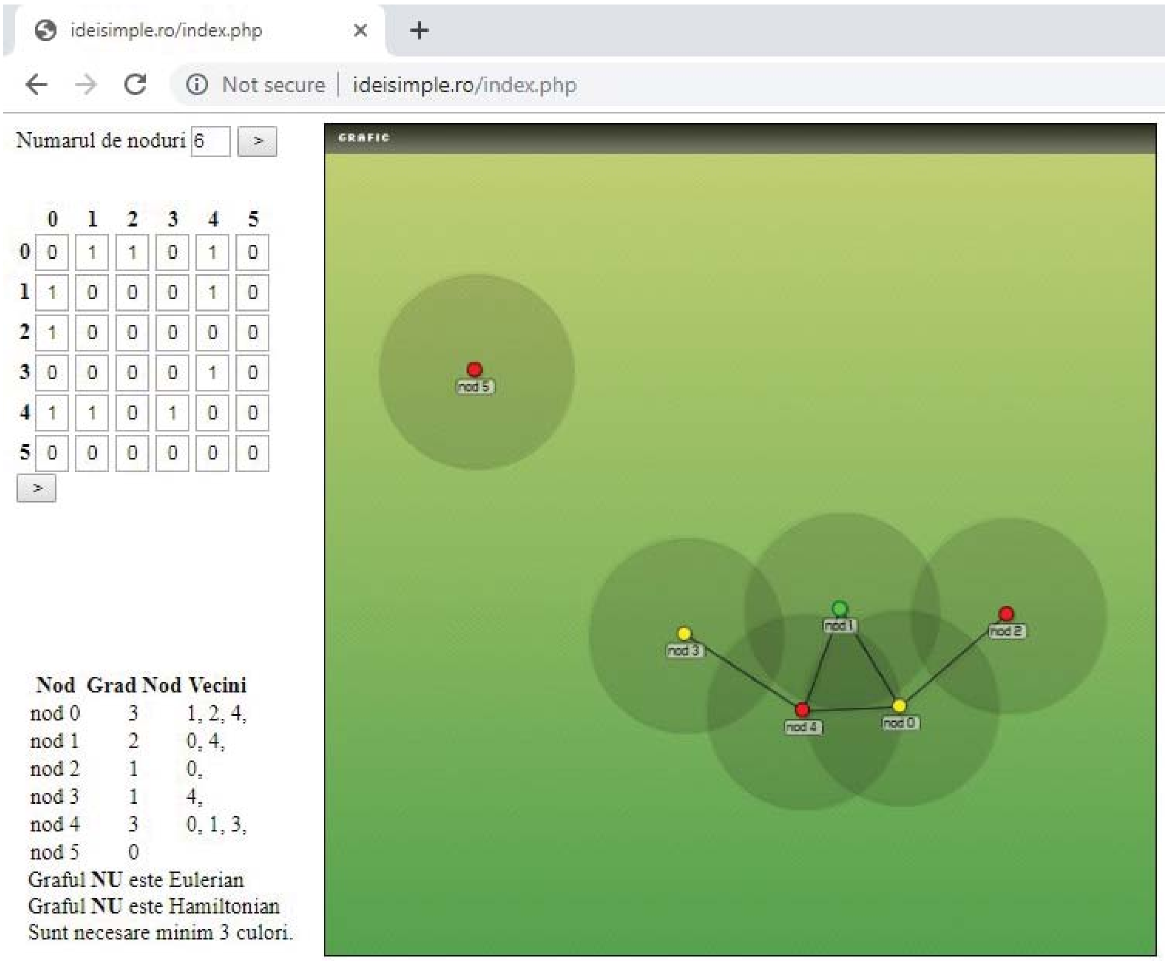

Fig. 1. Graf neorientat cu 6 noduri introdus prin matricea de adiacență

Dacă există un arc sau o muchie cu extremitățile x_i, $x_j \in X$, atunci vârfurile x_i și x_j sunt ***adiacente***; fiecare extremitate a unei muchii/unui arc este considerată incidentă cu muchia/arcul respectiv[4].

4. Reprezentare grafică

Un graf neorientat poate fi reprezentat vizual sub forma unei figuri geometrice alcătuite din puncte care reprezintă ***nodurile*** sau ***vârfurile*** și linii drepte sau curbe care unesc aceste puncte, numite ***muchii*** sau ***arce***. [4]

Dacă graful este neorientat, vom reprezenta fiecare muchie ca o linie (dreaptă sau curbă), care unește cele două extremități ale muchiei. [4]

Se numește ***grad*** al unui vârf *x* numărul de muchii incidente cu vârful respectiv. Gradul vârfului x se notează ***d(x)***. [4, 5]

De exemplu, pentru graful din Fig. 2, am reprezentat în Tabelul 1 gradele vârfurilor.

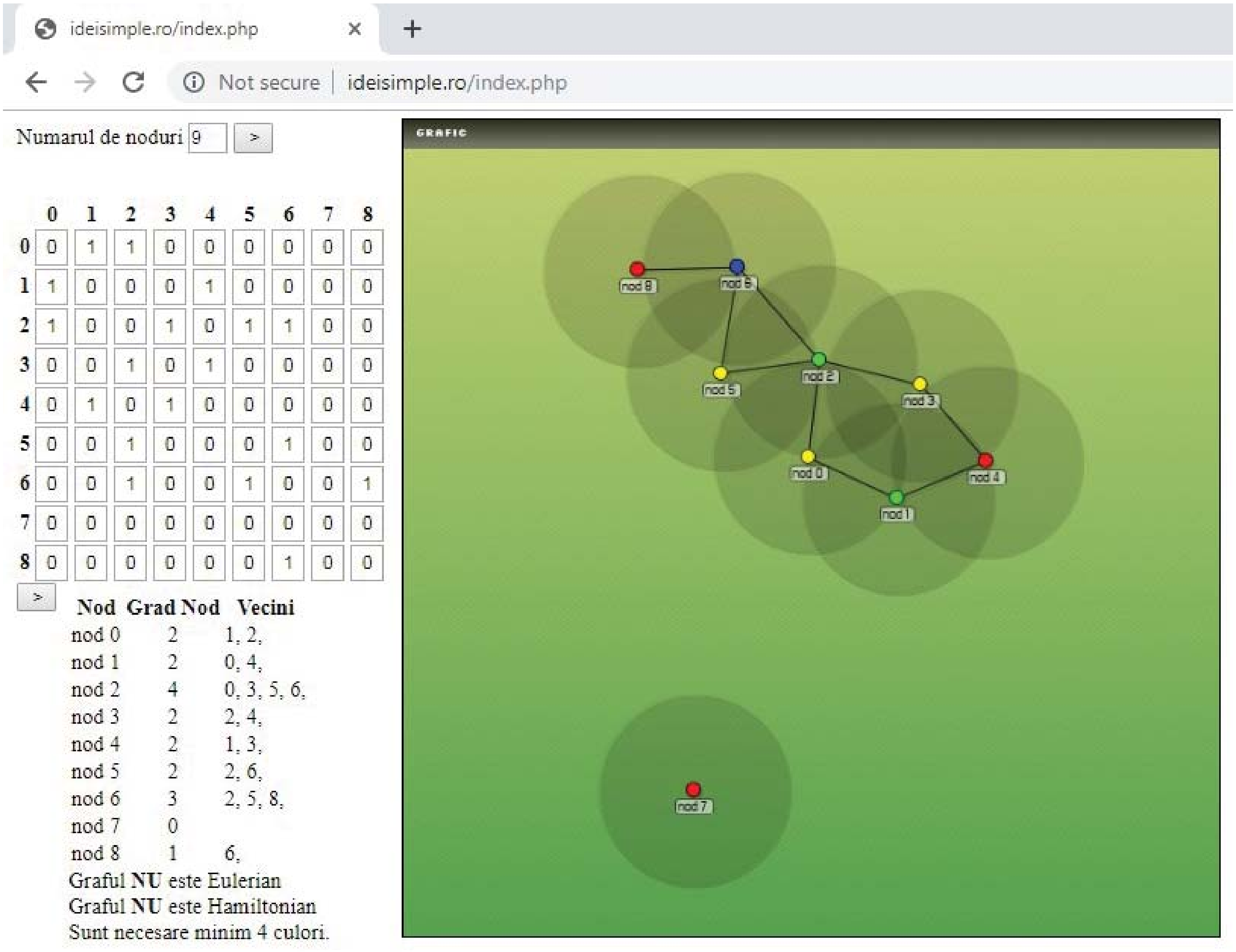

Fig. 2. Graf neorientat cu 9 noduri introdus prin matricea de adiacență

Tabelul 1. Gradul vârfurilor unui graf neorientat

x	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
d(x)	2	2	4	2	2	2	3	0	1

Suma gradelor unui graf neorientat este egală cu dublul numărului de muchii din graf.

Se numește ***lanț*** într-un graf neorientat, o secvență de vârfuri $\{x_1, x_2, ..., x_p\}$, cu proprietatea că oricare două vârfuri consecutive din secvență sunt adiacente. [4]

Un lanț este ***elementar*** dacă el nu conține de mai multe ori același vârf. [4] Un lanț este ***simplu*** dacă el nu conține de mai multe ori aceeași muchie. [4] Se numește ***ciclu*** un lanț simplu pentru care extremitatea inițială coincide cu extremitatea finală. Ciclul se numește ***elementar*** dacă nu conține de mai multe ori același vârf (exceptând extremitățile sale). [5]

5. Implementarea aplicației

Pentru implementarea aplicației, la prima accesare s-a folosit un generator automat de graf care creează un graf cu minim 4, maxim 11 noduri.

Parametrii de intrare sunt numărul de noduri și matricea de adiacență.

Accesarea aplicației se face direct, în orice browser, pe numele de domeniu ideisimple.ro si permite salvarea imaginii, gradului, listei vecinilor pentru fiecare nod și verificarea dacă este sau nu Eulerian sau Hamiltonian.

Se numește *ciclu eulerian* într-un graf, un ciclu care conține toate muchiile grafului. Un graf este ***eulerian*** dacă conține un ciclu eulerian. Pentru a implementa in aplicație am folosit teorema: Un graf fără vârfuri izolate este *eulerian*, dacă și numai dacă este conex, și gradele tuturor vârfurilor sunt numere pare.

Se numeste *ciclu hamiltonian* într-un graf, un ciclu elementar care conține toate vârfurile grafului. Un graf este ***hamiltonian*** dacă conține un ciclu hamiltonian. Se numește lanț hamiltonian într-un graf, un lanț elementar care conține toate vârfurile grafului. Pentru a implementa în aplicație am folosit teorema: Daca într-un graf $G=(X,U)$ cu $n>=3$ vârfuri, gradul fiecarui vârf x verifica conditia $d(x)>=n/2$, atunci graful este hamiltonian. [6]

Tabelul 2. Cod sursă pentru a verifica tipul de graf

```
      $esteEulerian = 1;
      $esteHamiltonian = 1;
      $n2 = intval($n/2);

for ($i=0; $i<$n; $i++){
$grad = 0;
      for ($j=0; $j<$n; $j++){
            if ($adiacenta[$i][$j]==1) {
                  $grad++;
            }
      }
if (($grad %2 != 0) || ($grad == 0))
            $esteEulerian = 0;
if ($grad < $n2)
            $esteHamiltonian = 0;
}

if ($esteEulerian)
      echo 'Graful este Eulerian<br/>';
else
      echo 'Graful NU este Eulerian<br/>';

if ($esteHamiltonian)
      echo 'Graful este Hamiltonian<br/>';
else
```

```
        echo 'Graful NU este Hamiltonian<br/>';
```

Aplicația oferă posibilitatea modificării numărului de noduri, dar și a matricei de adiacență, asigurând restricția ca matricea să fie simetrică față de diagonala principală și valorile din matrice să fie reduse la 0 şi 1, indiferent ce valoare introduce utilizatorul.

Pentru un prim test am introdus numărul ca date de intrare 7 noduri pentru care am obținut o soluție posibilă reprezentată în Fig. 3, dar si confirmarea faptului că este un graf eulerian.

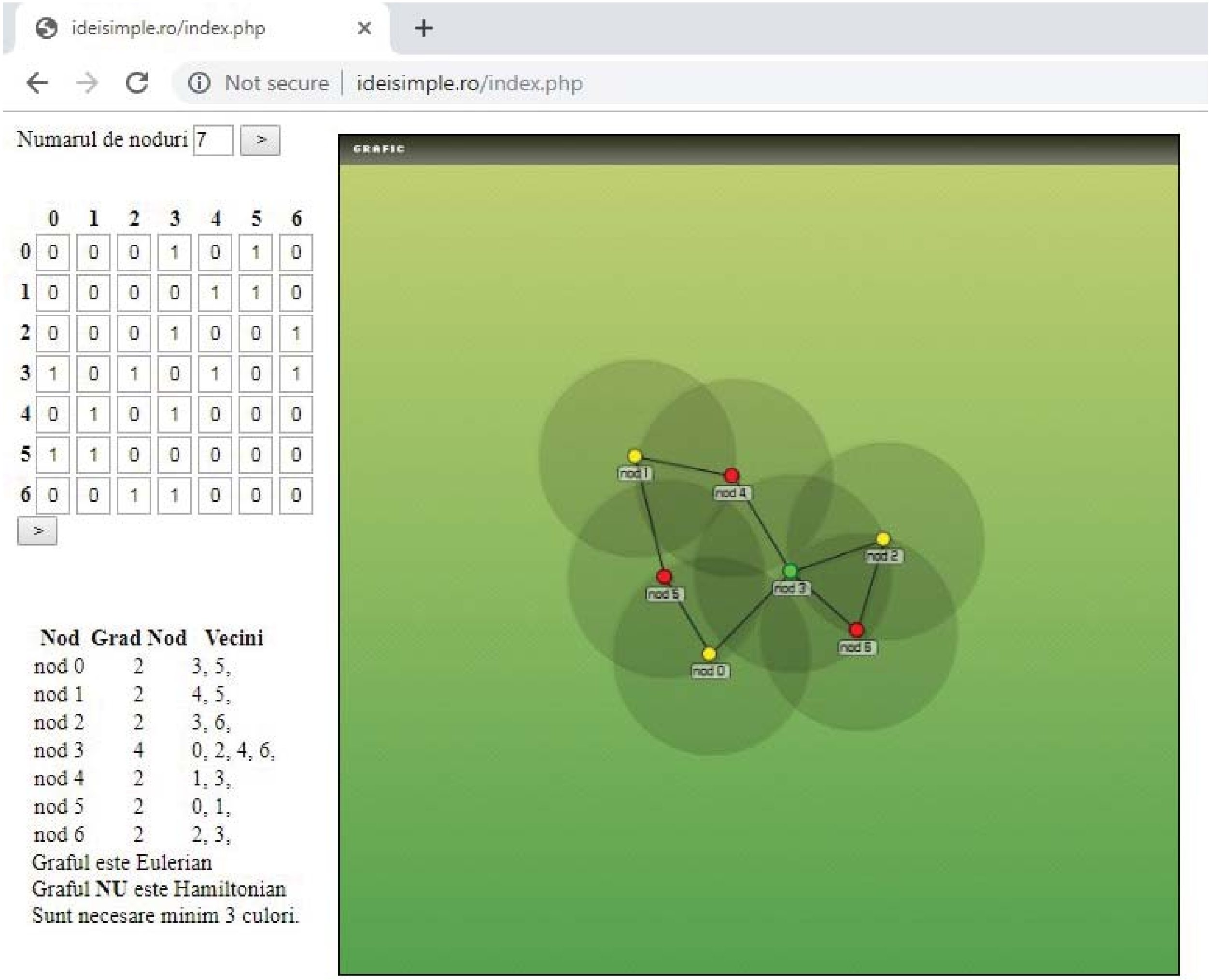

Fig. 3. Verificarea tipului de graf Eulerian sau Hamiltonian

6. Colorarea nodurilor

Colorarea validă a nodurilor unui graf est validă dacă oricare două noduri vecine sunt colorate distinct. Formula de numărare a lui Pòlya determine numărul a(n_1, n_2,..., n_m) de colorări distincte în raport cu permutările unui grup de simetrii G, dacă avem constrângeri si trebuie să folosim culoarea y_1 de exact n_1 ori, culoarea y_2 de exact n_2 ori, ..., și culoarea y_m de exact n_m ori, cu menționea *că* $n_1 + n_2 + \dots + n_m = n$. [7]

Pòlya a descoperit o formulă de calcul direct al polinomului [7]

$$F_G(y_1, y_2, \dots, y_m) = \sum_{n_1+n_2+\dots+n_m} a_{(n_1, n_2, \dots, n_m)} y_1^{n_1} y_2^{n_2} \dots y_m^{n_m} \qquad (1)$$

Acest polinom se numește **inventar de metode de colorare**, iar formula de numărare a lui Pòlya este: [7]

$$F_G(y_1, y_2, \ldots, y_m) = P_{G\left(\sum_{i=1}^{m} y_i, \sum_{i=1}^{m} y_i^2, \ldots, \sum_{i=1}^{m} y_i^m,\right)} \tag{2}$$

7. Concluzii

Contribuțiile originale din cadrul lucrării contau în generarea dinamică a reprezentării graficelor neorientate, colorarea validă a grafului fără a repeta culoarea pe două noduri vecine, iar în cercetările viitoarele vor fi aprofundate și generate toate grupurile de graficele colorate folosind numărarea echivalentă definită de Pòlya.

8. Bibligrafie

[1] American National Standards Institute, *ANSI/X3/SPARC Study Group on Data Base Management Systems*; Interim Report. FDT, ACM SIGMOD bulletin. Volume 7, No. 2, 1975;

[2] Lungu I., Bâră A., Bodea C, Botha I., Diaconiţa V., Florea A., Velicanu A., *Tratat de baze de date – Baze de date. Organizare. Proiectare. Implementare*, 2011, ISBN 978-606-505-481-3;

[3] Comănescu A., Comănescu D.M., Dugăeşescu I. şi Boureci A. — *Bazele modelării mecanismelor*, POLITEHNICA Press, București, 2010;

[4] Berge C.: *Teoria grafurilor şi aplicaţiile ei* (traducere din limba franceză), Ed. Tehnică, Bucureşti, 1969.

[5] Mirescu P., Roşu Al.: *Teoria grafurilor*, Editura militară, Bucureşti, 1968.

[6] Dobrjanskyj L., Freudenstein F.: *Some applications of Graph Theory to the Structural Analysis of Mechnaisms*. In: Journal of Engineering for Industry, 89, Series B, ASME 1967, pp 153 - 158.

[7] G. Chartrand, L. Lesniak, *Graphs and Digraphs*, 3rd Ed., Chapman and Hall, London, 1996.