International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:7, No:4, 2013

Courses Pre-Required Visualization Using Force Directed Placement Technique

Imen Ammari, Mourad Elloumi, and Ala Eddine Barouni

***Abstract*—**Visualizing "*Courses – Pre – Required - Architecture*" on the screen has proven to be useful and helpful for university actors and specially for students. In fact, these students can easily identify courses and their pre required, perceive the courses to follow in the future, and then can choose rapidly the appropriate course to register in. Given a set of courses and their pre-required, we present an algorithm for visualization a graph entitled "*Courses-Pre-Required-Graph*" that present courses and their pre-required in order to help students to recognize, lonely, what courses to take in the future and perceive the contain of all courses that they will study. Our algorithm using "*Force Directed Placement*" technique visualizes the "*Courses-Pre-Required-Graph*" in such way that courses are easily identifiable. The time complexity of our drawing algorithm is O (n^2), where n is the number of courses in the "*Courses-Pre-Required-Graph*".

***Keywords*—**Courses–Pre-Required-Architecture, Courses-Pre-Required-Graph, Courses-Pre-Required-Visualization, Force directed Placement, Resolution.

I. Introduction

THIS paper deals with the visualisation of course pre required architecture for students. It helps them along their registration for choosing courses to study. Our algorithm proposes an automatic visualization in graphic forms of the totality of the courses and their pre-required. The challenge of our algorithm is to produce a clear visualization of the graph with minimum crossing for an important number of courses. The problem of drawing "*Course-Prerequired-Architecture*" is classified to the subject of graph drawing in general and to the problem of visualization courses pre-required in particular for students. Many information visualization systems require graphs to be drawn so those are easy to read and understand for the students. The visualization of complex conceptual structures is a key component of supports tools for many applications in science and engineering stated in [6]-[10]. Examples include software engineering (data flow diagrams [8]), databases (entity-relationships diagrams [7]), information systems (pert diagrams [9]). Further applications can be found in other science and engineering disciplines, such as medical science (concept lattices), biology (evolutionary trees), chemistry (molecular drawings), civil engineering (floorplan maps), and cartography (map schematics), Survivable telecommunication networks (multi-ring architecture drawings) [3]-[4].

In this paper we consider an application related to visualize a special graph that represents courses and their pre-required for students in order to recognize their future courses and their pre-required. The design and analysis of course pre-required architecture is a very important problem [1]. For that matter, computer tools to aid the design and analysis of "*Course-Pre-Required-Architecture*" are in great demand. A central problem of such tools is how to draw a courses and their pre-required on the computer screen so that important aspects of the "*Course-Pre-Required-Architecture*" can be easily captured and an improved solution can be obtained by a student interactively.

This algorithm allows drawing automatically on the screen courses and their pre-required as a graph. The visualization of the graph allows students to know the details of these courses as well as the complete path of courses. The algorithm generates a clear pattern with minimal crossing to a large number of courses. In fact we have define the graph "*Courses-Pre-Required-Graph*" that represent courses and their pre-required. The nodes of "*Courses-Pre-Required-Graph*" represent the set of courses and the edges represent the pre-required relation between two courses. Our algorithm produces a clear visualization of the graph with a minimum crossing between edges and nodes. The rest of the paper is organized as follows: In section II, we discuss the criteria for visualization a "*Courses-Pre-Required-Graph"*, Section III is devoted to the description of the "*Courses- Pre-Required-Visualization*" algorithm. Section IV is devoted to the description of empirical results of our algorithm. In section V we summarize the results and discuss some open problems.

II. Preliminaries

Similar to graph drawing, many criteria should be considered in order to enhance the readability. Usually, a general optimization method is used, i.e., minimize the crossings, the area, the number of bends (in orthogonal drawings), and the number of slopes (in polyline drawings) [2]-[5]. We define "*Courses-Pre-Required-Graph*" G=(V,E) with a set of nodes V (representing the courses) and a set of links E (representing the pre-required links between courses). The "*Resolution*" is a constant defined by the user which is considered to be the minimal distance between any two nodes

I.Ammari Assistant with the High Institute of Technological Studies of Radès of Tunisia. Laboratory of Information Technologies and Communication and Electrical Engineering (LaTICE). (e-mail: AmmariImen@gmail.com).

M.Elloumi Professor with the Faculty of Economic Sciences and Management of Tunis (FSEG Tunis) (LaTICE). (e-mail: Mourad.Elloumi@fsegt.rnu.tn).

A.Barouni Assistant Professor with the high school of Technology and Computer Science (ESTI Tunis) (LaTICE). (e-mail: Ala.barouni@fst.rnu.tn).

in the drawing of the graph G. The resolution rule is that the nodes must be kept far enough from each other, so that, the human eye can tell them apart. The resolution rule prevents the drawing algorithm from arbitrary scaling down the picture.

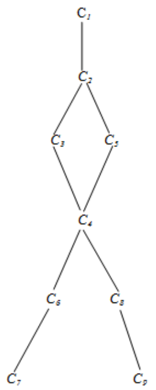


Fig. 1 Example of Courses Pre-Required Graph

III. Algorithm

Our algorithm entitled “*Courses-Pre-Required-Visualization*” uses “*Force Directed Placement*” Technique based on a mechanical system where the nodes of the "*Courses-Pre-Required-Graph*" transformed on charged particles and the edges to springs[11]-[12]. The particles practice each other repulsive forces and the springs apply two kinds of forces attractive or repulsive forces in order to bring them to their ideal positions. The algorithm stopped visualization of graph until to reach a minimum energy state where the sum of all forces in each node equals to zero; this corresponds to the equilibrium position.

For nodes connected by edges which replaced by springs exert attractive or repulsive forces according to their current length over the initial length of the spring. For nodes that are not connected by edge exert only repulsive force:

(1) *Fs*: attractive or repulsive forces exerted by the springs between neighbors nodes which represent the courses.

(2) *Fr*: repulsive forces between every pair of non-neighboring nodes.

The formula to calculate these forces are as follow:

$$F_s = C_1 * \log(\frac{d}{C2}) \tag{1}$$

$$Fr = \frac{resolution}{d^2} \tag{2}$$

where we denote by d the distance between a pair of courses, C1 and C2 are constants, and the resolution present the minimum distance between any two nodes in the drawing of the "Courses-Pre-Required-Graph".

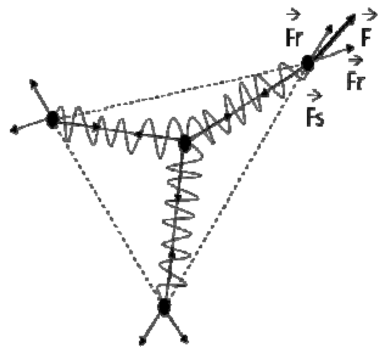


Fig. 2 The "Courses-Pre-Required-Graph" represented by a mechanical system using Force Directed Placement Technique

The formula of the calculation of the force $\vec{F}$ for each nodes of V is presented as follow:

$$\vec{F} = \sum \vec{Fs} + \sum \vec{Fa} \tag{3}$$

We present our “*Courses-Pre-Required-Visualization*” algorithm that visualizes the “*Courses-Pre-Required-Graph*” clearly.

Algorithm “*Courses-Pre-Required-Visualization*”

Input : Graphe G

Output : Visualization of the graphe G on the screen

Begin

(i) Place the nodes of G on the screen in random way

(ii) **For i=1 To M Do**

- Calculate_Sum_Forces (V,$\vec{F}$)
  // Calculate the sum of forces of each nodes v of V.
- Move each v in V by C4 * $\vec{F}$

End For

End

Remark: In our case we have fixed M to one hundred because we treat graphs about three hundred nodes maximum. This constant M can be modified according to the number of nodes of "*Courses-Pre-Required-Graph*".

The calculation of the sum of the forces F for each nodes of Vis presented in this procedure:

Procedure Calculate_Sum_Forces (V,F)

For i=1 To n Do

F =0

For j =1 To n Do

if i ≠ j then

if (the node i is connected to node j) **then**

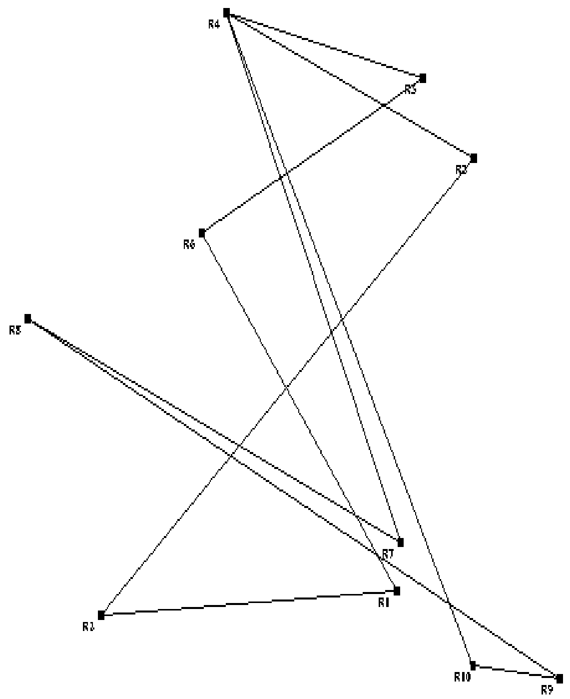
```
            Calculate the force Fij of the spring
            F= F + Fij
          else
           Calculate the repulsive Force Fr_ij
           F= F +Fr_ij
         End if
       End if
   End for
```

The time complexity of our algorithm "*Courses-Pre-Required-Visualization*" is O (n^2) where n is the number of nodes in the "*courses pre-required graph*".

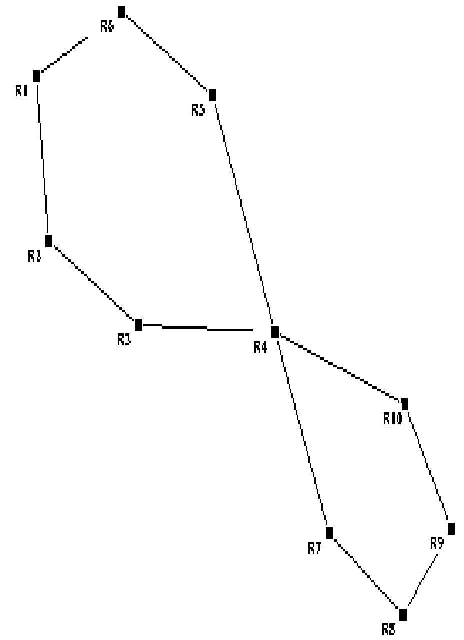
IV. Empirical Results

In this section we present the empirical results of our algorithm "Courses-Pre-Required-Visualization".

First example of visualization of the "*Courses-Pre-Required-graph*" with ten nodes.

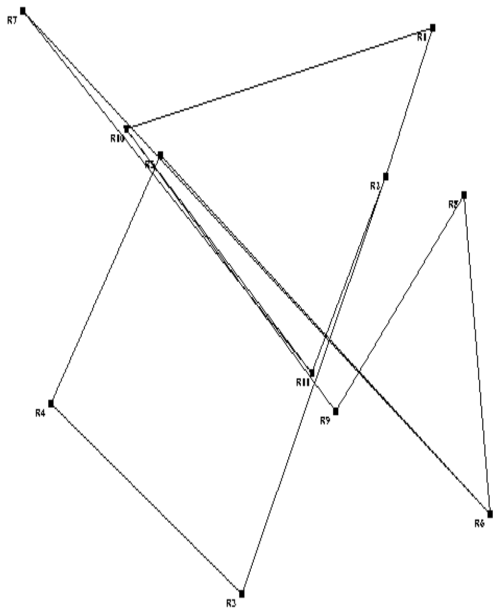


(a) A first random visualization of the "*pre-required graph*" (presence of crossing)

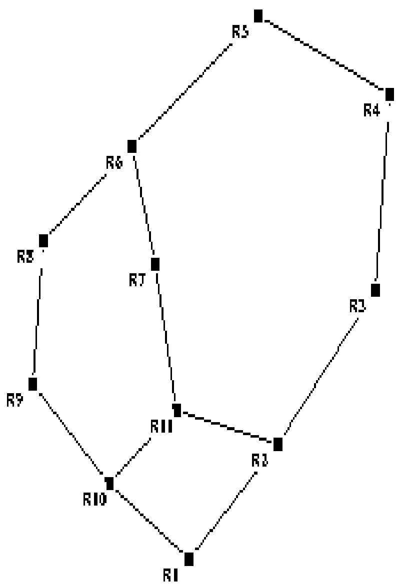


(b) A final Visualization of the "*Pre-Required graph*" using our Courses-Pre-Required Visualization (absence of crossing)

Second Example of visualization of the "*Courses-Pre-Required-Visualization*" with eleven nodes.

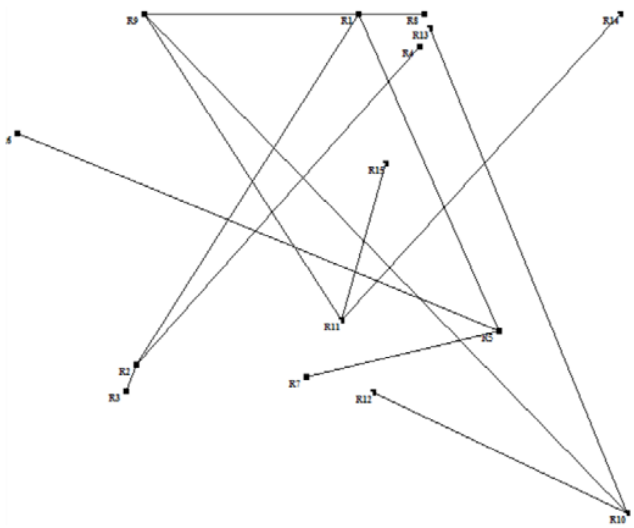


(c) A first random visualization of the "*Courses-Pre-Required graph*" (presence of crossing)

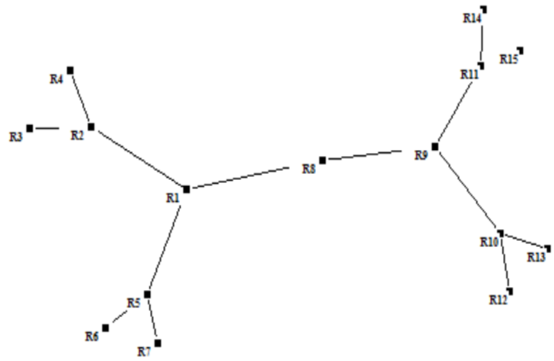


(d) A final Visualization of the "*Courses-Pre-Required graph*" using our Courses-Pre-Required Visualization (absence of crossing)

Third Example of visualization of the "*Courses-Pre-Required-Visualization*" with fifteen nodes.

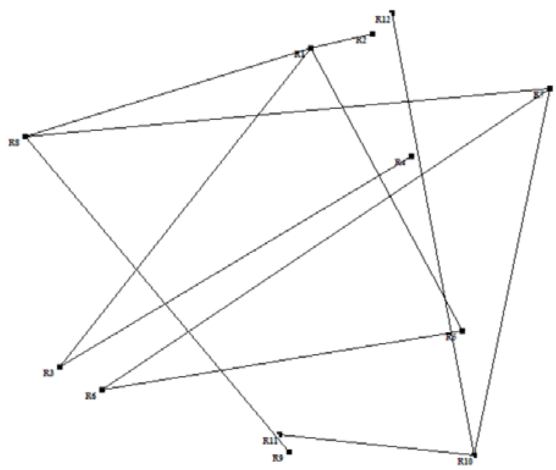


(e) A first random visualization of the "*Courses-Pre-Required graph*" (presence of crossing)

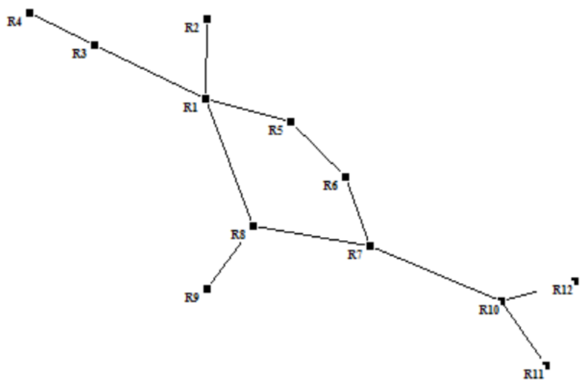


(f) A final Visualization of the "*Courses-Pre-Required graph*" using our Courses-Pre-Required Visualization (absence of crossing)

Fourth Example of visualization of the "*Courses-Pre-Required-Visualization*" with twelve nodes.

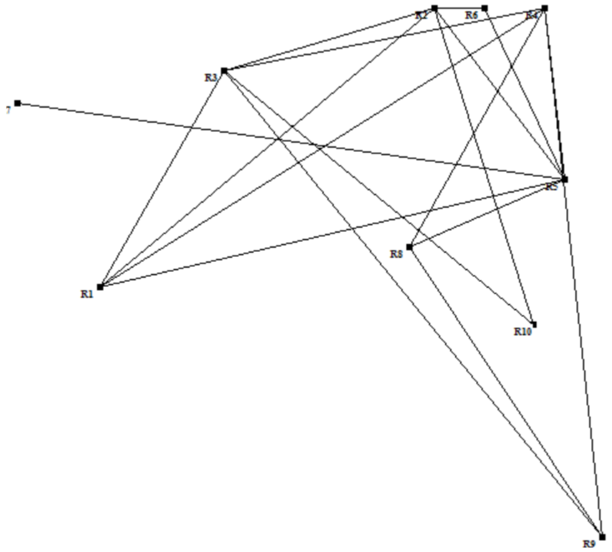


(g) A first random visualization of the "*Courses-Pre-Required graph*" (presence of crossing)

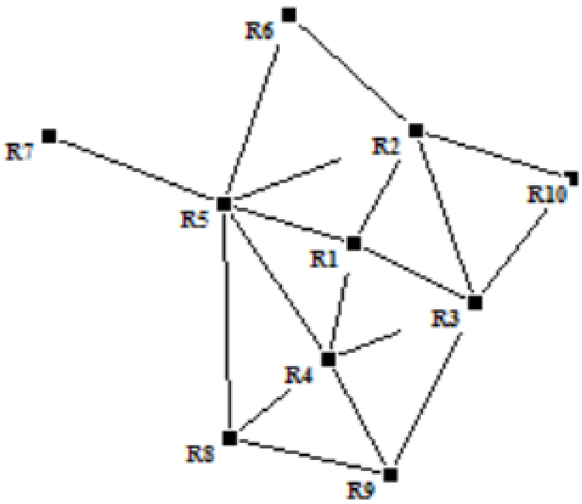


(h) A final Visualization of the "*Courses-Pre-Required graph*" using our Courses-Pre-Required Visualization (absence of crossing)

Fifth Example of visualization of the "*Courses-Pre-Required-Visualization*" with ten nodes.



(i) A first random visualization of the "*Courses-Pre-Required graph*" (presence of crossing)



(j) A final Visualization of the "*Courses-Pre-Required graph*" using our Courses-Pre-Required Visualization (absence of crossing)

Fig. 3 Snapshot of the system screen. Visualization of a "Courses-Pre-Required-Graph"

V. Conclusion

In this paper, we have presented an algorithm for visualization courses and their pre-required for students. In fact our contribution in this subject is to define a new concept of the "*courses pre-required graph*" and to present a new algorithm of visualization of this special graph. Our visualization is clear with a minimum crossing of edges and comprehensible by students. Our algorithm is based on "*Force-Directed Placement*" technique. The complexity of our algorithm is $O(n^2)$ where n is the number of nodes in the "*courses pre-required graph*". Our proposed visualization algorithm produces drawings that are clear and easy to understand by students. Besides our theoretical research, we have done an experimental work which consists of implementing the previously designed algorithm. For future research in this subject, we propose the following open problems:

- Visualize the path of all courses and their pre-required in all discipline,
- Define the graph that presents courses and their pre-required for all the disciplines.
- Adopt and implement the appropriate visualization algorithm.

References

[1] I. Ammari, A. Barouni, M. Elloumi, "Modélisation et développement d'une plateforme D'E-Learning pour l'Ecole Virtuelle de Tunis (EVT) en appliquent l'algorithme de ressort," in International Conference on New Technologies and Communication. Hassiba Ben Bouali University of Chlef, Algeria, December, 2012, pp. 15-22.

[2] M. Alam, T.Biedl, S. Felsner, M. Kaufmann, S. Kobourov, " Proportional Contact Representations of Planar Graphs," in the 19th International Symposium Graph Drawing 2011. Eindhoven University of Technology, Netherlands.

[3] A. Barouni, A. Jaoua, N. Zaguia, "Drawing Algorithms for Telecommunication networks," in 3rd CGC workshop on Comptuational Geometry. Brown University, Providence, RI, USA. October 1998.

[4] A. Barouni, A. Jaoua, N. Zaguia, "Planar Drawing Algorithms for survivable telecommunications networks," in proceedings of the Japan Conference on Discrete and Computational Geometry. Tokai University, Japan. December 1998.

[5] A. Barouni, A. Jaoua, N. Zaguia, "An Inside and Outside Algorithm for Drawing A Multi-Ring Architecture," in the International Journal Machine GRaphics & Vision Vol. 9, Nos. ½. 2000.

[6] A. Barouni, "Algorithme De Dessin Des Couvertures D'anneaux Utilisant L'emplacement Dirigé Par Des Forces," Phd Theses At The University Of Tunisia. December 2002.

[7] G.Di Battista, C. Batini, M. Talamo, R. Tamassia, "Computer Aided Layout Of Entity-Relationships Diagrams," The Journal Of Systems And Software, Vol. 4, 1984.

[8] G.Di Battista, C. Batini, E. Nardelli, R. Tamassia, "A Layout Algorithm For Data Flow Diagrams," Ieee Trans Software Eng., Vol. 12 1986.

[9] G. Di Battista, E. Pietrosanti, R. Tamassia, I.G. Tollis, "Automatic Layout Of Pert Diagrams With Xpert, " Proc. Ieee Workshop On Visual Languages (Vl'89).

[10] G.Di Battista, P. Eades, R. Tamassia, I.G. Tollis. "Algorithms For Automatic Graph Drawing: An Annotated Bibliography," Department Of Computer Science, Brown University, Technical Report, 1993.

[11] P.Eades, "A Heuristic For Graph Drawing, Congressus Numerantium," 42, 49-160 (1984).

[12] Fruchterman, T.M.J, Reingold, E.M.: "Graph Drawing By Force Directed Placement, Technical Report, Department Of Computer Science," University of Illinois at Urbana-Champaign, Urbana, IL, 1990.