edu.umbc.cs.maple.graphlayout_N

--INTRO--

edu.umbc.cs.maple.graphlayout_N is a package for producing multipurpose force directed graph layouts that can be used to build different types of graphs layouts that can serve as basis for creating and analyzing the following:
- Structure-based clusters;
- Random clusters;
- Attribute-based clusters;
- Combined clusters, etc;

--REQUIREMENTS--

The requirements to run the demo GraphLayoutMAPLE_N.java and the program with an extension example from the package are:
- Java 1.4 (available for free download at http://java.sun.com/j2se/1.4.2/download.html/;
- The Eclipse integrated development environment (available for free at http://ww.eclipse.org);
- Package prefuse, version prefuse-alpha-20050401.zip, - a user interface toolkit for building highly interactive visualizations of structured and unstructured data (available for free at http://prefuse.sourceforge.net/);
- input XML files in XGMML format (available in the CVS or in the prefuse;

See about creating new files in this format in the package edu.umbc.cs.maple.data_generation_MAPLE_N)

--BUILDING--

The package edu.umbc.cs.maple.graphlayout_N consists from:
- demo GraphLayoutMAPLE_N.java:
  a) To produce different types of the graph layout for any input XML files in XGMML format;
  b) To manipulate with graph's nodes;
  c) To build different types of clusters;
  d) To visualize the nodes and edges in a different way (shaping, coloring);
  e) To print nodes’, clusters’ and graph's information;

GraphLayoutMAPLE_N.java contains few commented lines that begin with ""//!!!..... "

When you uncomment these lines, it allows:
  a) To build different types of graph layouts;
  b) To serve different input files for visualization nodes in the created graph and clusters, especially for painting these nodes and different clusters;
  c) To insert the images into graph's nodes.

More explanations about commented lines see the demo GraphLayoutMAPLE_N.java description.

- printFileN.java - file that is used in the demo to print information about the hovered over node by mouse, and also to print the information about all nodes in the graph;

- timer.java - the functions of which can be used to delay for printing nodes’ information. With the delay we can print the node's information only once, but without these serving functions the information is printed every millisecond during the time when the node was hovered over;

- ColorDisplay.java - the program which is used only for choosing colors for better visualization. GraphLayoutMAPLE_N.java contains a collection of colors, but for additional colors we can use this program;

- Examples of demo GraphLayoutMAPLE_N.java with different types of uncommented lines are:

```
      GraphLayoutMAPLE_N_example_CRA_W_DMP.java;
      GraphLayoutMAPLE_N_example_GUASS.java;
      GraphLayoutMAPLE_N_example_OSCAR.java;
      GraphLayoutMAPLE_N_exampleFORCE_DIR.java;
```

The difference between the demo and these example_files are 3-4 lines and are inserted into package only as the demo's presentation for different purposes and different input files. The best way to use the demo for different purposes is to create a simple switch statement or add the switch panel into application, but the very short time of the internship program didn't allow creation of all codes for a more user-friendly demo. Demo GraphLayoutMAPLE_N.java has a structure that allows for simple modification of this program in the case of adding new functions for another clustering algorithm and clustering analyzing.

Demo GraphLayouMAPLE_N description

Created on Jul 22, 2005

Demo GraphLayouMAPLE_N allows to:
- read an input file by the XMLGraphReader, which is presented as a public static final Strings in the program and is uploaded into CVS; xml files which are presented in the prefuse/etc or any other files in the input XML files in XGMML format see http://www.cs.rpi.edu/~puninj/XGMML/ for a description of the XGMML format;

- print information about all nodes of the graph;
- build an action list that filters a tree in its entirety to perform a radial layout of visualized items
- build Renderer Factory
    - To create a new item registry for a visualization
    - To render an item as an image and a text string;
- build an action list which filters graph data into visual items, and imposes a tree structure on these items;
- build next types of the laoyut
    1) Radial Tree Layout;
    2) Force Directed Layout;
    3) Fruchterman Reingold Layout;

The descriptions and their combinations of above mentioned layouts are located at prefuse/extensions/edu.bekeley.guir.prefusex.layout;

The graph layout which was produced by this demo, GraphLayoutMAPLE_N_example_CRA_W_DMP.java, GraphLayoutMAPLE_N_example_GUASS.java and GraphLayoutMAPLE_N_example_OSCAR.java combines Radial Tree Layout and Force Directed Layout together. This combination allows constructing structure-based clusters (cluster creation based on the edge connection).
The clicking on any of the nodes moves this node to the center of the display, the forces that are between all nodes interrupt for this node, next displacement of this node after clicking on another node is solved by Radial Tree Layout. The perfect shape of the structure-based clusters is created after clicking on all nodes in the graph. The shape of the active node is increased and it is seen that the force on the node's edges disappears. This allows for the user to recognize which nodes were not clicked. After this the nodes are reshaped in the edge-connected clusters. Every next click on the node will move this node to the center of the display and collect the nodes that were joined from this node by outgoing edges to other nodes of the graph. The screenshot of this application you can find at my website: http://maple.cs.umbc.edu/~nlozova/ in MAPLE_square/project/application.

The node which is in the center changes its color, and then returns to its original color when another node is clicked;
A problem comes up with a large number of nodes; but process can be modified by adding a cycle function for clicking on all nodes in the graph

The graph layout that produced by GraphLayoutMAPLE_N_exampleFORCE, java doesn't employ clicking on the node. But all other functions including reshaping are the same for all programs. The best way to use this type of the demo is using the input files guass-centroi_n[i], which create the colored clusters by value from the input file (see datadescription.txt in the CVS).

- Functionality during hovering over or clicking on the node:
	- Changes node scaling;
	- Prints the node's information;
	- After a few milliseconds a new window appears near the node. It contains text information about node's label and it is very useful at a very large zoom out, when the label is unreadable;
	- Changes color of this node and returns to previous color after the mouse leaves this node;
	- Shows the outgoing edges from this node;

- Dragging:
	Only the nodes can be dragged; to drag a node – left click and drag the node to a different location;
	After dragging a node, it remains fixed at the new place (for Fruchterman Reingold Layout the node will be returned to the old place by edge forces);

- Zooming:
	To zoom in – right click the mouse and drag down; to zoom out – right click the mouse and drag up;
	The best way to run any layout is to zoom out very far, which allows to extend the forces and distances between nodes – it makes it more visible;

- Coloring

The color function was created for a better visualization of every type of input files which are presented in the demo. The name of the color function is mapped to the name of the input file. The presented examples in the package show the way how to use different types of the color function.

Some examples of the graph layout coloring with different input files are represented on the screenshot of these applications at http://maple.cs.umbc.edu/~nlozova/ in MAPLE_square/project/application

- Panning:

To pan the whole application in all ways, click and drag the background

GraphLayoutMAPLE_N_example_GUASS.java, which loads guass_centroid_N[i].xml file (see dataset description) allows to separate attribute-based clusters from started force-directed layout by simple dragging of nodes with a label starting from letter ”c..” (cluster’s centroid) into a different part of the display.

Demo GraphLayouMAPLE_N doesn't contain the algorithms for clustering analysis, but they can be easy inserted into the demo, or the demo can be performed for other clustering types

```
/**
   * @author nlozova
   * LSU student
   * nlozov1@lsu.edu
   * Summer DMP Internship Program 2005 at UMBC
   * Revised: August 11, 2005
*/
```