Skinable graph drawing

Mountaz Hascoët

Skinnable Graph Drawing

Mountaz Hascoët

LIRMM, CNRS UMR 5506, Univ. Montpellier II, 161, rue Ada 34392 Montpellier Cedex, France
mountaz@lirmm.fr

1 Introduction

Skinning is a recent concept emerging from long lasting efforts to separate look-and-feel from core functional aspects in software [4] or web design. In this paper, we propose skinnable graph drawing as a process that takes a general graph $G(V, E)$ and computes a drawing of G from a skin of G and a signature of G. Skinnable graph drawing objective is two-fold: (1) reduce the time and effort spent drawing and customizing a drawing for general graphs that are drawn and customized several times and (2) maintain invariants in the drawing for the preservation of both personal mental maps [5] and group mental maps.

2 Skins

Let $G(V, E)$ be a graph. A drawing of G is a geometric representation of G such that each element of G (vertex, or edge) is drawn as a geometric shape (e.g point, circle, polygon, Jordan curve, etc)[2, 3]. We call a skin for G a set of geometric shapes with the following two properties: (1) the skin can be customized, reused, shared, copied, etc, independently of the graph description and (2) a skin of G and a signature of G are sufficient to create a drawing of G. Collections of skins can be created once and shared amongst experts using the same graphs, hence preserving not only an individual mental map but also a sort of a collective mental map for most important graphs in a community. Furthermore, different perspectives on the same graphs can further be embodied by different skins.

3 Graph Signature

Let $G(V, E)$ be a graph, we introduce a context coefficient denoted by $c(u)$ and defined in equation (1). A context coefficient can be considered as a node invariant [1] at the crossroads of clustering coefficient and twopath [1]. Based on context coefficient, a stamping for G is defined as a function f (equation (2)). Finally, the signature of G is an ordered list of tuples $(f(u), n_k)$. The order of the list is based on f(u) values.

$$c(u) = \frac{n(u)}{\tau(u)(\tau(u) - 1)/2} \tag{1}$$

where $n(u) = card\,\{(a,b) \in E \mid dist(a,u) <= 2\ and\ dist(b,u) <= 2\}$ and $\tau(u)$ is the number of vertices reachable along a path of length 2 starting from u - e.g. twopath.

$$\forall u \in V, f(u) = (c(u), d(u)) \tag{2}$$

where $d(u)$ is the degree of node u and $c(u)$ its context coefficient.

$$S = \{(f(u), n_k)\}_{u \in V, k \in N} \tag{3}$$

where $n_k = card(\{u \in V | f(u) = (c(u), d(u))\}$

4 Skinnable graph drawing

A general algorithm for skinnable graph drawing can be derived from the concepts previously defined. Skin elements are identified by composite invariants composing the signature so that the mapping of graph elements to their corresponding graphical elements in the skin is straightforward. In the case of graphs where several nodes might have the same invariants, the choice of which node correspond to which graphical element is random and therefore subject to errors. Powerful interaction techniques can be used to leave necessary adjustments to experts. Guaranteing a unique choice in the general case would lead us far beyond the scope of this paper considering the isomorphism problem [1].

5 Conclusion

Skinnable graph drawing builds upon previous work on software engineering, graph drawing and graph canonical labeling. Future work includes the integration of skin computation from various sources. For example, skins can be produced as direct output of most graph drawing programs or skins can be created from stored drawings of commonly used graphs or skins can even be produced from artistic drawings. Future work also includes the design of interaction techniques that facilitate interactive skin adjustments when necessary.

References

1. Fortin, S.: The graph isomorphism problem. Tech. rep., University of Alberta (1996)
2. Giacomo, E.D., Didimo, W., Liotta, G., Palladino, P.: Visual analysis of one-to-many matched graphs. J. Graph Algorithms Appl. 14(1), 97–119 (2010)
3. Jünger, M., Mutzel, P.: Graph drawing software. Mathematics and visualization, Springer (2004)
4. Krasner, G.E., Pope, S.T.: A description of the model-view-controller user interface paradigm in the smalltalk-80 system. Journal of Object Oriented Programming 1(3), 26–49 (1988), `http://citeseer.nj.nec.com/krasner88description.html`
5. Misue, K., Eades, P., Lai, W., Sugiyama, K.: Layout adjustment and the mental map. Journal of Visual Languages And Computing 6(2), 183–210 (1995)