How Important Is the "Mental Map"? – An Empirical Investigation of a Dynamic Graph Layout Algorithm

Helen C. Purchase[1], Eve Hoggan[1], and Carsten Görg[2,⋆]

[1] Department of Computing Science, University of Glasgow, Glasgow, UK
[2] College of Computing, Georgia Institute of Technology, Atlanta, USA
{hcp,eve}@dcs.gla.ac.uk, goerg@cc.gatech.edu

Abstract. While some research has been performed on the human understanding of static graph layout algorithms, dynamic graph layout algorithms have only recently been developed sufficiently to enable similar investigations. This paper presents the first empirical analysis of a dynamic graph layout algorithm, focusing on the assumption that maintaining the "mental map" between time-slices assists with the comprehension of the evolving graph. The results confirm this assumption with respect to some categories of tasks.

1 Introduction

Research on algorithms for the effective layout of graphs has been active for many years - these algorithms have typically been valued for their computational efficiency and the extent to which they conform to pre-defined layout principles. Only recently any empirical work had been conducted to determine the effect of conformance to these principles on user understanding [7]. While static graphs are still applicable and very useful in a variety of situations, recent developments in graph layout research have concentrated on the layout of dynamic graphs, representing changing relational information over time. Examples of applications of such dynamic processes include software engineering visualizations where graphs depict the execution time behavior, changing Internet usage and changes in social network structures. Research on the effects of dynamic layout principles on user understanding is therefore timely.

1.1 Dynamic Graph Layout Systems

Empirical investigation of a dynamic graph layout system requires that the system have changeable parameters, so that comparative tests can be performed. Several dynamic graph layout systems are domain specific. For example, Brandes and Corman [1] present a method for visualizing network evolution in which

⋆ The author was supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD).

M. Kaufmann and D. Wagner (Eds.): GD 2006, LNCS 4372, pp. 184–195, 2007.

each modification is shown in a separate layer of 3D representation with edges common to two layers represented as columns connecting the layers. Gevol is a system that visualizes the evolution of software using a modification of a force-directed algorithm [3]. Gevol extracts information about a Java program stored within a CVS version control system and displays it using a temporal graph visualization method. These systems have all been built for a specific purpose or specific type of information visualization. Thus there is little room for the manipulation of layout parameters and the use of different types of algorithms or data.

GraphAEL is a more generic system for graph animation of evolving layouts which has been designed to provide the necessary structure and flexibility for force-directed graph drawing research [5]. The system contains several novel algorithms and visualization techniques, such as force-directed methods in hyperbolic and spherical spaces. However, the research prototype system made available did not have an obvious way of manipulating parameters.

We therefore chose GraphAnimation [4] as the first dynamic graph layout system for our experiments. GraphAnimation provides two layouts: one based on the spring algorithm, and one based on a hierarchical algorithm. It provides easy manipulation of important parameters.

1.2 The User's "Mental Map"

Dynamic graph drawing involves laying out graphs which evolve over time by the addition or deletion of edges and nodes at the end of each time period. In creating systems that produce dynamic graph animation, algorithm designers have needed to take into account an additional aesthetic criterion known as "preserving the mental map" [2]. The term mental map refers to the structural cognitive information a user creates internally by observing the layout of the graph [4]. This internal cognitive structure represents the user's underlying understanding of the information. It is important that this remains consistent throughout the dynamic graph animation, otherwise confusion may result. A good preservation of the mental map can help people to understand the application, but a display where it is difficult for the user to maintain a consistent mental map can be misleading.

In developing a dynamic graph layout algorithm, it is of course useful to take advantage of existing static graph layout algorithms. Given that each time-slice could be considered as a static graph, a static graph layout algorithm can be applied to the graph after each update. Using animation between time-slices to show how nodes and edges are moved to the new positions may assist in preserving the mental map over time.

There are therefore two important criteria to consider: the readability of the individual static layouts and the mental map preservation in the sequence of drawings. Thus, some interaction is required between the static layout applied at each time-slice, and the movement of nodes and edges between each time-slice. The readability of the individual graph drawings produced by a static layout algorithm depends on aesthetic criteria such as minimal number of bends, uniform edge lengths, and minimal number of crossings. Preservation of the mental map can be achieved by ensuring that nodes that appear in consecutive graphs in the sequence remain in the same positions, so that they can easily be identified as the same nodes over time.

These two criteria are often contradictory. If each graph is laid out individually, without regard to other graphs in the sequence, its readability may be optimized at the expense of mental map preservation. This may result in an animation where nodes change position radically from time-slice to time-slice. On the other hand, if the node positions are fixed in all graphs, mental map preservation is optimized but the individual layouts may not conform to static graph drawing aesthetics. This may result in an animation where the time-slices themselves are difficult to understand because of, for example, a high number of edge crossings, or several awkward edge bends.

The experiment reported in this paper addresses this contradiction in hierarchical layout of graphs, attempting to determine whether, from a user-understanding point of view, it is preferable to maintain the mental model, conform to static layout aesthetics at each time-slice, or make a compromise between the two.

2 Experimental Methodology

The experimental methodology used was based on former static graph layout experiments [7], using an online system to present the graphs, asking the participants to enter their answers to questions on the graphs, and collecting error and response time data. As before, tutorial and worked example material was given at the beginning to familiarize the participants with the experimental tasks, with a ranking and qualitative questionnaire at the end. A within-subjects methodology was used to reduce any subject variability, with the inclusion of practice tasks and randomization controlling for the learning effect. User-controlled rest breaks were included throughout the duration of the experiment, to address any problems of fatigue.

For the empirical analysis of dynamic graphs, additional experimental design decisions needed to be made with regard to the timing of the presentation of the graphs and questions. Through pre-pilot and pilots tests, we determined appropriate animation speed, time per time-slice, the number of times to show the complete animation, pause time before and after the animation, and an appropriate range for the feasible number of time-slices. We also determined when, and for how long, the question should be displayed. These pilots tests were essential, there being no other former experiments of this kind to inform these necessary details for our experimental design.

The experimental system used, DynaGUESS was implemented as a generic system for information visualization experiments. It facilitates easy preparation and customization of online experiments, enabling the experimenter to set parameters regarding timing, randomization, and rest breaks. The dependent variables are the accuracy of the question answers, and the response time.

2.1 The GraphAnimation Hierarchical Layout Algorithm

The algorithm evaluated was the hierarchical dynamic layout provided by GraphAnimation [4]. GraphAnimation provides a generic framework for a dynamic layout process. It offers the possibility to trade local layout quality for dynamic stability (preserving the user's mental map) by adjusting a parameter delta that limits the changes between two layouts of succeeding graphs. A large delta offers more freedom to support the optimization of the local layout quality whereas a small delta fortifies the preservation of the mental map.

The hierarchical dynamic layout process in GraphAnimation has many facets. The static part of the layout algorithm used at each time-slice maintains consistent vertical flow of directed edges; it attempts to produce a layout such that all edges point downwards. This is done for each graph of the sequence. The dynamic part of the layout algorithm tries first to keep nodes on the same hierarchical level as in the previous time-slice, and secondly tries to keep nodes in the same horizontal order. There is therefore a possible tension that could lead to a contradiction: the static part of the hierarchical layout algorithm could try to change the vertical level of a node to fulfill the flow constraint, while the dynamic part of the algorithm tries to keep its level and horizontal order. It is the choice of parameters that determines the result if there is a conflict.

GraphAnimation provides two delta parameters by which the extent to which the mental model is maintained between time-slices can be controlled. The two parameters affect the main phases of the hierarchical layout process – the layer assignment and the layer sorting:

- Delta on ranks: maximal number of changes in the layer assignment
- Delta on orders: maximal number of changes in the order of the nodes within one layer

These parameters were used to produce dynamic graph layout animations under three conditions, these were our independent variables. The values of the parameters were chosen after extensive visual analysis.

1. High Delta Condition (both deltas=40): priority is given to layout quality at each time-slice rather than preserving the mental map. This value was defined in pilot tests as the minimum value for all nodes to move freely.
2. Medium Delta Condition (both deltas=20): equal priority is given to layout quality at each time-slice and preserving the mental map.
3. Low Delta Condition (both deltas=0): priority is given to preserving the mental map rather than the layout quality at each time-slice. This is the best this algorithm can do to maintain the mental model.

The details of the hierarchical dynamic layout algorithm used in GraphAnimation are presented in [6].

2.2 Experimental Tasks

Three different evolving graphs were created, each with between 14 and 20 nodes, between 15 and 30 edges and 4 changes per time-slice, where a change is an insertion or deletion of a node or an edge. We aimed to keep the size and changes of these graphs as similar as possible, while keeping them distinctive. The graphs represented the changing structure of an imaginary web site over six time periods. Each graph animation was created using GAML, an XML-format based on GraphML which is required by GraphAnimation. Overall, each task had a time limit of 30 seconds. Each time-slice was displayed for 3 seconds with the animation between time-slices set at a speed of 1 second. There was a pause of 4 seconds at the beginning of each task in order to allow the user enough time to read the question before the animation began. A warning beep was sounded 3 seconds before the end of each task to ensure that participants entered their answer before running out of time.

For each of these graphs, three versions were created, one for each of our three experimental conditions. Thus, we had nine graph animations in total. These are here referred to by their graph number (1,2,3), followed by their condition (L, M, H). For example, the high delta version of the third graph is referred to as G3H. The appendix shows the diagrams of all the individual layouts of the three versions of graph 2. The animations of all nine evolving graphs used for the experiment are available at `http://www.cc.gatech.edu/~goerg/GD06`.

Each of these nine animations was presented four times, once for each of four questions, resulting in 36 total tasks. The order of the presentation of tasks was random. Before beginning the actual 36 tasks, the participants were presented with a practice set of 8 tasks.
The questions were:

1. How many new links were added to the site over the years?
2. Which page has been changed the most over the years? (meaning the node which has had the most changes in its incoming and outgoing edges).
3. In which year did the site reduce in size by a quarter? (meaning the year in which the number of nodes reduced by 25%)
4. In which year did [page name] become accessible through only one other page?

The first three questions therefore covered the addition and removal of both nodes and edges, while the fourth one considered the structure of the graph. Each question was multiple choice with the participants indicating one of four possible answers their answer with a check box. The incorrect options that were listed with the correct option were chosen randomly. The worked example and tutorial at the start of the experiment ensured that the participants knew the meaning of the questions.

2.3 Experimental Process

20 student participants were recruited; they were typically of a computing science background, although there were also some zoologists, musicians and law students. The experiments were held in 5 sessions over a period of 10 days. Each experiment, including time spent at the beginning on the tutorial and the worked example, and on the questionnaire at the end, took approximately 45

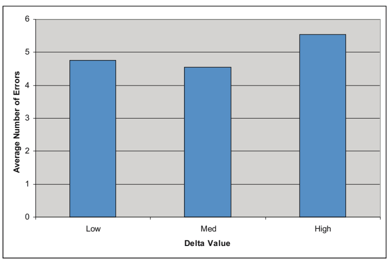
minutes. No problems were experienced during the experiments, and all participants appeared to engage in the task seriously. Participants were paid 5 for their participation.

3 Results and Analysis

3.1 Analysis by Delta Condition

Our hypothesis was that the extent to which the mental model was maintained between time-slices would affect performance. Intuitively, we felt that a low delta value (which results in an animation that attempts to maintain the mental model) would produce a better performance than a high delta value (which attempts to produce a 'good layout' at each time-slice).

Performance Data. The average number of errors and the average response time for the three delta conditions are shown in Figure 1.



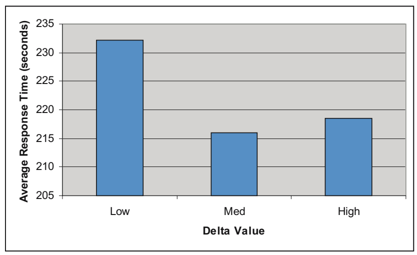


Fig. 1. Average Errors and Response Time according to delta. Low delta means few visual changes between time-slices, so a high conformance to maintaining the mental model.

To test the hypothesis, first the significance of the effects of each delta condition were investigated [1]. The statistical analysis used here is a standard two-tailed ANOVA analysis, based on the critical values of the F distribution, with $\alpha = 0.05$. In all cases, conservative readings of the critical values of the F distribution were used.

Errors. There are no significant differences between delta conditions in the error data, as $F = 1.366 < (F(2, 57) = 3.23)$.

Response Time. There are no significant differences between delta conditions in the response time data, as $F = 2.489 < (F(2, 57) = 3.23)$.

[1] Identifying a statistical significance between results collected from different conditions indicates that the difference between the results can be attributed to the differing nature of these conditions, rather than being due to mere chance. In these experiments, we test whether the probability of the difference being due to chance is less than 0.05.

3.2 Further Analysis

At first glance these results appear to disprove our hypothesis. However, further analysis was performed to identify any possible significant differences between delta conditions with respect to a particular question or graph. The analysis above aggregates the data from all four questions and all three graphs: it may be that delta conditions only affect particular questions or graphs.

We anticipated that there would be a difference in the difficulty of the questions, and that the delta conditions may have significant effects when considered independently with respect to the data from each of the different questions. However, having made an effort to keep the three evolving graphs comparable (similar size, similar number of changes per time-slice), we did not anticipate that there would be any difference in difficulty between the graphs. The further analysis took the form of first applying the ANOVA test to the data from the four questions, to see if there were differences in performance between them. If there were significant differences between the performances on the questions, we then analyzed each question separately according to delta condition, applying the Bonferroni correction as appropriate. A similar process was followed for the three graphs.

Analysis with respect to Question. There are no significant differences in the number of errors between questions ($F = 0.6540 < F(3, 76) = 2.76$).

However, there are significant differences in the response time data between questions ($F = 35.52 > F(3, 76)$). Tukey's pairwise analysis showed that the average response time for Q1 was significantly greater than the response times in Q2, Q3, and Q4. It also showed that the response time for Q2 was significantly less than the response time in Q4, and the response time for Q2 approaches significance when compared with Q3. There were no other pairwise differences (see Figure 2).

As expected, the questions were of different difficulty. Separate analyzes were performed on the effect of delta condition on the response times for each of the four different questions.

There were no significant differences in the response time data between delta conditions in Q1 or in Q2.

For Question 3, ("In which year did the site reduce in size by a quarter?"), there are significant differences in the response time data between delta conditions ($F = 5.832 > F(2, 57)$). Tukey's pairwise analysis showed that the average response time for low delta value was significantly greater than the response times in both medium and high delta value conditions. There were no other pairwise differences (see Figure 3(a)).

For Question 4 ("In which year did [page name] become accessible through only one other page?"), there were also significant differences in the response time between conditions ($F = 6.903 > F(2, 57)$). Tukey's pairwise analysis showed that the average response time for low delta values was significantly greater than the response times in medium and high delta values. There were no other significant pairwise differences (see Figure 3(b)).

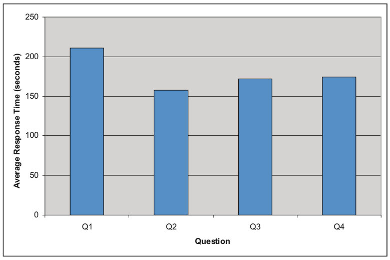
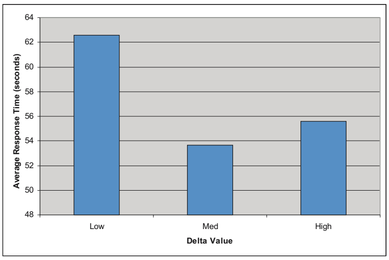
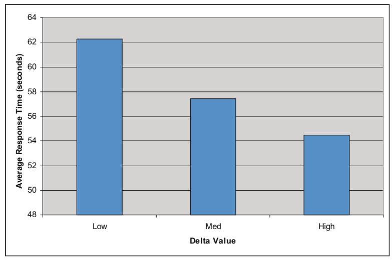


Fig. 2. Average Response Time according to Question



(a) Question 3



(b) Question 4

Fig. 3. Analysis by delta condition

Analysis according to Graph. Our assumption was that the three evolving graphs themselves were of comparable difficulty (in terms of size and number of changes per time-slice), and therefore it was appropriate to aggregate the data from all three of them when testing our initial hypothesis. Our next step was to test this assumption.

There were no significant differences in the number of errors between each graph. However, for response time, there were significant differences between the graphs ($F = 15.98 > F(2, 57)$). Tukey's pairwise analysis showed that the average response times for graphs 2 and 3 were significantly greater than the response times in graph 1. There were no other pairwise differences (see Figure 4).

This indicated that the graphs were unexpectedly of different difficulty. Separate analyzes were therefore performed on the effect of the delta condition on response times for each of the three different graphs. There were no significant differences in the response time data between delta conditions in graph 1, but there were differences between delta conditions in graphs 2 and 3.

On examining the three graphs, we could see no discernible differences between them that would explain the better performance on graph 1, apart from the fact that graph 1 had more levels and was narrower than the other two graphs. The fact that there was no variation in data for graph 1 indicates that it was so easy that a 'floor' effect resulted. We therefore removed graph 1 from our analysis, and repeated the delta condition tests, for each of the four questions.

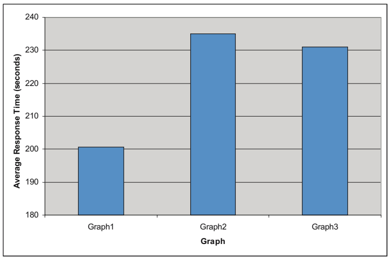


Fig. 4. Average Response Time according to graph

The results were more encouraging. There was no significance for question 1, probably due to a 'ceiling effect', as the question was particularly difficult[2]. Significance was found in all three other questions:

- For question 2, low delta produced better performance than high delta for errors ($F = 3.23 > F(2, 57)$)(see Figure 5).
- For question 3, low and medium delta produced better performance than high delta for errors ($F = 3.23 > F(2, 57)$); however, as there was also significance in time data (low and medium delta produced worse performance than high delta ($F = 3.23 > F(2, 57)$), this indicates a correlation between time and error data, so little weight can be attached to these results(see Figure 6).
- For question 4, low delta produced better performance than high delta for errors ($F = 3.23 > F(2, 57)$)(see Figure 7).

3.3 Analysis Summary

The significant results with respect to our independent variable, the mental model (delta) condition, were:

- Question 2 errors ("which page has been changed the most over the years"), over graphs 2 and 3: high mental model was important.
- Question 3 response time ("in which year did the site reduce in size by a quarter"), over all graphs: high mental model was not important.
- Question 3 errors ("in which year did the site reduce in size by a quarter"), over graphs 2 and 3: high mental model was not important.
- Question 3 response time ("in which year did the site reduce in size by a quarter"), over graphs 2 and 3: high mental model was important.
- Question 4 response time ("in which year did [page name] become accessible through only one other page?"), over all graphs: high mental model was not important.

[2] A floor/ceiling effect is when the task is so easy/difficult that the manipulation of experimental condition has no effect on the data variation.

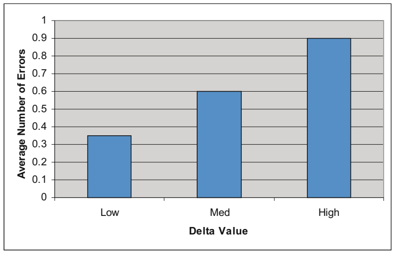


Fig. 5. Average Errors according to Q2, for graphs 2 and 3

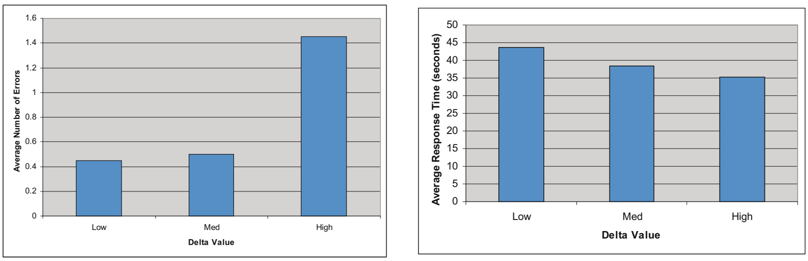


Fig. 6. Average Errors and Response Times according to Q3, for graphs 2 and 3

– Question 4 errors ("in which year did [page name] become accessible through only one other page?"), over graphs 2 and 3: high mental model was important.

So, in summary, we cannot make any claims about the usefulness of the mental model for question 1 (as there were no significant results), nor for question 3 (as the significant results are contradictory, indicating a correlation between the dependent variables). However, the data implies that when we remove graph 1 from our analysis (as it was found to be too easy), the mental model did assist with understanding for questions 2 and 4.

– Question 2: "Which page has been changed the most over the years?" In this case, the mental model would have kept the nodes in similar positions, so that the participants could focus on the differing density of edges in the diagram.
– Question 4: "In which year did [page name] become accessible through only one other page?" The mental model would have helped with this question, as the participants would have needed to follow a single node through all time-slices.

Although we cannot make any concrete claims about questions 1 and 3, on examining these questions, we suggest the following:

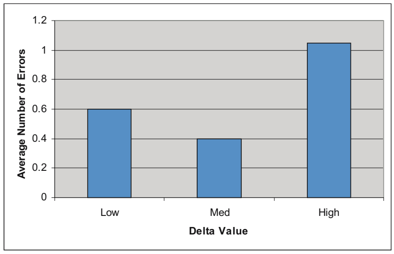


Fig. 7. Average Errors according to Q4, for graphs 2 and 3

- Correctly answering question 1 ("How many new links were added to the site over the years?") does not rely on node movement or identification, so the delta value is irrelevant. This was an extremely difficult question, even when multiple choice options are given.
- Similarly, correctly answering question 3 ("In which year did the site reduce in size by a quarter?") only focuses on the number of nodes, not their identification. Thus, the delta value is irrelevant.

We therefore suggest that maintaining the mental map is important for the comprehension of an evolving graph for tasks that require that nodes be identifiable by name, but that it is less important for tasks that focus on edges rather than nodes, or which do not require that nodes be nominally differentiated from each other.

4 Conclusions

These conclusions are, of course, limited by our choice of graph parameters: the dynamic algorithm used, the size and structure of the graph, the number of changes per time-slice, and animation speed, and any conclusions can be made only within the context of these choices. Choices that resulted in tasks being too easy (G1) or too difficult (Q1) further constrained the conclusions.

Devising an experimental methodology for dynamic graph layout proved more difficult than for static graph drawings. In particular, we could not rely on experimental tasks based on common graph theoretic questions (for example, "What is the shortest path between two nodes") as such questions to do take into account the temporal, changing nature of the information. It was also very difficult, as shown, to create three dynamic graphs of similar complexity, even when the quantitative descriptors of these graphs were comparable. Other practical experimental issues needed to be addressed carefully, for example, when to display the question, how many times to show the animation, how to present a worked example with explanation of correct answers, what pauses are necessary at the start and end. It became clear that the temporal aspect of a dynamic algorithm makes for a more complex experimental process.

These results are the first empirical evidence that the mental map factor is important in the layout of dynamic graphs and are important despite the restricted size and scope of this experiment. However, the contribution of this paper to graph drawing research is more than merely affirming assumptions made about the importance of maintaining the mental map in the context of two types of tasks. As the first experiments in this area, it represents a significant step forward in human empirical research on the efficacy of graph layout algorithms. There is, of course, substantial future work to be performed in this area - there are other dynamic algorithms to investigate (e.g. force directed, orthogonal), other visualization features that can change over time (for example, node sizes, color), other domains, different sizes and structures of graphs, and alternative tasks. This initial research opens up this wide, rich area for investigation.

The animations of the graphs used for this experiment are available at the following webpage `http://www.cc.gatech.edu/~goerg/GD06`.

References

1. Ulrik Brandes and Steven R. Corman. Visual unrolling of network evolution and the analysis of dynamic discourse. In *INFOVIS '02: Proceedings of the IEEE Symposium on Information Visualization (InfoVis'02)*, page 145, Washington, DC, USA, 2002. IEEE Computer Society.
2. Michael K. Coleman and D. Stott Parker. Aesthetics-based graph layout for human consumption. *Softw. Pract. Exper.*, 26(12):1415–1438, 1996.
3. Christian Collberg, Stephen Kobourov, Jasvir Nagra, Jacob Pitts, and Kevin Wampler. A system for graph-based visualization of the evolution of software. In *SoftVis '03: Proceedings of the 2003 ACM symposium on Software visualization*, pages 77–ff, New York, NY, USA, 2003. ACM Press.
4. Stephan Diehl and Carsten Görg. Graphs, they are changing – dynamic graph drawing for a sequence of graphs. In *Proceedings of Graph Drawing 2002*, pages 23–30, London, UK, 2002. Springer-Verlag.
5. Cesim Erten, Philip J. Harding, Stephen G. Kobourov, Kevin Wampler, and Gary V. Yee. Graphael: Graph animations with evolving layouts. In Giuseppe Liotta, editor, *Graph Drawing*, volume 2912 of *Lecture Notes in Computer Science*, pages 98–110. Springer, 2003.
6. Carsten Görg, Peter Birke, Mathias Pohl, and Stephan Diehl. Dynamic graph drawing of sequences of orthogonal and hierarchical graphs. In János Pach, editor, *Graph Drawing, New York, 2004*, pages pp. 228–238. Springer, 2004.
7. Helen C. Purchase. Effective information visualisation: a study of graph drawing aesthetics and algorithms. *Interacting with Computers*, 13(2):147–162, 2000.