Dynamic Visualization of Graphs with Extended Labels

Pak Chung Wong[1, ‡], Patrick Mackey[1], Ken Perrine[1], James Eagan[2], Harlan Foote[1], and Jim Thomas[1]

[1]Pacific Northwest National Laboratory

[2]Georgia Institute of Technology

Abstract

The paper describes a novel technique to visualize graphs with extended node and link labels. The lengths of these labels range from a short phrase to a full sentence to an entire paragraph and beyond. Our solution is different from all the existing approaches that almost always rely on intensive computational effort to optimize the label placement problem. Instead, we share the visualization resources with the graph and present the label information in static, interactive, and dynamic modes without the requirement for tackling the intractability issues. This allows us to reallocate the computational resources for dynamic presentation of real-time information. The paper includes a user study to evaluate the effectiveness and efficiency of the visualization technique.

CR Categories: I.6.9 [Visualization] – Information Visualization, Visualization Systems and Software, Visualization Techniques and Methodologies

Keywords: Graph Label Placement, Dynamic Animation, Graph Visualization, Information Visualization

1 Introduction

Graph drawing [4][6][18] has long been recognized as a profound and challenging research topic. Scientists soon found out that it was equally difficult to draw graph "labels" [3][8][9][13][17][20][22] alongside the graphs. To begin with, competing with the graph for drawing space to accommodate all the labels is a challenge. Determining optimal locations for the link labels has turned out to be algorithmically intractable [10]. So when we encountered the task of visualizing graphs with "extended" labels, we decided to seek an alternative solution.

Our visualization problem is unique not because of the structure or content of the underlying graphs but because of the extended size (or length) of their graph labels. The length of these labels can range from a short phrase to a full sentence to an entire paragraph of text per graph node or link. While graphs like these are not necessarily the norm today, we see convincing evidence [15][19] that the problem is looming large on the horizon.

Our problem can be described as a graph visualization task that balances the appearance of both the graph and its labels. Although this paper focuses mainly on visualizing link labels, we do provide solutions for short- to medium-length node labels.

The nature of the graph structure has itself created chaos in many graph visualizations. So instead of solving two problems (i.e., both graph and label drawing) at the same time, we chose to leave the already well-explored graph drawing area [4][18] alone and build a label visualization solution on top of an existing graph. In other words, we set out to intelligently insert label information on an already nicely drawn graph. Thus the research challenge became a task to invent a new drawing space from a potentially messy graph visualization.

Our solution to the limited-space problem is to recycle existing resources already allocated for the graph and share the graph drawing space with the labels. For example, both the link arrow and its label in Figure 1a are replaced with the green label in Figure 1b. We argue that our visualization approach is at least *as effective as any graph drawing* (i.e., without labels) techniques because we merely overlay the graph link in this example with additional information. Our visualization approach is also *guaranteed to be more space-friendly* because a label alone always occupies less space than a link and a label together, as shown in Figure 1.

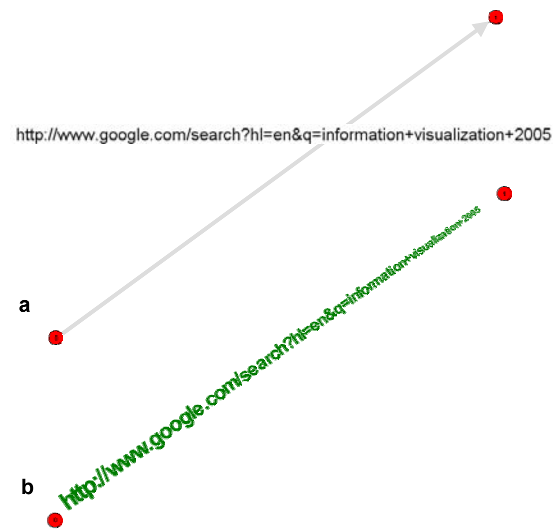

Figure 1: Visualizing a link label using a) the traditional approach and b) the new approach.

This paper presents a system prototype, known as GreenArrow, which implements our new graph label visualization technique and a few other supporting tools to address requirements of different applications. Part of our research effort is to conduct a user study to evaluate the efficiency and effectiveness of GreenArrow. We hypothesize our design expectations and compare them with the outcomes of the study. The results are discussed in the second half of the paper.

2 Related Work

The label placement problem is about properly tagging a text label next to a graphical entity within a picture or visualization. It is a particularly important task in the field of cartography where map label placement is still not a fully automated process today. A major difference between placing map labels and graph labels is that the coordinates of the map objects are almost always pre-defined, whereas the locations of the graph objects are dynamic in nature. We will discuss the literature of both applications and then highlight the major difference between our work and others'.

‡pak.wong@pnl.gov

IEEE Symposium on Information Visualization 2005
October 23-25, Minneapolis, MN, USA
0-7803-9464-X/05/$20.00 ©2005 IEEE.

2.1 Graph and Map Label Placement

Christensen et al. [3] present an earlier empirical study of algorithms on label placement methods. The paper defines the problem, highlights solutions, and compares their results. Its reference section represents one of the most comprehensive surveys of the cartographic text placement work before the early '90s.

A variety of reported results address different types of graph labeling problems. The most common approach has been to use an optimization approximation algorithm like simulated annealing to optimize a cost function of the label assignment. How the costs are determined and the specific method of optimization used depends on the solution and the problem at hand. Some of these solutions focus on the node (or point) label problems [3][17]; others emphasize the link (or line) label problems [9][20]; and others [8][13] provide solutions to both node and link labels.

Wolff's map-labeling bibliography [21] contains a wealth of information on the topic of map labeling. In addition to research resources supported by the owner, the website also includes links to a dozen URLs that discuss different types of label-placement problems and applications. Other recent map-label placement publications include Iturriaga [7] and Kameda and Imai [12]. Both provide customized algorithms for label presentation that also can be applied to graph label visualization.

In any case, the label assignment is very computationally intensive and can be even more difficult to compute depending on the type of graph drawing used, e.g., if edges contain bends. In fact, Kakoulis and Tollis [10] have proved that the link (or edge) label placement problem is at least NP-hard [2].

2.2 Our Unique Approach

Our visualization approach is unique because it does not require additional display space for graph labels. Not only do we not need to worry about the NP-hard [2] problem but also we do not even need to seek open space for the labels. Thus our light-weight solution executes faster and requires less drawing space. As we will show later, our design also performs better in our user-evaluation studies in both static and dynamic modes.

3 The Design of GreenArrow

GreenArrow is a prototype system that comprises a suit of *interactive* and *animated* tools designed to visualize a graph with extended node and/or link labels. The system is implemented using Java and Java 2D graphics running on desktop Windows XP machines. This section describes the visualization and user-interface tools of GreenArrow.

3.1 Link Label Visualization

GreenArrow supports a variety of link label visualization in static, interactive, and dynamic modes. We will explain the rationale behind our designs and provide examples.

3.1.1 Link Label Overloading

The link label design shown in Figure 1b is partly inspired by a couple of distorted fisheye-like images presented by Ostromoukhov and Hersch [14] at ACM SIGGRAPH 95. It was reported 10 years ago that it would take minutes to render a 3"x4" graphic using the then state-of-the-art algorithm. Today's high-speed computers and high-resolution monitors make it possible for GreenArrow to render a large number of crisp, clear text streams onscreen dynamically in interactive time on a modest Windows P4 machine.

Generally speaking, we draw link labels as the links. The label font starts out larger from the source node and shrinks gradually until it reaches the destination node. The tapered label also indicates the direction of the graph label. Figure 2 shows a simple example of a link label extending from left to right.

Figure 2: The varying font sizes indicate the graph link direction.

If a label is too short to fill the distance between the nodes, we repeat the label to fill the space. If a label is too long, we either truncate the label in static mode or animate the label in dynamic mode. In other words, the label itself becomes the representation of the edge. Because of the repetition of the labels, we can resolve the overlap problems in most cases by looking at another portion of the edge.

3.1.2 Label along a Curve Link

GreenArrow uses a fast and elegant quadratic arc as the basis to compute the curve that the label letters are to be placed on. The distance between two graph nodes is first determined by the graph layout function. After the graph is drawn, the distance can be modified interactively by moving the graph nodes around. GreenArrow determines the width (w) and height (h) of the pre-transformed curve based on the positions of the two corresponding nodes, and then computes the pre-transformed coordinates (x, y) using the following equation (the coordinates are later rotated by an affine transform to their final location):

$$y = \frac{(x - w/2)^2}{(-w^2/(4h))}$$

A location to place a letter is retrieved by giving the curve drawing function a starting location, and the time parameter t, which represents how far along the *x-axis* we want our location on the curve. Because we could have more than one link from the same two nodes (see also Section 3.1.3 below), we added a curve ratio value for additional links so that they are not all covering the same area.

When the letters are placed on the arc, they are also rotated to give a better appearance of a curve. The letter rotation angle (θ_i) is determined by the following equation:

$$\theta_i = \sin^{-1}\left(\frac{y_i - y_{i-1}}{\sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}}\right)$$

Figure 3 shows two link labels before and after rotation. The quadratic arc approach is simple and effective. We have experienced no delay even when we draw a screen full of link labels with anti-aliasing turned on.

Figure 3: Two link labels before (top) and after (bottom) rotation.

3.1.3 Multiple Links between Two Nodes

Multiple links between a pair of nodes, especially those with different labels, can be represented as arcs as shown in Figure 4. Label behaviors such as the direction (left and right) and the orientation (up and down) of the text stream can be interactively selected by users or pre-defined in a style file. The link curvature is dynamically determined by number of links between the nodes.

Figure 4: Multiple graph links are represented as arcs.

3.1.4 Multi-resolution Link Label

GreenArrow draws label links as lines when the text is too small to be legible. When the scale is sufficiently large, GreenArrow then draws text as the links instead of a single-pixel line. This threshold is hand-tuned to be on the conservative side in that the lines may be replaced by text before the text is actually legible. Figure 5 shows five selected resolutions of the same link label. The second to last one from the bottom is a tapered solid line. The bottom one is just a one-pixel line.

Figure 5: Examples of five multi-resolution link labels.

We rely on the fact that humans are good at reading by shape [16]. So even if the individual characters of the text are not legible, the words themselves might be—or at least enough of the word—to find the desired link label. This ability is particularly useful when the labels come from a limited set of keywords, in which case the shapes of the word are further restricted.

3.1.5 Semi-Transparent Label

Similar to all the other graph drawing techniques, GreenArrow has difficulty showing the fine details surrounding a graph node with high in- or out-degree centrality. While this is an inherit deficiency of a graph structure, we offer a workaround to alleviate the congestion problem.

Figure 6a shows a force-directed [11] layout of a graph dataset collected from a previous drawing contest [5]. The graph contains two high-centrality root nodes highlighted by the green rectangle in Figure 6a. Figure 6b shows a zoomed view of the highlighted area. In this example, we also intentionally turn off the labels to accentuate the fact that we cannot see the links or even the labels in Figure 6b.

GreenArrow provides an option to clarify the area by applying the alpha blending technique to turn the dark links into semi-transparent ones. All the links connected to the hidden root highlighted with a cyan circle in Figure 6c suddenly become very visible in the figure. (Readers are referred to the high resolution electronic version of this paper where it is possible to zoom in and see details that are not apparent on small-scale print.)

3.1.6 Interactive Label Brushing

GreenArrow can be operated in an interactive mode where users can brush [1] a subset of links or nodes and then highlight them for other operations. The users can also adjust the font size of the brushed graph labels independently from the rest of the un-brushed links. This brushing process is particularly useful in the dynamic mode (discussed later) when the users need to focus on

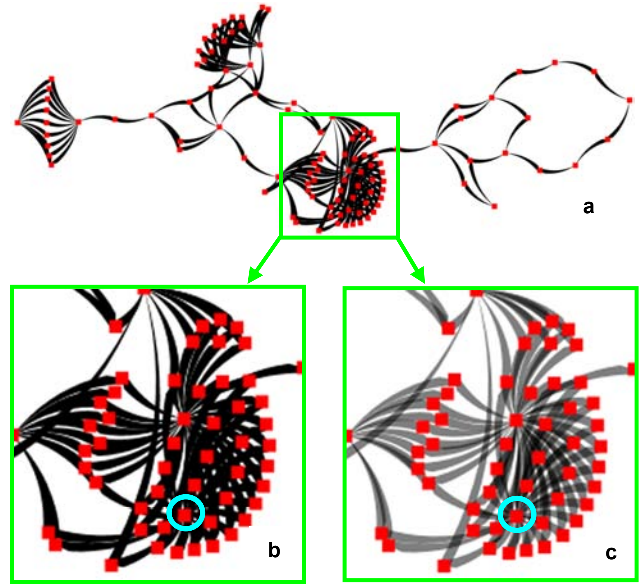

Figure 6: a) A force-directed layout of a telephone connection graph. b) Serious clutter surrounding a high-centrality (cyan) node. c) Alpha blending helps clear the clutter of the same node.

the changes of only brushed information and push everything else to the background of the visualization.

3.1.7 Dynamic Label Scrolling

When the text label is too long to be displayed along the graph link, GreenArrow has an option to animate the text by scrolling the words along the link. Readers are encouraged to watch the accompanying movie on the DVD proceedings to gain a feeling of the contribution of animation.

3.1.8 Link Label Orientation

In cases where the starting node is placed to the left of the ending node, the text in the corresponding link appears upside-down. GreenArrow provides an interactive option to re-orient the labels such that the text is upright.

3.2 Node Label Visualization

As the system name GreenArrow suggests, the technology is mostly about putting labels along a long straight or curved line. Nevertheless, we include two practical node label designs that work well together with the above link-label technology in our discussion. More examples on node label visualization can be found later in the evaluation study.

3.2.1 Circular Node Labels

We know that it does not take a long label to clutter a graph layout. Figure 7 shows two examples where a graph node represents the name of a person; the entire graph is shown in

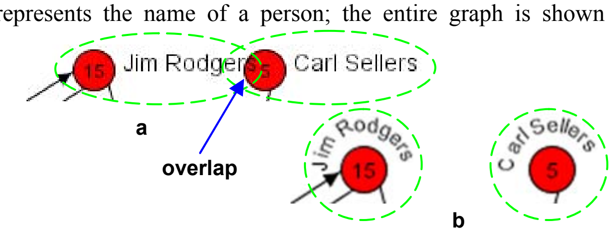

Figure 7: The a) basic approach and b) GreenArrow approach to show node labels.

Figure 9e. The footprints of the nodes and their labels are highlighted in green. The node label on the left of Figure 7a easily overlaps with its neighbor. On the other hand, the more compact design of the circular labels shown in Figure 7b is much less likely to overlap the other nodes or labels.

3.2.2 Dynamic Node Labels

Similar to the concept of a dynamic link label, a dynamic node label in GreenArrow is either a square- or circular-shaped window icon that shows the scrolling text labels. Figure 8 shows the concept and simulates an example of a dynamic node label connected by tapered line links.

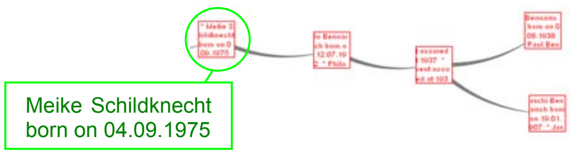

Figure 8: Example of a square-shaped dynamic label node with its contents highlighted in the green rectangle.

3.3 Additional Visualization Tools

GreenArrow has a user-friendly toolbox to support visual analysis of graphs. For example, GreenArrow has options to change the size and font of the labels as well as the color of the graph entities and background. Users can zoom, pan, and rotate the graphs by dragging their mice. They can highlight a particular link so that GreenArrow can zoom in and rotate the link and the graph to the right aspect for easy reading. Because the process is done through animation instead of screen swapping, users can easily register the motion and maintain the continuity of the visualization.

4 User Evaluation Study

We have conducted a series of empirical studies to evaluate the efficiency and effectiveness of GreenArrow. During the design phase of GreenArrow, we determined a set of definitive expectations for the final product. These expectations later became the foundation of the questions asked in our three empirical studies that involved 16 human subjects. Each of these studies investigated different aspects of the GreenArrow visualization. The first two focused mostly on individual graphics features such as a straight arrow versus a curved arrow. The third one, which is presented in this paper, asked the pragmatic question: does it perform as we expected?

4.1 Subjects, Setup, and Procedures

The third study was conducted on two separate days. Six subjects were tested—three on the first day and three on the second. The subjects were all junior researchers recruited from different departments at the Pacific Northwest National Laboratory. Half of them had seen GreenArrow briefly beforehand but had never used it before the study. All of them had programming experience. None of them had been involved in the two previous studies.

We conducted the test with one subject at a time using the same Windows desktop computer and 20-inch panel display in our visualization lab. All interaction was done using the mouse. One host asked the questions; the other recorded the results.

Before we started, we explained the purpose of the study and described two different visualization methods—traditional and GreenArrow—used in the tests. The subjects had the chance to see two sample graphs—one for each approach—on screen and manipulated them using the mouse. The host then asked a sample question that identified a particular node and then ended the pre-test training for individual subjects.

During the real tests, we would 1) ask a question, 2) show a graph on screen, 3) turn on the timer, 4) wait for the answer, 5) turn off the timer, and finally 6) record the results.

To provide a consistent environment for this study, we deactivated the brushing option and always showed all the labels—nodes and links—in the study. We used Arial font to draw the labels. On all of the tests except for the last one, the node labels were displayed as the ones similar to the example in Figure 7b. On the last test, the node labels were shown as dynamic scrolling streams similar to the ones shown in Figure 8.

4.2 Study Graph Data

There were six pairs of graph datasets that generated six pairs of visualizations shown in Figures 9a to 9l. The top visualization of each of these sub-figures was generated using a traditional approach; the bottom one was generated using the GreenArrow approach. Each dataset pair is actually representative of two slightly altered graphs. They were identical in size and shape but often with different text for the nodes and links. We also changed the orientation of the two slightly altered graphs and made them distinguishable in the test.

The zoomed graphs shown in Figure 9 are not meant to show the fine details of the graphs. We use them to show the overall structure of our customized graphs that support individual questions described in Section 4.3.

4.3 Study Questions

We developed six questions to evaluate GreenArrow. The rationale behind each question and the hypothesis we made before the tests are described below in the *objective* paragraph. In all cases, we collected results that measured the accuracy of the answer (%), elapsed time, user satisfaction (0 to 5; 0: bad experience; 5: good experience), and in some cases, mouse interaction count (click, move, zoom, pan, and turn).

4.3.1 Question 1 – Find a specific node

Observations: Accuracy, elapsed time, user satisfaction.

Test 1: Show a 19-node graph (Figure 9a), traditional view, in which there is a maximum of one link going from one node to another. Link labels appear horizontally (like Figure 1a) and are moderate in length. Node labels are positioned horizontally close to the nodes. Nodes are fictitious people. Links are their relationship (friend, sibling, etc.). The subject could visually scan the entire layout and point to the requested node, so no mouse-click information was collected.

Test 2: Show a slightly altered graph (Figure 9b) using GreenArrow and a different rotation. The graph has round nodes with labels circling around them (like Figure 7b). All graph link labels are on. The subject had to visually scan the entire layout and point to the requested node.

Objective: We hypothesized that the user would find the GreenArrow layout to be more visually accommodating and make it easier to identify specific node labels than a traditional graph view. Various features of the GreenArrow layout provide distinct visual cues to discriminate node label text from link label text; while the traditional view positions horizontal lines of text in proximity to nodes and links, the GreenArrow view positions the two types of text differently and gives them distinct appearances.

4.3.2 Question 2 - State all of the link names that are associated with the highlighted node

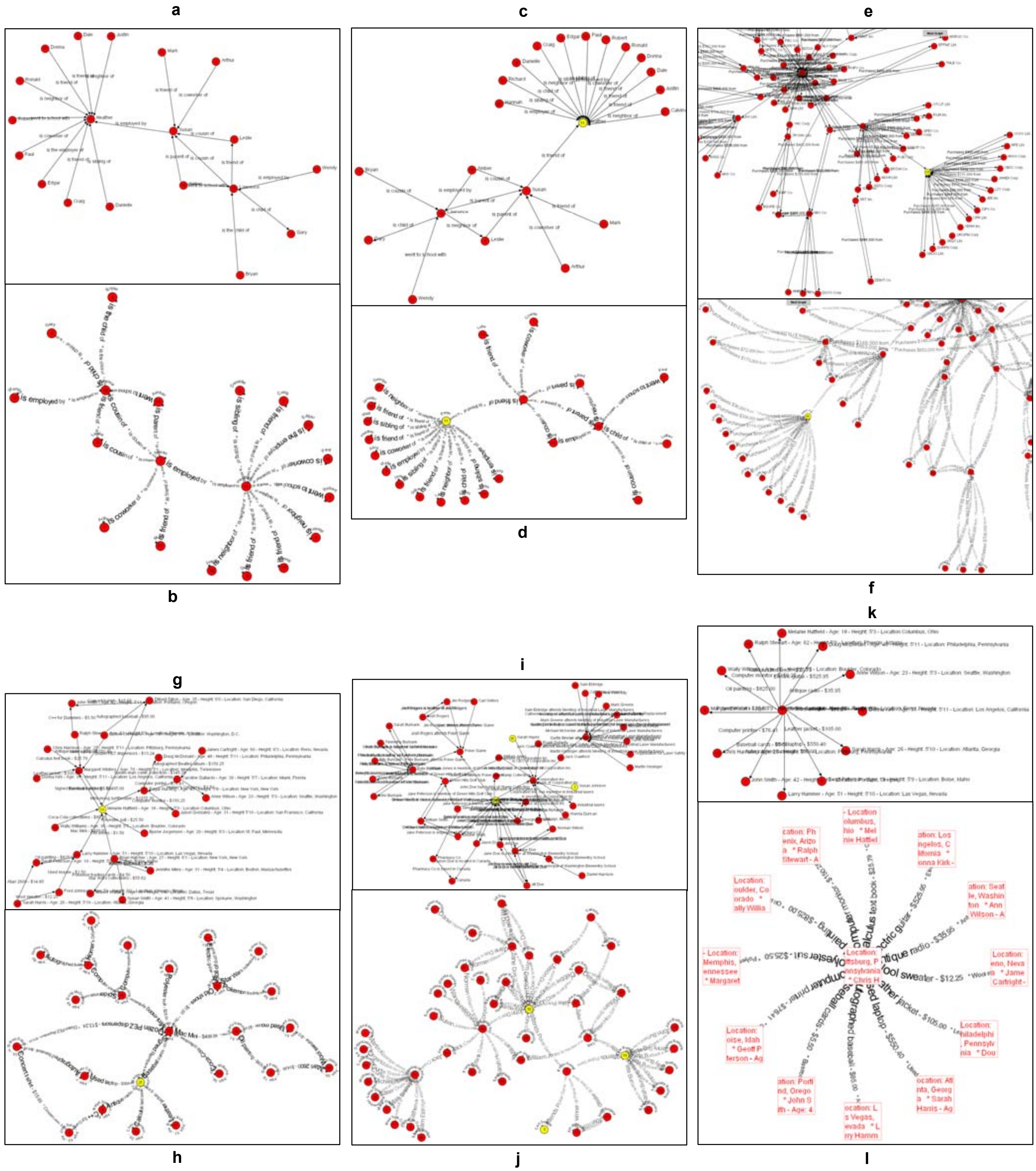

Figure 9: Visualize six graphs using both the traditional approach and the GreenArrow approach to show node and link labels. The nodes highlighted in yellow are part of the questions.

Observations: Accuracy, elapsed time, mouse interaction count, user satisfaction

Test 1: Provided a 22-node graph (Figure 9c), traditional view, in which one node (#11) is pre-highlighted in yellow. The highlighted node has 13 fan-in links. Link labels appear horizontally and are a little longer than moderate in length. Node labels are positioned close to the nodes. The nodes are fictitious people and the links are their relationship. The subjects were asked to name all of the labels of the links that were associated with the highlighted node. The users could use mouse interactions to move nodes around in order to clarify the text.

Test 2: Provided a slightly altered graph (Figure 9d), different rotation, GreenArrow view, in which the same node (#11) is pre-highlighted in yellow.

Objective: We hypothesized that the subject would find the horizontal labels in the traditional graph view confusing, as label lengths would be long and labels overlap. The subject would need to pan and zoom to distinguish which label was associated with which link and to read the text. The labels in the GreenArrow view would be more separated and the scrolling text would allow the subject to read the entire string within a small screen area.

4.3.3 Question 3 – How many links enter the selected node?

Observations: Accuracy, elapsed time, mouse interaction count, user satisfaction

Test 1: Provided a 136-node graph (Figure 9e), traditional view, in which there are complex structures but also significant fan-outs. Link labels appear horizontally and are moderate in length. One node (#126) is marked in yellow; this node is attached to 12 fan-out links and 10 fan-in links. The links have a mixture of bidirectional and single direction links. Nodes are fictitious people, and the links are their transactions from an online auction site. The subject had to say the observed number of fan-in links that were attached to the colored node.

Test 2: Using a slightly altered graph (Figure 9f), different rotation, color another node that is also adjacent to a significant fan-out. There are 13 fan-in and 5 fan-out links. This was displayed with the GreenArrow layout. Subjects were asked to state how many fan-in links were attached to the highlighted node.

Objective: We hypothesized that the traditional view's layout of links would cause arrowheads in fan-outs to be positioned very close to each other, causing ambiguity. The subject would either have to zoom way in to distinguish the arrowheads or look at other, less prominent, visual cues to determine the direction of each link. The GreenArrow view, on the other hand, offers a tapered arc of text. The subject would be able to observe taper direction and also text direction and determine link direction from those cues.

4.3.4 Question 4 – Select all nodes that can be reached by the highlighted node

Explanation: Imagine that the links are one-way streets, and you are in a car at the highlighted node. Which other nodes could you reach without going the wrong way through a one-way node?

Observations: Accuracy, elapsed time, mouse interaction count, user satisfaction

Test 1: Provided a 24-node graph (Figure 9g), traditional view, that contains interesting structures. Nodes are fictitious people, and the links are their transactions from an online auction site. One node (#13) is colored yellow, different from the others. There are seven links exiting that node and a couple more exiting neighboring nodes. The subject had to identify all links that could be followed from the highlighted node and the neighboring nodes by clicking on the respective links.

Test 2: Using a slightly altered graph (Figure 9h), different rotation, and in GreenArrow view, color a different node (#21) from the one used in Test 1 with seven fan-out links.

Objective: We hypothesized that GreenArrow's suitability allows the subject to readily determine link direction. We wanted to determine whether the tapered arc and text direction cues in GreenArrow were preferable to the traditional line and arrowhead approach.

4.3.5 Question 5 – Identify the relationship among the three people that are represented by the highlighted nodes

Observations: Accuracy, elapsed time, mouse interaction count, user satisfaction

Test 1: Provided a 46-node graph (Figure 9i), traditional view that has 3 nodes (#45, #2, #32) highlighted, each one link away from each other. The central portion of this graph is busy, and link label lengths are moderate. Nodes are fictitious people, organization, companies, hobbies, events, and locations. The subject had to deduce from the links the relationship that the three neighboring nodes had with each other.

Test 2: Using the same graph (Figure 9j), possibly rotated at different angle, and in GreenArrow view, color another three neighboring nodes (#19, #5, #32) in yellow.

Objective: We hypothesized that the traditional view, with its horizontally positioned labels, would appear very busy and have some labels that interfered with other labels. The GreenArrow view, on the other hand, would reduce such clutter by printing the link label text directly on the link path.

4.3.6 Question 6 – Name the label of the central node

Observations: Accuracy, elapsed time, mouse interaction count, user satisfaction

Test 1: Provided a 13-node graph (Figure 9k), traditional view, with lengthy node labels, positioned horizontally. The subject had to name the label that was associated with the central node, the node with the busiest link activity. (Busy, central link activity should be immediately apparent.)

Test 2: Provided a similar graph (Figure 9l) in GreenArrow view (but rotated differently), and enable all node labels to be displayed in the scrolling box mode.

Objective: We hypothesized that the traditional graph view would contain an overwhelming amount of text, much of which interfered with other portions of text. The interference could be clarified by zooming in and panning. The GreenArrow view, on the other hand, was expected to readily present label information. The scrolling box view would possibly allow the subject to read the node label without requiring any interaction.

4.4 Study Results

Our discussion emphasizes the major similarities and differences between our pre-test hypotheses and the final outcomes. Because of very limited space, we focus mainly on primary observation data instead of secondary information such as the statistical distributions of our observations. However, it is our desire to present all of our analyses in detail from all three studies at a different forum later.

Figures 10a through 10d show four bar graphs that depict the averaged results in the four collected categories: accuracy, elapsed time, user satisfaction, and number of mouse clicks. In all four graphs, the blue bars represent the average scores of the tasks using traditional visualization, whereas the green bars represent the average scores using GreenArrow.

4.5 Quantitative Analysis

Following the design of our user study, we divide our discussion into four sections based on the scores of the four bar graphs shown in Figure 10.

4.5.1 Elapsed Time

We hypothesized that the traditional method was weak at handling moderate (Question 1) to longer labels (Question 2), especially in more complex graphs (Questions 3 and 4) or graphs with many high centrality nodes (Question 6). Our study results shown in Figure 10a concur with five of our six hypotheses. In some cases (Questions 2, 3, and 5), GreenArrow performs ~50% better (or faster) than the traditional visualization method.

However, we were puzzled with the results of Question 4 that show a poor performance by GreenArrow. All six subjects consistently spent more time answering Question 4 when using GreenArrow.

4.5.2 Number of Mouse Interactions

We hypothesized that the unique design approach of GreenArrow that overlay links and labels would create fewer overlaps in many cases (Questions 1, 2, 4, and 6), especially surrounding nodes with high centrality (Questions 3 and 5). Because the overlaps would cause ambiguity, we expected our subjects would need to manipulate (zoom, rotate, etc.) the graph to the right positions before they could answer our questions. We counted each mouse interaction as one movement and the average results of the six questions are shown in Figure 10b. The results shown in Figure 10b are mostly consistent with our pre-test expectations, with the exception of Question 4. In answering Question 4, our subjects worked longer (i.e., used more mouse clicks) with GreenArrow to come up with their answers.

4.5.3 Accuracy of the Answers

Our objective was to discover the advantages of GreenArrow in terms of effectiveness and efficiency in answering simple questions using smaller graphs (< 25 nodes in Questions 1, 2, 4, and 6; 44 in Question 3; and 74 in Question 5) on a 20-inch panel screen. We expected the subjects would give all the right answers, given enough time and manipulations of the graph. So we were not surprised to find that the scores of both visualizations were almost identical in Figure 10c, with the exception that GreenArrow slightly (3%) outperformed the traditional visualization in Question 3.

4.5.4 User Satisfaction

GreenArrow was designed to visualize graphs with extended labels. We hypothesized that if we could provide a tool that gave more accurate answers in less time and fewer interactions, our subjects would appreciate the *compact* and *intuitive* design of GreenArrow more than the traditional way. It turned out that, except in Question 1, we were correct: the traditional visualization scored slightly higher than GreenArrow. Evidently, the graph used in Question 1 (Figure 10a) was also the simplest one with the shortest node and link labels among all six test graphs.

4.6 Post-Test Interview

Those who participated in the design of GreenArrow and the above study were, in general, satisfied with its outcome. However, everyone was puzzled with the negative answers (Figures 10a and 10b) in Question 4 scored by all of our subjects. In both cases, the traditional visualization consistently out-scored the GreenArrow design in both the time and interaction categories.

We conducted a post-test interview with our subjects, hoping to find a clue for the cause of our failure. When the subjects were presented with the results depicted in Figure 10, our discussion quickly turned into the design breakdown of Question 4.

It turned out that although the use of a simple line arrow is a daily experience to indicate direction, the tapered link design (when overlaid with labels) does not possess the same intuitive power. ALL of our six subjects were confused by the directions indicated by the tapered ends of the link when they traced the path among a set of nodes. *Without sufficient training and experience, they tended to forget the sense of direction attached to the tapering*. This resulted in longer elapsed time and the higher error rates shown in Figures 10a and 10b.

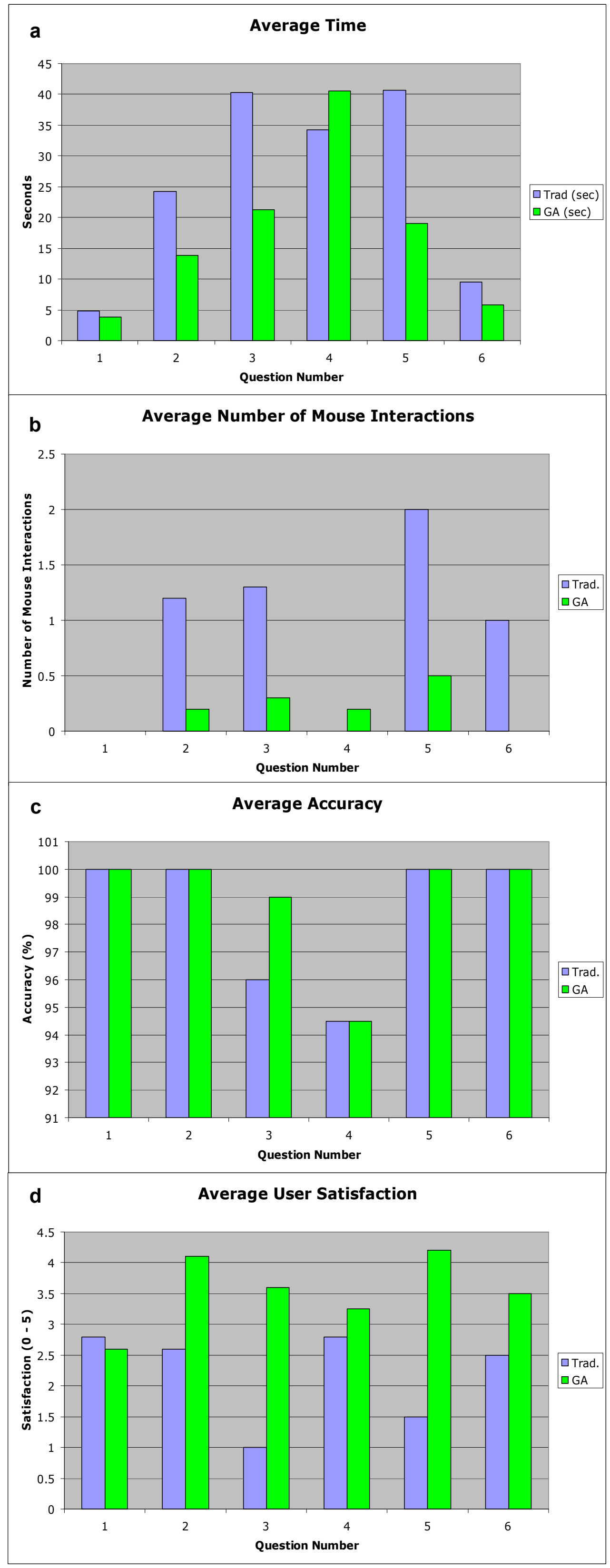

Figure 10: a) to d) Average scores of each of the six questions are (blue=traditional, green=GreenArrow) shown separately in four categories.

4.7 The *Dynamic* Factor

After discovering the weakness of our design, we conducted extensive experiments and eventually identified the source of our mistake. We strived for the artistic creativity of GreenArrow's overlay design and totally failed to appreciate the dynamic advantages provided by GreenArrow—the tapered link with "scrolling" labels. In other words, we can use motion to supplement the static tapered links and remind users of the direction of the otherwise static non-arrow link.

We re-designed a new version of Question 4 that included the use of scrolling fonts to enhance the direction of the links. This time we hypothesized that the use of scrolling text label would supplement the direction information provided by a tapered link design. The subjects would provide more accurate answers in less time and fewer mouse interactions.

We were able to bring back five of the six original participants to answer the new Question 7—a revised version of Question 4—with the scrolling text turned on. The results showed that our subjects took less time (31.6 seconds vs. 40.5 seconds) and fewer mouse action counts (0.0 vs. 0.2) to give more accurate answers (100% vs. 94.5%). The overall satisfaction of the GreenArrow design in solving the stated graph problem also improved from 3.25 to 4.1 (out of a 5.0 scale).

4.8 Discussion

A great discovery found in the user study was the surprisingly strong contribution of animation in showing the link directions when users identify a path within a graph.

A major disappointment of the GreenArrow design was the weak performance by the static version of the tapered links when they were applied to show labels surrounding high-centrality nodes of a complex graph.

The steady performance of the traditional visualization when the graph is small and its labels are short was at least a mild surprise in our study. The fact that they scored head-to-head in all four categories in Figure 10 shows that the design of a traditional layout is hard to beat, unless the underlying graphs possess a high complexity and/or long labels.

In short, the test scores of the user study show a reduced time to answer the question with GreenArrow; the accuracy is on average at least as good or better; less mouse interaction is required to answer the questions; and the average satisfaction is much better for the most part.

5 Conclusions and Future Work

We have presented a practical technique to visualize graphs with extended node and link labels. We developed a system prototype, known as GreenArrow, to extensively study the strengths and weaknesses of our visualization design. This paper presents major features of GreenArrow and the results of a complete user study that challenged our design rationale and discovered some of the design weaknesses.

Our immediate work is to organize the results from all three studies, analyze them from a statistical standpoint, and come up with a comprehensive evaluation report on GreenArrow. The technology, if proven to be viable, will be integrated into our visual analytics work on very large semantic graphs.

Acknowledgements

This work has been sponsored by the National Visualization and Analytics Center™ (NVAC™) located at the Pacific Northwest National Laboratory in Richland, WA. The Pacific Northwest National Laboratory is managed for the U.S. Department of Energy by Battelle Memorial Institute under Contract DE-AC05-76RL01830.

References

[1] William S. Cleveland, *Visualizing Data*, Hobart Press, 1993.
[2] Michael R. Garey and David S. Johnson, *Computer and Intractability. A Guide to the theory of NP-completeness*, WH Freeman & Company, New York, 79.
[3] Jon Christensen, Joe Marks, and Stuart Shieber, "An Empirical Study of Algorithms for Point-Feature Label Placement," *ACM Transactions on Graphics (TOG)*, volume 14, issue 3, pages 203-232, ACM Press, 1995.
[4] Giuseppe Di Battista, Peter Eades, Roberto Tamassia, and Ioannis G. Tollis, *Graph Drawing: Algorithms for the Visualization of Graphs*, Prentice Hall, 99.
[5] GD96 Graph Drawing Contest, http://www.research.att.com/conf/gd96/contest.html.
[6] GD2005, 13th International Symposium on Graph Drawing 2005, http://www.ul.ie/gd2005/.
[7] Claudia Iturriaga, *Map Labeling Problems*, Ph.D. Thesis in Computer Science, University of Waterloo, 1999. See also http://www.cs.unb.ca/profs/citurria/Research/.
[8] Konstantinos G. Kakoulis and Ioannis G. Tollis, "A Unified Approach to Automatic Label Placement," *International Journal of Computational Geometry and Applications*, volum2 13, number 1, pages 21-60, 2003.
[9] Konstantinos G. Kakoulis and Ioannis G. Tollis, "An Algorithm for Labeling Edges of Hierarchical Drawings," Lecture Notes in Computer Science, *Proceedings of the 5th International Symposium on Graph Drawing*, volume 1353, pages 169-180, Springer-Verlag, London, UK, 1997.
[10] Konstantinos G. Kakoulis and Ioannis G. Tollis, "On the Edge Label Placement Problem," Lecture Notes in Computer Science, volume 1190, *Proceedings of the Symposium on Graph Drawing*, pages 241-256, Springer-Verlag, London, UK, 1996.
[11] T. Kamada and S. Kawai, "An Algorithm for Drawing General Undirected Graphs," *Information Processing Letters*, volume 31, issue 1, pages 7-15, Elsevier North-Holland, April 1989.
[12] Takayuki Kameda and Keiko Imai, "Map Label Placement for Points and Curves," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, volume E86-A, number 4, pages 835-840, April 2003.
[13] Gunnar W. Klau and Petra Mutzel, "Combining Graph Labeling and Compaction," *Lecture Notes in Computer Science*, volume 1731, *Proceedings of the 7th International Symposium on Graph Drawing*, pages 27-37, Springer-Verlag, London, UK, 1999.
[14] Victor Ostromoukhov and Roger D. Hersch, "Artistic Screening," *Proceedings of SIGGRAPH '94*, ACM Computer Graphics, Annual Conference Series, pages 218-228, ACM Press, 1995. See also http://citeseer.ist.psu.edu/ostromoukhov95artistic.html.
[15] T. Kolda, D. Brown, J. Corones, T. Critchlow, T. Eliassi-Rad, L. Getoor, B. Hendrickson, V. Kumar, D. Lambert, C. Matarazzo, K. McCurley, M. Miller, N. Samatova, D. Speck, R. Srikant, J. Thomas, M. Wetheimer, and P.C. Wong, *Data Sciences Technology for Homeland Security Information Management and Knowledge Discovery*, *Report of the DHS Workshop on Data Sciences*, 2004. See http://csmr.ca.sandia.gov/~tgkolda/pubs/DSW2004_LoRes.pdf.
[16] Wolfgang Kohler, *Gestalt Psychology*, Liveright, NY, 1975
[17] Michael Schreyer and Günther R. Raidl, "Letting Ants Labeling Point Features," *Proceedings IEEE Congress on Evolutionary Computation (CEC '02)*, pages 1564-1569, IEEE Press, 2002.
[18] Kozo Sugiyama, *Graph Drawing and Applications*, World Scientific Publishing, 2002.
[19] James J. Thomas and Kristin Cook, *Illuminating the Path, the Research and Development Agenda for Visual Analytics*, IEEE CS Press. In press. See http://nvac.pnl.gov/agenda.stm for an executive summary.
[20] Frank Wagner and Alexander Wolff, "A Combinatorial Framework for Map Labeling," *Proceedings of the 6th International Symposium on Graph Drawing, Lecture Notes in Computer Science*, vol 1547, pages 316-331, 1998.
[21] Alexander Wolff, *The Map-Labeling Bibliography*, 2005. See http://i11www.ira.uka.de/map-labeling/bibliography/.
[22] Alexander Wolff, Lars Knipping, Marc van Kreveld, Tycho Strijk, and Pankaj K. Agarwal, "A Simple and Efficient Algorithm for High-Quality Line Labeling," *Proceedings GIS Research UK 7th Annual Conference (GISRUK'99)*, pages 146-150, Southampton, April 1999.