A System for Visualizing and Analyzing the Evolution of the Web with a Time Series of Graphs

Masashi Toyoda
toyoda@tkl.iis.u-tokyo.ac.jp

Masaru Kitsuregawa
kitsure@tkl.iis.u-tokyo.ac.jp

Institute of Industrial Science, University of Tokyo
4-6-1 Komaba Meguro-ku, Tokyo, JAPAN

ABSTRACT

We propose WebRelievo, a system for visualizing and analyzing the evolution of the web structure based on a large Web archive with a series of snapshots. It visualizes the evolution with a time series of graphs, in which nodes are web pages, and edges are relationships between pages. Graphs can be clustered to show the overview of changes in graphs. WebRelievo aligns these graphs according to their time, and automatically determines their layout keeping positions of nodes synchronized over time, so that the user can keep track pages and clusters. This visualization enables us to understand when pages appeared, how their relationships have evolved, and how clusters are merged and split over time. Current implementation of WebRelievo is based on six Japanese web archives crawled from 1999 to 2003. The user can interactively browse those graphs by changing the focused page and by changing layouts of graphs. Using WebRelievo we can answer historical questions, and to investigate changes in trends on the Web. We show the feasibility of WebRelievo by applying it to tracking trends in P2P systems and search engines for mobile phones, and to investigating link spamming.

Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia—*Navigation*; H.3 [**Information Storage and Retrieval**]: Information Search and Retrieval

General Terms

Design, Experimentation

Keywords

Visualization, Web graph, evolution, link analysis, link spamming

HT'05, September 6–9, 2005, Salzburg, Austria.
Copyright 2005 ACM 1-59593-168-6/05/0009 ...$5.00.

1. INTRODUCTION

The Web has been dramatically growing and changing its hyperlink structure by reflecting real and virtual activities. When major events occur, various web pages about these events are created, and then pages with important information become pointed to by many pages. Such events could be war and terrorism in the real world, and could be appearance of a new type of software such as P2P file sharing systems in the virtual world.

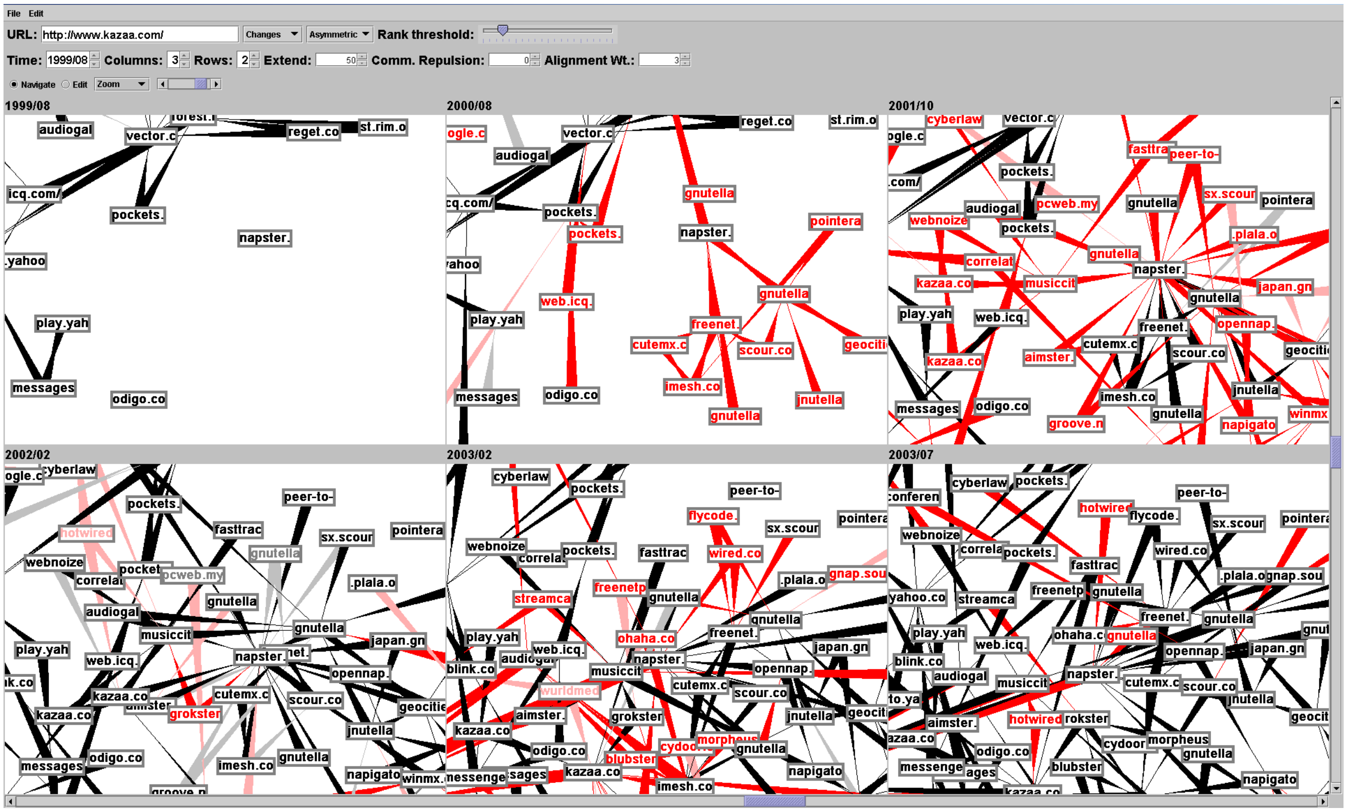
Moreover, the structure of the Web is now intentionally changed to control ranking of electronic commerce sites in search engines. The main reason is the fact that higher ranked sites have ability to pull in more customers. Such commerce sites mainly target link based ranking method such as PageRank[2] that gives high scores to pages pointed to by many other pages with high scores. One way to manipulate such scores is concentrating links to their sites. Then those sites seem to be popular, and can collect high scores. Such manipulation is called link spamming.

Since hyperlinks represent attention of page authors to the destination pages for better or worse, we could see various changes in trends on the Web from the evolution of the hyperlink structure. Tracking structural changes in the Web is important in the following situations:

- Answering historical questions about web pages, such as when pages appeared and disappeared, and how their relationships changed over time.
- Investigating link spamming structure to eliminate its effect and to correct ranking in search engines.
- Observing and tracking social and cultural trends over time for sociological research.

We propose a system WebRelievo (WEB RELatIonship EVOlution) for visualizing and analyzing the evolution of the web structure based on a large web archive including snapshots of web pages periodically collected by crawlers. Currently, we use six web archives of Japanese web snapshots crawled from 1999 to 2003.

WebRelievo visualizes a time series of web graph, in which nodes are web pages and edges are relationships between pages, such as hyperlinks and results of link analysis. To show the structure of the web from various aspects, WebRelievo provides several variations of the web graph. The most basic graph consists of web pages and hyperlinks. However, this graph is too complicated to understand and to visualize its structure. Since famous web pages are linked to



URL: http://www.kazaa.com/

Rank threshold:

Time: 1999/08 Columns: 3 Rows: 2 Extend: 50 Comm. Repulsion: 0 Alignment Wt.: 3

Figure 1: A screen snapshot of WebRelievo

by thousands of other pages, it is difficult to display even a subgraph of the web in limited screen space. Therefore, we also provide another graph based on link analysis that calculates related pages by extracting densely connected in a certain pattern. In this case, relationships between pages are results of link analysis.

Figure 1 shows a screen shot of WebRelievo. Six graphs represent the same part of those six web snapshots. From each web snapshot, WebRelievo extracts subgraphs around focused pages given by the user. In Figure 1, each node represents a web page, and each edge represents that nodes at both ends are considered to be related by a link analysis method. These graphs are aligned from left to right then top to bottom according to their time. Positions of URLs are synchronized over time, so that the user can track the changes in graphs.

WebRelievo differs from previous work in visualizing evolution of clustered graphs to show the overview of changes in large graphs (See Figure 3). A cluster is a set of densely related web pages, and represents some topic in the Web. WebRelievo shows how clusters have been organized and how their members changed over time. In addition, clusters can be partially expanded to see detailed changes in the overview.

This visualization allows the user to understand when pages appeared and disappeared, and how their relationships have changed over time. In addition, the user can interactively browse the evolution of graphs by changing the focused pages and by dragging nodes.

The rest of this paper is organized as follows. Section 2 shows related work. The system architecture is described in Section 3. Section 4 explains details of the time series of web graphs. Section 5 describes how to visualize the evolution and how the user can interact with WebRelievo. Section 6 shows some evolution examples with WebRelievo. Finally, we conclude in Section 7.

2. RELATED WORK

2.1 Visualizations of Evolution

There have been various work on visualizing evolution of information structure. Chen examined animated visualization of the evolution of co-citation networks of scientific publications [3]. Chen compared two link reduction techniques to show which one is suitable for animation of the evolution [4]. Diehl et al. proposed a graph drawing technique for making animations from a sequence of evolving graphs [10, 9]. This technique first builds an union of all graphs, so called a super graph, and calculates a layout of the super graph. Then, it generates frames of the animation based on the layout of the super graph. In this way, it avoids drastic movements of nodes in the animation. Erten et al. also built a framework for drawing evolving graphs in various visualization methods such as aligning graphs in 2D and overlaying graphs in 3D space [12]. For visualizing graphs, this framework extended a force-directed model [14] to synchronize node positions. They applied the framework to the evolution of the software structure [7] and computing literature [11]. WebRelievo differs from previous work in visualizing evolving graphs of clusters based on their merging and splitting behavior to show the overview of changes in large graphs.

Chi proposed the time tube technique [6, 5] to visualize the evolution of a single web site structure, and accesses patterns on that site. It visualizes the hierarchical structure of the web site as a disk tree or a 3D cone tree, in which the root page is put on the center, and child pages fan out from the root. Multiple views are created for each time period, and aligned left to right according to their time, so that the user can observe changes over time. WebRelievo puts focus on visualizing relationships between multiple web sites in large web archives, and does not restrict the structure to the hierarchy.

Our previous work [18, 19] visualized evolution of web communities. A web community is a set of web pages with a common interest on a topic. In [18], we proposed a method for extracting all web communities and their relationships from a single web archive. In [19], we extracted all web communities from periodically crawled web archives, and visualized changes of these communities, such as growth and shrinkage. Rather, WebRelievo visualizes changes of relationships from cluster level to page level. It can be used for detailed examination of web communities.

2.2 Link Analysis

In addition to the hyperlink structure, WebRelievo can use results of link analysis to represent relationships between pages. WebRelievo uses a related page algorithm (RPA) that takes a seed page as an input, and outputs related pages to the seed.

There are several RPAs based on the notion of *authorities* and *hubs* proposed by Kleinberg [15]. An authority is a page with good contents on a topic, and is pointed to by many good hub pages. A hub is a page with a list of hyperlinks to valuable pages on the topic, that is, points to many good authorities. This structure commonly occurs in various topics in the Web, such as fan pages of a baseball team, and official pages of computer vendors.

HITS [15] is an algorithm that extracts authorities and hubs from a given subgraph of the Web with efficient iterative calculation. HITS first builds a subgraph of the Web near the seed, and extracts densely connected authorities and hubs in the graph. Then authorities are returned as related pages. As a result, HITS extracts frequently co-cited pages as authorities. There are some variants of RPA based on HITS, such as Companion [8].

There are also some RPAs that are not based on HITS. Lempel and Moran [16] proposed another approach based on a random walk model for calculating authorities. Flake et al. [13] redefined a community including given seed pages as a subgraph that is separated from the Web using a maximum flow/minimum cut framework.

3. ARCHITECTURE

The architecture overview of WebRelievo is shown in Figure 2. WebRelievo is based on three databases built from our web archive. We use six snapshots of Japanese web pages crawled from 1999 to 2003 (See Table 1). Our crawler collected pages in the breadth-first order. From 2001, the number of pages became more than twice of the 2000 archive, since we improved the crawling rate. Until 2002, we collected pages in only .jp domain. From 2003, we began to collect pages in other domains, such as .com, if they are written in Japanese. Those collected documents are stored in the document archive on the right-bottom of Figure 2. Each document can be retrieved by its URL and the time of crawling.

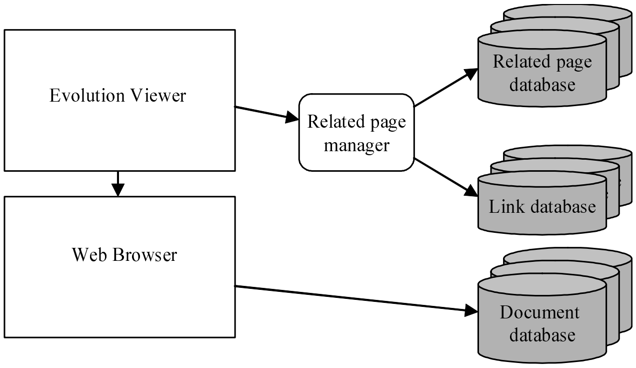


Figure 2: Architecture overview

Year	Period	Crawled pages	Total URLs	Links
1999	Jul. to Aug.	17M	34M	120M
2000	Jun. to Aug.	17M	32M	112M
2001	Oct.	40M	76M	331M
2002	Feb.	45M	84M	375M
2003	Feb.	66M	384M	1058M
2003	Jul.	98M	601M	1587M

Table 1: Details of web archives

From each archive, we built a link database (on the right-center of Figure 2) with URLs and links by extracting anchors from all pages in the archive. Our link database included not only URLs inside the archive, but also URLs outside pointed to by inside URLs. As a result, the graph included URLs outside .jp domain, such as .com and .edu. Table 1 also shows the number of links and the total URLs. For efficient link analysis, each link database is implemented as a main-memory database that provided out-links and in-links of a given URL. Its implementation was similar to the connectivity server [1].

The related page database on the right-top of Figure 2 is used to speed up the retrieval of relationships. It stores pre-calculated results of RPA for popular URLs. The popularity of a URL is determined by the number of in-links. In our implementation, we pre-calculate related pages for URLs that have three or more in-links from other web servers.

The related page manager, in the center of Figure 2, provides information of relationships to the evolution viewer on the top-left of Figure 2. Various models of relationships and neighborhood are implemented here. This manager builds subgraphs for the focused pages using those databases. When required URLs are not stored in the related page database, it calculates related pages by RPA using the link database. The RPA used by the manager can be changed to any RPA based on link analysis. In that case, the related page database should be also replaced.

The evolution viewer is the user interface of WebRelievo. The viewer displays graphs provided by the related page manager. We use TouchGraph [17], a graph drawing library for Java, to layout graphs, and modify the layout algorithm to synchronize node positions. The web browser (e.g. Mozilla) is used to browse contents of URLs designated by the user. The browser accesses the document database to show past documents.

4. TIME SERIES OF WEB GRAPHS

WebRelievo visualizes a time series of Web graphs (TG), in which each graph represents the structure of each snapshot in the web archive:

$$TG = \{G_t = (V_t, E_t) | 1 \leq t \leq T\}.$$

The subscript t denotes the time when each archive was crawled (1 is the first time and T is the last time). In TG, each graph G_t is a directed graph with weighted edges that consists of a set of nodes V_t and a set of weighted edges E_t. In each graph, nodes represent web pages, and weighted edges represent relationships between these pages. Relationships might be changed according to users' requirements. The most basic relationship between pages is hyperlink itself. Since the hyperlink structure is too complicated to visualize and to understand, simpler relationships are required in many cases. As an example, we can use results of related page algorithms as relationships between pages. That is, we visualize how each page derives other pages by RPA. In this case, each directed edge $e \in E_t$ from a node p to another node q, represents that p derives q as one of the top related pages, and has a higher weight when it is ranked higher.

4.1 Subgraphs and Clustering

Since each G_t is too large to display in a normal desktop screen, a subgraph should be extracted from each G_t according to the user's interest. In WebRelievo, the user's interest is given as a set of focal pages F, which are designated by URL or keyword search, and can be interactively added or deleted. WebRelievo extracts the neighborhood of F as the subgraph in each time, and displays changes in relationships by a time series of these subgraphs:

$$TG(F) = \{G_t(F) = (V_t(F), E_t(F)) | 1 \leq t \leq T\}.$$

The neighborhood of F is also determined according to the user's requirements. When edges of G_t are hyperlinks, the neighborhood of F might be a set of pages pointed to by or pointing to every $u \in F$. When edges of G_t are results of RPA, the neighborhood of F might be top N pages derived from p by RPA.

To show appearance and disappearance of neighborhood pages over time, WebRelievo extracts the neighborhood of F in every time, and tracks all of these pages over time. The following is the process to build the $TG(F)$:

1. For each time t, extract a set of neighborhood nodes $R_t(F)$.
2. For each $G_t(F)$, $V_t(F) = V_t \cap (\cup_t R_t(F))$.
3. For each $G_t(F)$, $E_t(F) = \{(u, v) \in E_t | u, v \in V_t(F)\}$.

WebRelievo also supports clustering of nodes. By collapsing multiple nodes into a single cluster node, we can show an overview of web graphs, and can save calculation cost of graph layout. Clustering methods should be variable according to the user's requirements. For example, clustering by server names may be used when edges in G_t are hyperlinks.

By clustering, each $G_t(F)$ is replaced to the clustered graph, $G'_t(F) = (V'_t(F), E'_t(F))$, when a set of clusters of $V_t(F)$ is given by some clustering method as follows:

$$C_t(F) = \{C^k_t \subset V_t(F) | C^i_t \cap C^j_t = \emptyset \ for\ all\ i \neq j\}.$$

For simplicity, we suppose that clusters are disjunctive sets of pages. Nodes in clusters are replaced by cluster nodes, and edges between cluster nodes are created as follows:

$$\begin{aligned} V'_t(F) &= C_t(F), \\ E'_t(F) &= \{(C^i_t, C^j_t) | \exists (u, v) \in E_t(F), u \in C^i_t \land v \in C^j_t\}. \end{aligned}$$

4.2 Variations of Graphs

As mentioned above, TG may have various combinations of relationships (E_t), neighborhood ($R_t(F)$), and clustering method($C_t(F)$). Currently, WebRelievo implements several combinations as follows. Further combinations can be easily implemented and added to WebRelievo.

Relationships

As relationships, WebRelievo can use hyperlinks or results of a RPA. Currently, as the RPA, we use a variation of Kleinberg's HITS algorithm, so called Companion–[18]. Using this RPA, we can see connections between densely co-cited pages. That is, pages with the same topic are tend to be connected. Note that the algorithm can be easily replaced.

Neighborhood

When relationships are hyperlinks, the neighborhood can be (1) pages pointed to by $u \in F$, (2) pages pointing to $u \in F$, (3) both (1) and (2), or (4) top N hubs and authorities derived from $u \in F$. (4) is useful for investigating the results of RPA.

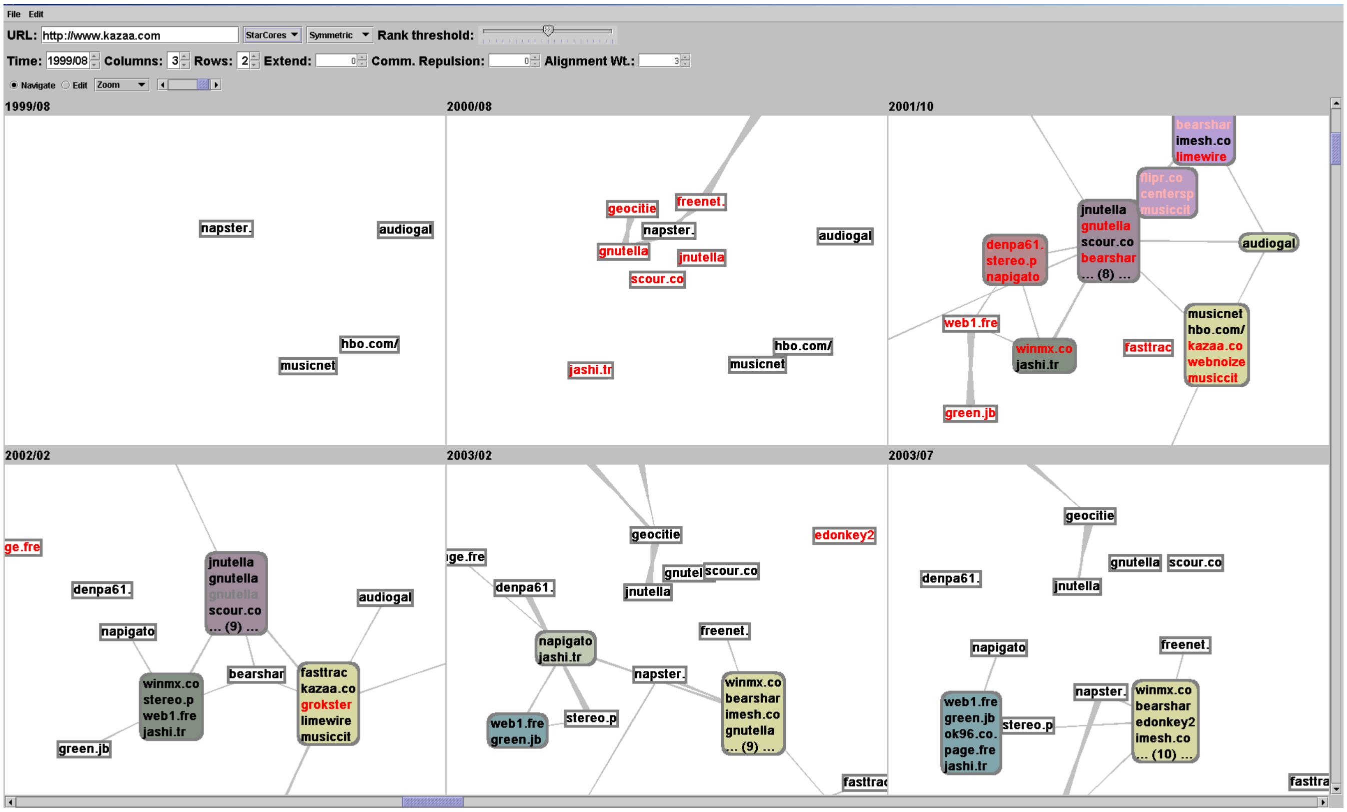
When relationships are results of RPA, the neighborhood is top N pages derived from $u \in F$. Since an appropriate value of N would differ according to the focused topic, the user can change the parameter N. When N becomes smaller, $G_t(F)$ only connects densely connected authorities. Therefore, $G_t(F)$ becomes sparser but reliable. When N becomes larger, $G_t(F)$ also connects sparsely connected authorities, and becomes denser and noisy.

Clustering method

When relationships are hyperlinks, clustering is performed by using server names. That is, pages in the same server are gathered into a single cluster.

When relationships are results of RPA, we extract densely connected pages in $G_t(F)$ as clusters. Although cliques seem to be a good definition of clusters, we have found many meaningful clusters sparser than cliques. Therefore, we use a heuristic definition of clusters.

First, we focus only on mutual connections in $G_t(F)$, in which two nodes derive each other by RPA, and ignore all one way edges. Then we extract dense subgraphs connected by this mutual connections. We use a triangle consists of mutual edges as a unit, and a cluster is defined as a set of the triangles that share edges. Note that a cluster becomes a 1-connected subgraph, and may be a complete graph with four or more nodes. Finally, each node belongs to multiple clusters selects a cluster that has most connectivity, so that



URL: http://www.kazaa.com StarCores Symmetric Rank threshold:

Time: 1999/08 Columns: 3 Rows: 2 Extend: 0 Comm. Repulsion: 0 Alignment Wt.: 3

Figure 3: Evolution of P2P file sharing systems in the cluster view

clusters become disjunctive. This method is based on our previous work. Refer to [18], for more detailed descriptions.

5. VISUALIZATION AND USER INTERACTION

5.1 Basic User Interaction

WebRelievo visualizes a time series of web graphs (TG) like a comic strip as shown in Figure 1. These graphs are aligned from left to right then top to bottom according to their time. The user can change the number of rows and columns, and can also change the time of the first graph to slide graphs through time.

Each $G_t(F)$ is displayed as a directed graph with weighted edges. The direction of each edge is shown by its thickness at each ends. That is, each edge starts at the thick end, and goes to the thin end. Weights of edges are shown by overall thickness of each edges. Positions of nodes are synchronized over time, so that the user can recognize the changes in graphs. For graph layout, we use an automatic and interactive layout algorithm based on a kind of force-directed model [14].

The user can designate a focused page by typing the URL in the input box, which is in the top-left of Figure 1. The focused page is added to F, then $TG(F)$ is extracted and displayed. When the user designates another URL in the input box, the another URL is added to F, and $TG(F)$ is extracted again. The same operation can be done on existing nodes using a pop-up menu.

WebRelievo provides two kinds of views, a difference view and a cluster view. The difference view mainly shows appearance and disappearance of nodes and edges by their colors, so that changes in nodes and edges can be recognized. The cluster view simplify graphs by collapsing nodes into clusters, so that the user can see overview of graphs, and changes in clusters. In the following, we first describe two kinds of views, then explain the synchronized graph layout algorithm.

5.2 Difference View

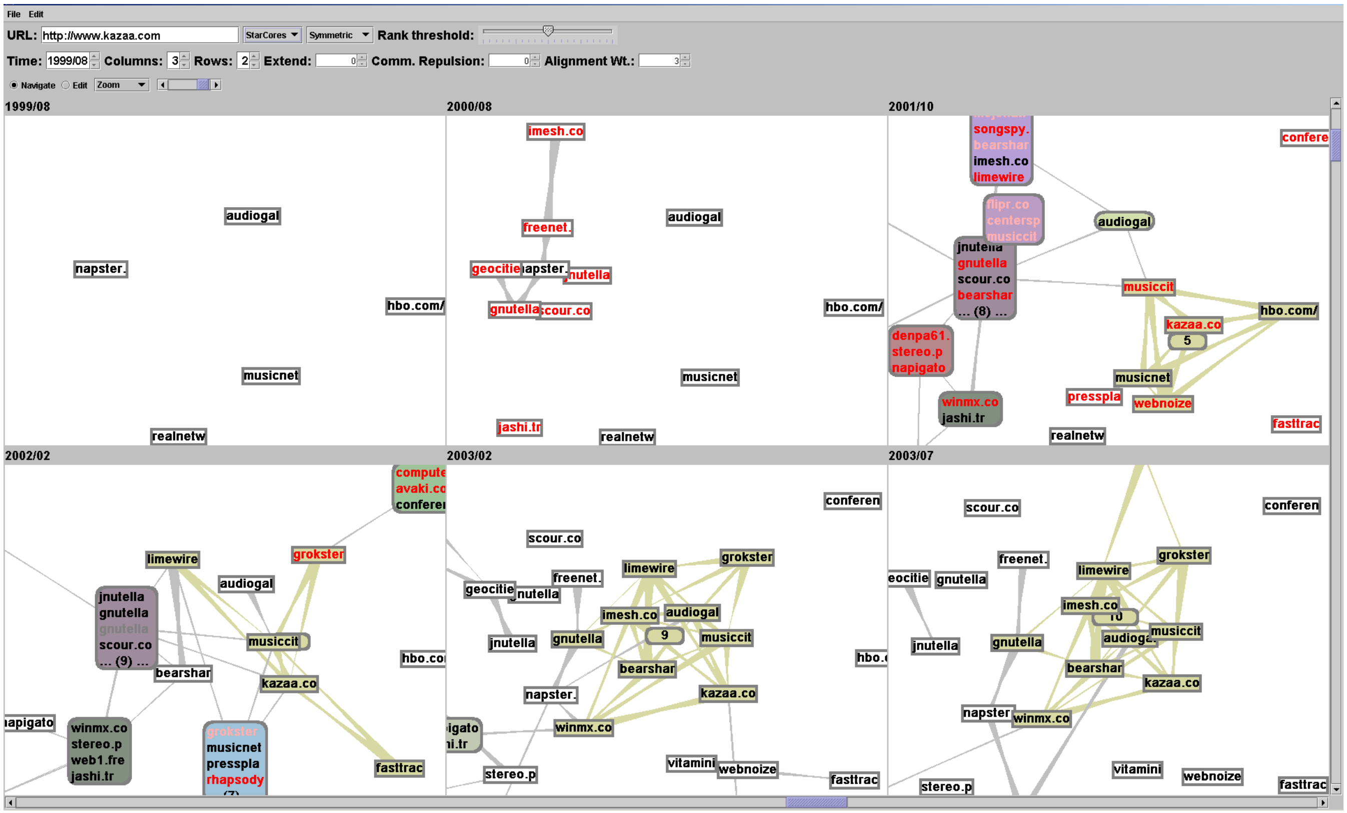
In the difference view, nodes and edges are colored by their types of changes. For example, if a node or an edge appeared at time t, we use a red color for it. In this way, the user can see what kinds of changes have occurred and will occur on each node and edge.

Each $G_t(F)$ is compared with the previous one $G_{t-1}(F)$, and the next one $G_{t+1}(F)$, then changes in each node (or edge) are classified into four types and colored as follows:

Stay–Stay When the node (or edge) exists in both $G_{t-1}(F)$ and $G_{t+1}(F)$, it is colored black to show its stability.

Stay–Disappear When the node (or edge) exists in $G_{t-1}(F)$, but does not exist in $G_{t+1}(F)$, it is colored light gray to show its stay from $t-1$ and disappearance at $t+1$.

Appear–Stay When the node (or edge) does not exist in $G_{t-1}(F)$, and exists in $G_{t+1}(F)$, it is colored red to show its appearance and stay to $t+1$.



URL: http://www.kazaa.com StarCores Symmetric Rank threshold:

Time: 1999/08 Columns: 3 Rows: 2 Extend: 0 Comm. Repulsion: 0 Alignment Wt.: 3

● Navigate ○ Edit Zoom

Figure 4: More detailed view of the P2P evolution

Appear–Disappear When the node (or edge) exists neither in $G_{t-1}(F)$ and in $G_{t+1}(F)$, it is colored light red to show its volatility.

5.3 Cluster View

In the cluster view, nodes in the same cluster is collapsed into a single cluster node to show the overview of graphs. Figure 3 shows the cluster view of Figure 1. In this view, graphs can be displayed in relatively small screen space.

Each round rectangle represents a cluster including two or more pages, in which URLs are drawn in the same color in the difference view. When the number of URLs in the cluster is too large to display, that number is drawn at the last line in the cluster. Note that clusters with a single node are displayed as normal nodes. In addition, the corresponding clusters have the same background color over time, so that the user can track changes in clusters.

Each cluster can be expanded, when the user wants to see the details of a cluster. In Figure 4, a cluster in July 2003 is expanded. The round rectangle in the center of expanded nodes shows the number of nodes in the cluster. When the user expands a cluster, all the corresponding clusters are expanded. In Figure 4, the corresponding clusters in February 2003 and 2002 are expanded.

5.4 Synchronized Graph Layout

We modify the force-directed model [14] by adding the feature to synchronize multiple graphs with clusters. The force-directed model considers a graph as a physical system, in which attractive forces F_a are exerted on all pairs of connected nodes, and repulsive forces F_r are exerted on all pairs of nodes. First, it randomly determines positions of nodes. Then it moves nodes according to those forces, and find a stable layout in which all forces are balanced.

In WebRelievo, F_a and F_r is defined as a function of the distance d between two nodes as follows:

$$F_a(d) = d^2/c_1^2, \quad F_r(d) = -c_1/d$$

Where c_1 is a constant representing the ideal length between nodes, which can be modified by the user.

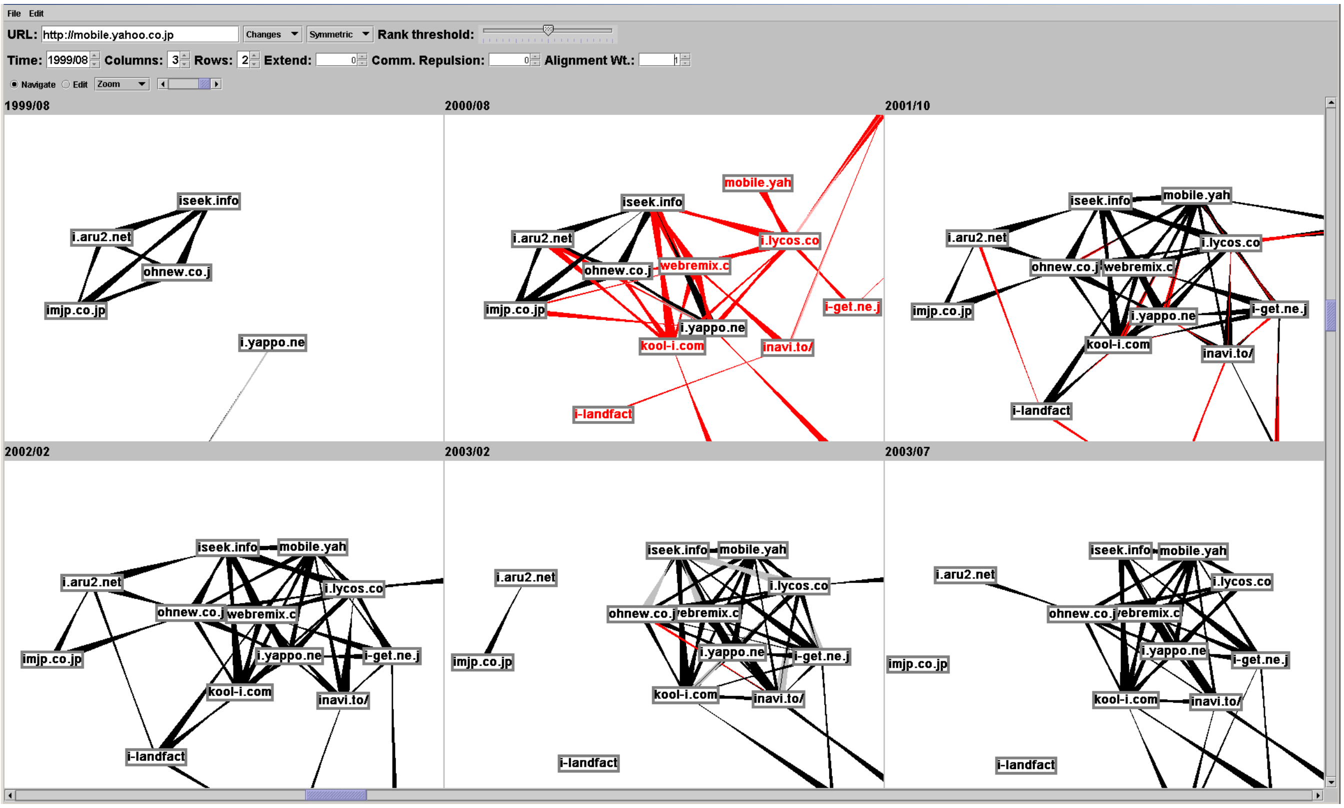
The synchronization of node positions is performed between neighboring graphs. In the difference view, each node in each $G_t(F)$ is simply attracted to the position of the same node in the previous and next graph. In the cluster view, clusters may merge and split over time, and share URLs with multiple clusters in the next or previous time. Therefore some extra forces are required to show changes in clusters.

Synchronizing the difference view

In the difference view, two forces F_{t-1} and F_{t+1} are exerted on a node $u_t \in V_t(F)$ to attract u_t to positions of $u_{t-1} \in V_{t-1}(F)$ and $u_{t+1} \in V_{t+1}(F)$, respectively. (u_{t-1} and u_{t+1} are nodes that have the same URL with u_t.) F_{t-1} is defined as functions of the distance d_{t-1} between u_t and u_{t-1}, and F_{t+1} is defined in the same way as follows:

$$F_{t-1}(d_{t-1}) = d_{t-1}/c_2, \quad F_{t+1}(d_{t+1}) = d_{t+1}/c_2.$$

Where c_2 is a constant greater than or equal to 2, representing strictness of synchronization. When $c_2 = 2$, each node



URL: http://mobile.yahoo.co.jp

Time: 1999/08 Columns: 3 Rows: 2 Extend: 0 Comm. Repulsion: 0 Alignment Wt.: 1

● Navigate ○ Edit Zoom

Figure 5: Evolution of search engines for mobile phone internet services

is positioned almost at the same place over time. When c_2 becomes greater than 1, the strictness of synchronization is weakened. This parameter can be also modified by the user.

Synchronizing the cluster view

To keep track of clusters in the cluster view, we first determine main lines of clusters (i.e. sequences of the corresponding clusters), and synchronize their positions in each main line. Then, clusters not in main lines are arranged according to their merging and splitting behavior. For example, two clusters are attracted when they are merged at the next time.

For each cluster C_t^k, we define the corresponding cluster C_{t+1}^k at time $t+1$ as the cluster that shares the most URLs with C_t^k. If there were multiple clusters that share the same number of URLs, we select a community that has the largest number of URLs. We can reversely identify the cluster at time t corresponding to C_{t+1}^k. When this corresponding cluster is just C_t^k, we call the sequence $(C_t^k,\ C_{t+1}^k)$ as the main line of C_t^k. The main line is recursively extended over time. On a cluster C_t^k in a main line, F_{t-1} and F_{t+1} are exerted as same as in the difference view.

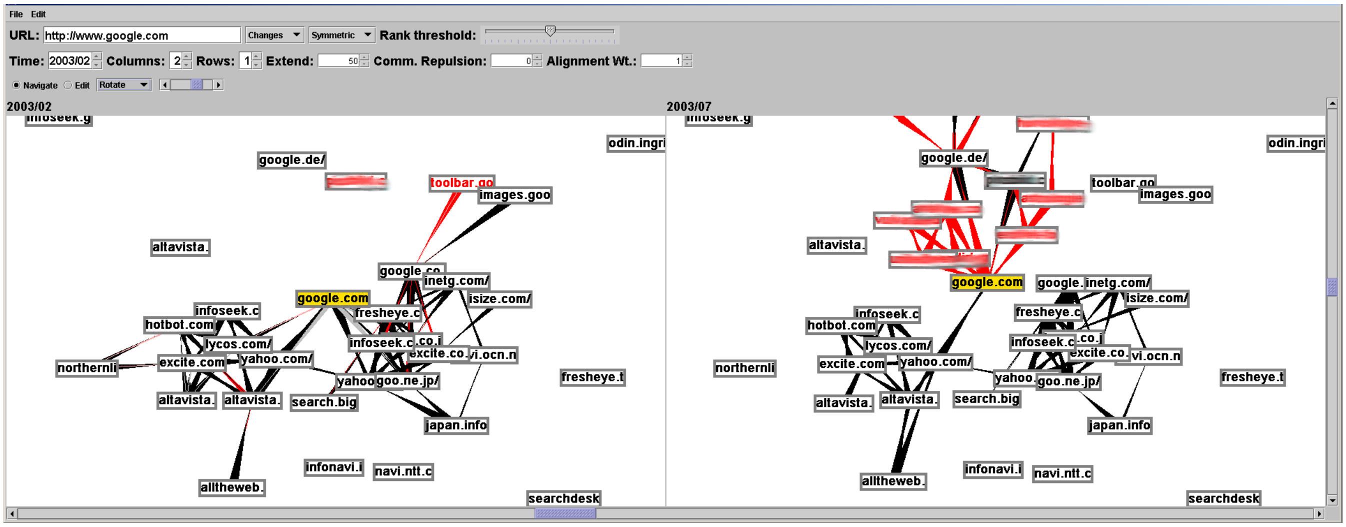
There are many clusters that are not in main lines, and are merged into or split from main lines. Such clusters are attracted to related main lines whether they are connected or not. In this way, we can show that these clusters will be merged at the next time, or have split from the previous time. For example, in Figure 3, we can see that P2P systems, such as Napster and Gnutella, are merged into a cluster at 2001, and they are located near to the cluster at 2000.

C_t^i is merged into a main line $(C_t^k,\ C_{t+1}^k)$, when $C_t^i \neq C_t^k$ and $C_t^i \cap C_{t+1}^k \neq \emptyset$. In this case, C_t^i is attracted to the main line. That is, the attractive force F_a (the same force on connected nodes) is exerted on C_t^i and C_t^k. When there are multiple main lines in which C_t^i is involved. C_t^i is attracted to each main lines. Similarly, C_{t+1}^i is split from a main line $(C_t^k,\ C_{t+1}^k)$, when $C_{t+1}^i \neq C_{t+1}^k$ and $C_{t+1}^i \cap C_t^k \neq \emptyset$. In this case, F_a is exerted on C_{t+1}^i and C_{t+1}^k.

Initially, nodes are randomly located in each panel, then each nodes are iteratively moved by those forces. The layout is fixed, when the total movement of nodes become less than a threshold. This iterative layout is shown by animation in WebRelievo.

The user can scroll and zoom into graphs. These kinds of changes in a graph are immediately propagated to all graphs for keeping layouts synchronized. Nodes in all graphs can be moved by dragging. When the user drags a node in a graph, the same node is moved in each graph. Layouts of all graphs are re-calculated and animated by the force-directed model simultaneously, so that the user can keep track of the evolution after those operations.

In our current implementation, WebRelievo can handle several hundred nodes in each graphs on a PC with a 2.8 GHz Pentium 4. In the cluster view, it means that we can see relationships between a few thousands of URLs. To display such large graphs, it requires a high resolution screen. We currently run our system on a high resolution wall display with 5120x2304 pixels. Note that screen snapshots in this paper are taken on a screen with 1900x1200 pixels, and sizes of fonts are bigger than usual for readability in the paper.

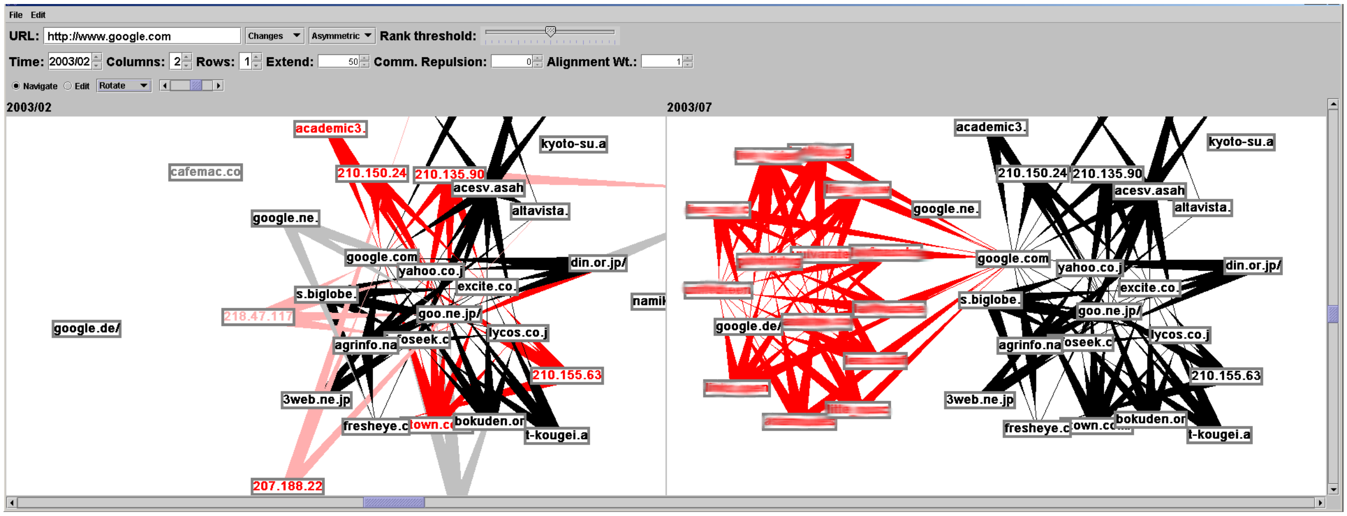

File Edit

URL: http://www.google.com Changes Symmetric Rank threshold:

Time: 2003/02 Columns: 2 Rows: 1 Extend: 50 Comm. Repulsion: 0 Alignment Wt.: 1

● Navigate ○ Edit Rotate

Figure 6: Link spamming around google.com (Using results of RPA as relationships)


File Edit

URL: http://www.google.com Changes Asymmetric Rank threshold:

Time: 2003/02 Columns: 2 Rows: 1 Extend: 50 Comm. Repulsion: 0 Alignment Wt.: 1

● Navigate ○ Edit Rotate

Figure 7: Link spamming around google.com (Showing hyperlinks between hubs and authorities)

6. EVOLUTION EXAMPLES

In this section, we show how the evolution of the Web can be investigated using WebRelievo with three examples.

Evolution of P2P file sharing systems

The first example in Figure 1, 3, and 4 shows the evolution of P2P file sharing systems. In this case, the user wants to know the history and trend of P2P systems, and designates the URL of a famous P2P system Kazaa (`www.kazaa.com`). In Figure 1, we use results of RPA as relationships in *TG*. We can see that the first P2P system Napster appeared at 1999. Then around Napster, various systems appeared such as Gnutella (2000), WinMX (2001), and Kazaa (2001). Then, famous systems became densely connected, and formed an almost clique.

Since it is still complicated to see the changes in recent years, we use a cluster view in Figure 3. From 2001 to 2002, The clustering results were unstable, in which small clusters appeared and merged as P2P systems were rapidly growing. We can see that Napster had lost its influence and was pushed out from the cluster on February 2003.

To investigate more detailed changes in this cluster, we expanded the cluster on July 2003 in Figure 4. The all corresponding clusters are expanded, and we can see the inside of the cluster. From this result, we can also see that Gnutella (the second P2P system) was also loosing connections to its cluster, while new systems such as Kazaa was increasing connections.

Evolution of search engines for i-mode

The second example shows the evolution of search engines for mobile phone internet services (mainly for the i-mode service of NTT DoCoMo). In this case, the user wants to know how the trend of those search engines has changed, and designate a famous search engine (`mobile.yahoo.co.jp`) as a focused page. We use results of the RPA as relationships, and only show mutual connections in the difference view. In Figure 5, we can clearly see the changes in the trend of i-mode search engines. In 1999, search engines for i-mode were mainly provided by small companies, and they formed a 4-clique. In 2000, major companies such as Yahoo! and Lycos began to provide i-mode search engines, and gradually the clique moved to these major companies. In 2003, the clique in 1999 disappeared.

Investigating link spamming

WebRelievo can be used to investigate the structure of link spamming by providing suspicious pages, such as pornographic pages highly ranked in search engines. The final example shows link spamming around `www.google.com`. In Figure 6, we first use results of RPA as relationships in TG, and find that many pornographic sites (nodes with blurred labels for inappropriate keywords) suddenly appeared around `www.google.com` in July 2003.

To investigate the reason of this phenomenon, we change relationships in TG to hyperlinks, and use hubs and authorities as neighborhood of `www.google.com`. The result is shown in Figure 7. In February 2003, search engines such as Google and Yahooare pointed to by appropriate hubs related to search engines. However, in July 2003, `www.google.com` is pointed to by bursty appeared pornographic hubs (nodes with blurred labels). Those hubs are pointing to many pornographic sites and Google together. As a result, they achieved higher PageRanks and authority scores. Now, we can eliminate such spam pages to correct ranking.

7. CONCLUSION

We have proposed the WebRelievo system for visualizing and analyzing the evolution of the web structure based on a web archive including a series of web snapshots. This system visualizes the evolution with a time series of graphs, in which nodes are web pages, and edges are relationships between pages. WebRelievo aligns these graphs according to their time, and automatically determines their layout keeping positions of nodes synchronized over time, so that the user can recognize the changes in graphs. WebRelievo differs from previous work in visualizing evolving graphs of clusters based on their merging and splitting behavior.

WebRelievo allows us to answer historical questions, and to investigate changes in topics on the Web, from appearance and disappearance of pages on a focused topic, and evolution of their relationships. We have shown that WebRelievo can be used for tracking trends in the Web, such as evolution of P2P softwares and i-mode search engines.

Investigating link spamming is now important issue, since the way of spamming has become more sophisticated as search engines avoid link spamming. As shown in the final example, WebRelievo can be also used for investigating link spamming structure.

WebRelievo can show detailed changes in relationships of pages. Though we support clustering of nodes, it is still difficult to see the global and more frequent changes in the Web. We plan to support hierarchical clustering of nodes to show more global view, and to crawl the web more frequently, and visualize more continuous evolution of the web.

8. REFERENCES

[1] K. Bharat, A. Broder, M. Henzinger, P. Kumar, and S. Venkatasubramanian. The Connectivity Server: fast access to linkage information on the Web. In *Proceedings of the 7th International World Wide Web Conference*, pages 14–18, 1998.

[2] S. Brin and L. Page. The Anatomy of a Large-scale Hypertextual Web Search Engine. In *Proceedings of the 7th International World Wide Web Conference*, pages 107–117, 1998.

[3] C. Chen and L. Carr. Visualizing the evolution of a subject domain: A case study. In D. Ebert, M. Gross, and B. Hamann, editors, *IEEE Visualization '99*, pages 449–452, San Francisco, 1999.

[4] C. Chen and S. Morris. Visualizing Evolving Networks: Minimum Spanning Trees versus Pathfinder Networks. In *IEEE Visualization 2003*, pages 67–74, 2003.

[5] E. H. Chi and S. K. Card. Sensemaking of evolving web sites using visualization spreadsheets. In *IEEE Symposium on Information Visualization (INFOVIS '99)*, pages 18–25, 1999.

[6] E. H. Chi, J. Pitkow, J. D. Mackinlay, P. Pirolli, R. Gossweiler, and S. K. Card. Visualizing the Evolution of Web Ecologies. In *Proceedings of ACM SIGCHI '98*, pages 400–407, 1998.

[7] C. Collberg, S. Kobourov, J. Nagra, J. Pitts, and K. Wampler. A system for graph-based visualization of the evolution of software. In *ACM Symposium on Software Visualization (SoftVis)*, pages 77–86, 2003.

[8] J. Dean and M. R. Henzinger. Finding related pages in the World Wide Web. In *Proceedings of the 8th World-Wide Web Conference*, pages 389–401, 1999.

[9] S. Diehl and C. Görg. Graphs, They are Changing. In *The 10th Symposium on Graph Drawing*, pages 23–30, 2002.

[10] S. Diehl, C. Görg, and A. Kerren. Preserving the Mental Map using Foresighted Layout. In *Proceedings of Joint Eurographics – IEEE TCVG Symposium on Visualization, VisSym 2001*. Springer Verlag, 2001.

[11] C. Erten, P. Harding, S. G. Kobourov, K. Wampler, and G. Yee. Exploring the Computing Literature Using Temporal Graph Visualization. In *Proceedings of SPIE, Visualization and Data Analysis 2004*, pages 45–56, 2004.

[12] C. Erten, S. G. Kobourov, V. Le, and A. Navabi. Simultaneous Graph Drawing: Layout Algorithms and Visualization Schemes. In *The 11th Symposium on Graph Drawing*, pages 437–449, 2003.

[13] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient Identification of Web Communities. In *Proceedings of KDD 2000*, pages 150–160, 2000.

[14] T. M. J. Fruchterman and E. M. Reingold. Graph drawing by force-directed placement. *Software - Practice and Experience*, 21(11):1129–1164, 1991.

[15] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, 1998.

[16] R. Lempel and S. Moran. The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect. In *Proceedings of the 9th World-Wide Web Conference*, pages 387–401, 2000.

[17] A. Shapiro. Touchgraph. http://www.touchgraph.com/.

[18] M. Toyoda and M. Kitsuregawa. Creating a Web Community Chart for Navigating Related Communities. In *Conference Proceedings of Hypertext 2001*, pages 103–112, 2001.

[19] M. Toyoda and M. Kitsuregawa. Extracting evolution of web communities from a series of web archives. In *Proceedings of the Fourteenth Conference on Hypertext and Hypermedia (Hypertext 03)*, pages 28–37, August 2003.