Finding and Visualizing Inter-site Clan Graphs

Loren Terveen and Will Hill
AT&T Labs - Research
180 Park Avenue, P.O. Box 971
Florham Park, NJ 07932-0971 USA
+1 973 360 {8343, 8342}
{terveen, willhill}@research.att.com

ABSTRACT
For many purposes, the Web page is too small a unit of interaction. Users often want to interact with larger-scale entities, particularly collections of topically related items. We report three innovations that address this user need.

- We replaced the web page with the web *site* as the basic unit of interaction and analysis.
- We defined a new information structure, the *clan graph*, that groups together sets of related sites.
- We invented a new graph visualization, the *auditorium visualization*, that reveals important structural and content properties of sites within a clan graph.

We have discovered interesting information that can be extracted from the structure of a clan graph. We can identify structurally important sites with many incoming or outgoing links. Links between sites serve important functions: they often identify "front door" pages of sites, sometimes identify especially significant pages within a site, and occasionally contain informative anchor text.

KEYWORDS
Social filtering, collaborative filtering, information access, information retrieval, information visualization, human-computer interaction, computer supported cooperative work, social network analysis, co-citation analysis.

INTRODUCTION
Web search and navigation are two difficult problems that have received much attention, with search engines and indices like Yahoo being the most widespread solution attempts. However, users have larger and longer term information needs, in particular, how to manage lasting interest in a broad topic and to comprehend collections of multimedia documents pertaining to the topic.

CHI 98 Los Angeles CA USA

Our goal is to address these user needs. We replaced the Web page with the *site* — a structured collection of pages, a multimedia document — as the basic unit of interaction. A site is more appropriate for several reasons. (1) A site usually contains a coherent body of content on a given topic (e.g., song lyrics, episode guides for a TV show, current weather conditions), divided into pages to ease navigation and download time. Thus, users want to know what's available at a given site, not a single page. (2) Most hyperlinks *to* a site point to the "front door" page, while most links *from* a site come from the index page. Thus to analyze inter-site structure appropriately (which is our goal), we must correctly group pages into sites.

Second, we defined a new information structure, the *clan graph*, to represent collections of densely connected sites. The clan graph has clear intuitive motivation based on concepts from social network analysis, social filtering, and co-citation analysis. A clan graph is defined in terms of a user specified set of seed (example) sites and is constructed by following hypertext links from the seeds. It is easy for users to specify seeds, e.g., they may get them from their bookmarks file, from an index page they found on the web, or from a search engine. And the clan graph construction algorithm is tolerant of "noise" in the seeds: a few off-topic seeds will not affect the quality of the graph.

Third, to enable users to comprehend and manage the information we extract, we have developed the *auditorium visualization*, which communicates key information such as whether a site is structurally central or peripheral, whether a site is more of a content provider or index, important internal structure of a site, and how sites link together. Figure 4 (which we discuss in a later section) shows an example auditorium visualization.

Our system is implemented in Java. We have built and analyzed clan graphs for dozens of topics, performed some experiments to evaluate our algorithms, and iterated our interface design significantly in response to user feedback.

RELATED WORK: EXTRACTING AND VISUALIZING HIGH LEVEL STRUCTURES FROM THE WEB
Our work is most closely related to research that aims to raise the level of abstraction at which users interact with the

Web. Researchers have sought to define useful, higher-level structures that can be extracted from hypertext collections, such as "collections" [19], "localities" [17], "patches" or "books"[3]. This approach opens up four major avenues of innovation: definitions of new structures, algorithms to extract the structures, visualization techniques that enable users to comprehend the structures, and interface techniques that create a workspace in which it is easy to specify, modify, and experiment with the structures. We survey some leading projects in this area, then compare and contrast our approach.

Kleinberg [11] defines algorithms that identify *authoritative* and *hub* pages within a hypertext. Authorities and hubs are mutually dependent: a good authority is a page that is linked to by many hubs, and a good hub is one that links to many authorities. An equilibrium algorithm is used to identify hubs and authorities in a hypertext collection. For both Kleinberg and WebQuery [4], a collection consists of the results of a search query augmented with all pages that link to or are linked to by any page in the original set of results. WebQuery sorts pages into equivalence classes based on their total degree (number of other pages in the collection they are connected with), and displays the pages in a "bullseye" layout, a series of concentric circles each containing pages of equal degree. WebCutter [14] builds a collection of URLs based on text similarity metrics, then presents the results in tree, star, and fisheye views. twURL [22] organizes URLs into outlines based on properties such as server, domain, and number of incoming links.

Pitkow and Pirolli [19] report cluster algorithms based on co-citation analysis[7]. The intuition is that if two documents, say A and B, are both cited by a third document, this is evidence that A and B are related. The more often a pair of documents is co-cited, the stronger the relationship. They applied two algorithms to Georgia Tech's Graphic Visualization and Usability Center web site and were able to identify interesting clusters.

Card, Robertson, and York [3] describe the WebBook, which uses a book metaphor to group a collection of related web pages for viewing and interaction, and the WebForager, an interface that lets users view and manage multiple WebBooks. They also present a set of automatic methods for generating collections (WebBooks) of related pages, such as recursively following all relative links from a specified web page, following all (absolute) links from a page one level, extracting "book-like" structures by following "next" and "previous", and grouping pages returned from a search query.

Pirolli, Pitkow, and Rao [17] defined a set of functional roles that web pages can play, such as "head" (roughly the "front door" of a group of related pages), "index", and "content". They then developed an algorithm that used hyperlink structure, text similarity, and user access data to categorize pages into the various roles. They applied these algorithms to the Xerox web site and were able to categorize pages with good accuracy.

Mackinlay, Rao, and Card [13] developed a novel interface for accessing articles from a citation database. The central UI object is a "Butterfly", which represents one article, its references, and its citers. The interface makes it easy for users to browse from one article to a related one, group articles, and generate queries to retrieve articles that stand in a particular relationship to the current article.

Mukherjea et al [16] and Botafogo et al [2] report on algorithms for analyzing arbitrary networks, splitting them into structures (such as "pre-trees" or hierarchies) that are easier for users to visualize and navigate.

Other efforts propose novel ways to view and navigate information structures. The Navigational View Builder [15] combines structural and content analysis to support four viewing strategies: binding, clustering, filtering and hierarchization. Through the extensive use of single user operations on multiple windows, the Elastic Windows browser [10] provides efficient overview and sense of current location in information structures. Lamping et al [12] explored hyperbolic tree visualization of information structures. Furnas [6] presents a theory of how to create structures that are easy for users to navigate.

Somewhat less directly related are the SenseMaker [1] and Scatter/Gather [18] systems. SenseMaker supports users in the contextual evolution of their interest in a topic. The focus is on making it easy for users to view and manage the results of a query and to create new queries based on the existing context. Scatter/Gather supports the browsing of large collections of text, allowing users to iteratively reveal topic structure and locate desirable documents.

There are some similarities between these research efforts and ours. We are experimenting with a purely structural analysis, like [2, 4, 13, 19], although we concentrate on links between sites, not pages. We are interested in the functional roles a web page can play, like [11, 17]. As in [3], seed sites in our system serve as "growth sites" that form the basis for a particular type of "related reference query" [1] that retrieves a structure of related sites. Finally, like [3] we are interested in citations between documents.

Our work also has important differences. Most significantly, we must induce both the basic units, the sites, and the collections into which they are structured. Previous efforts either took the collection as a given (e.g., all the web pages rooted at a particular URL like www.xerox.com), offered methods for supporting users in creating collections, or defined the collection as an augmentation of the results of a search engine query. Card et al [3] do offer some automated techniques for creating collections, but the basic unit out of which their collections are built is a single web page. Thus, the resulting collections are more local than our clan graphs; in particular, some of them are more or less a single site. Through the use of multiple seed sites, our

system benefits from a kind of "triangulation" effect when identifying new sites of interest. Another important difference is that the web consists of many ecologies of dynamic, evolving documents. Thus, mutual concurrent citation is possible, even normative, unlike with paper articles where lengthy publishing cycles makes it rare. (Note, however that if journals rather than articles are taken as the units for co-citation analysis, then by-year concurrent citation also is possible [7]). The clan graph is a new structure that generalizes the co-citation relationship, takes mutual citation and transitivity of citation into account, and draws on social filtering insights [5, 8, 21].

CLAN GRAPHS: CONCEPTS AND ALGORITHMS

A clan graph is a directed graph, where nodes represent content objects (such as documents) and edges represent a citation of or reference to the contents of the target node by the source node. Before we can describe how we construct and visualize clan graphs, we define our terms precisely.

Terminology

Universal Graph — the graph of all inter-document (e.g., inter-site) links in the information structure.

Topic Graph — A subgraph of the universal graph that contains sites on the same or similar topics. This is an ideal construct that can only be approximated, e.g., through analysis of structure or similarity of content.

Local Clan Graph — For a specified set of seed sites, this is the subgraph of the universal graph whose nodes are the seed sites or are "closely connected" to the seeds.

Observed Clan Graph — It is practically impossible to construct the entire local clan graph because:

- the web is huge: trying to fetch all the pages on a site and to follow all the links off a site takes a long time;
- the web is unreliable: some sites always are down.
- the web is constantly changing, so the universal and local graphs are moving targets.

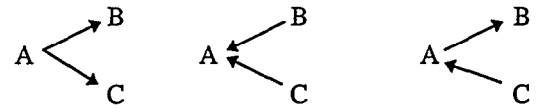
Thus, the observed graph is the subgraph of the local graph that we observe when we attempt to construct the graph.

Local clan graph: a formal definition

Our goal is to find the local clan graph for a set of seed sites. Precisely what does it mean to be "closely connected" to the seeds in the local clan graph? We experimented with several definitions, but converged on a simple, appealing definition building on concepts from social network analysis [9, 20], co-citation analysis [7], and social filtering [5, 8, 21]:

- the NK local clan graph for a seed set S is {(v,e) | v is in an N-clan with at least K members of S}.

An N-clan[20] is a graph where (1) every node is connected to every other node by a path of length N or less, and (2) all of the connecting paths only go through nodes in the clan. We are interested primarily in 2-clans, that is, the 2K local clan graph. The clan graph is a key construct for us; we



a) co-citation b) social filtering c) transitivity

Figure 1: Motivating the use of 2-Clans

believe it productively formalizes notions like "collection" and "locality". We now attempt to justify this belief.

Why 2-clans? Figure 1 graphically depicts three types of inter-document relationships. In each case, an interesting relationship between two of the documents can be inferred based on a known relationship between the other two. Co-citation analysis (1a) says that documents B and C are related if A cites them both. Social filtering (1b) says that if documents B and C both refer to A, then B and C may be link to similar sorts of items in general, and thus deal with similar topics. Figure 1c shows a limited (2-edge) transitivity; we could interpret this as "if C is on a particular topic, and cites A, then A is likely to be on topic; and if A cites B, then B is likely (though somewhat less so) to be on the same topic."

These three relationships are the minimal 2-clans. They show why 2-clans are appropriate to represent topically related subgraphs of items in a larger graph. 2-clans are necessary because no smaller structures allow us to make inferences about document relatedness, and they are sufficient because no larger structure enables other simple inferences.

Notice that 2-clans are defined over undirected graphs: in other words, we take A and B as connected whether A links to B or vice versa. Again, we think this is appropriate. We have observed many sites that are topically central but that have only in-links (content sites) or out-links (index sites). A measure that required bi-directional paths between nodes would underrate some important sites. Therefore we first establish connectivity; subsequent analysis takes directionality into account in order to identify important structures like sinks and sources.

Finally, the 2-clan definition avoids the use of an arbitrary graph density parameter (density is the proportion of actual links in a graph to the maximum possible number of links): is a graph dense enough if its density is 0.2? 0.4? 0.5? ... ?

Why K seeds? By requiring that sites be related to a certain number of seeds, we ensure that we find not just dense graphs, but graphs in which a certain number of the seeds participate. Since we assume that the seeds (at least most of them) deal with a specific topic, this is a way to stay on topic. This is how we operationalize the "triangulation" effect we mentioned earlier. And making K larger is a simple way to get smaller, more tightly connected graphs. This usually means that the sites in the graph are more likely to be both on-topic and central to the topic. Of

course, the optimal value of K depends in part on the number of seeds. For example, one almost certainly would want to pick a larger value of K for a seed set of size 20 than for one of size 5.

Constructing the observed clan graph

We designed a heuristic algorithm for constructing the observed graph. The algorithm is not guaranteed to produce the complete NK local clan graph; however, what it does produce always is a subgraph of the NK clan graph. In the next subsections, we discuss the role of the seed sites, sketch the algorithm and describe the scoring function for site selection, and discuss how sites are defined.

Input: the seed

The observed graph we obtain depends on the properties of the seed sites we start with. Our experience is that users are able to choose good seeds. Good seeds have three properties. First, the seed set must cohere: if the seed sites have few outgoing links or link to few of the same sites, the observed clan graph will be small or even empty (i.e., there is no N-clan that contains at least K seeds). This implies that the seeds do not participate in a significant dense subgraph within the universal graph. Second, the seeds must cover the topic: a poorly chosen seed set may lead to an observed graph that is a small subset of the topic subgraph. This can be the case if there are too few seeds, or the seeds are not well distributed across components in the topic graph. Finally, the seeds must be accurate: if some of the seeds are off-topic, then the clan graph may contain off-topic sites. However, if most of the seeds are on-topic, this is not a problem in practice. The parameter K plays an important role here: because any site added to the graph must be in a 2-clan with at least K seeds, as long as fewer than K off-topic sites are themselves related, sites they link to will not make it above this threshold.

The algorithm

We needed a type of web crawler, which fetches html pages, follows (some of the) links found on the pages and induces sites from pages. Sites that are linked-to are stored on a queue and become candidates for expansion (fetching and analysis). The major decision the algorithm must make is which sites from the queue to expand. Here is a sketch of the algorithm:

queue ← seed sites
while there is a queue element with a score above threshold **do**
  get the highest scored site from the queue
  expand this site
  add the expanded site to the observed graph
  merge new sites and links from the expanded site into the queue
  re-organize and re-score the sites on the queue
end

Scoring sites on the queue

We need a scoring metric that estimates the likelihood that a site on the queue is in the local graph with the seed sites, i.e., that it is in a 2-clan with at least K seeds. The metric must be efficient to compute, since it must be applied to each site on the queue, and the queue typically contains hundreds or thousands of sites.

We currently use the following scoring metric:

- score of site S = the number of seed sites that are linked to S by paths of length 2 or less.

This metric is cheap to compute. It also is a reasonable heuristic, since 2-clans are composed of 1 and 2-paths. Thus, if a site has a score of (say) 5, then it already is known to be in a 2-clan with 5 seeds. We are in the process of experimenting with and evaluating this heuristic and considering other heuristics at different points along the accuracy/efficiency continuum.

Sites

A site (multimedia document) is an organized collection of pages on a specific topic maintained by a single person or group. Sites have structure, with pages that play certain roles (front-door, table-of-contents, index). A site is not the same thing as a domain: for example, thousands of sites are hosted on www.geocities.com. And what counts as a site may be context dependent. For example, if one is taking a survey of research labs, www.media.mit.edu might well be considered a site, while if one is investigating social filtering projects, individual researchers' sites hosted on www.media.mit.edu are probably the proper units.

The last observation suggested a way to operationalize the definition of a site that suits our needs. When building a clan graph, the relevant known context is the set of URLs that have been linked to by the expanded sites. The intuition is that if sites in the clan graph link to two URLs, one of which is in a directory that contains the other, then they are likely to be from the same site[1]. More precisely:

- if url A has been linked to and url A/B has been linked to, then assume that A is the root page of the site and that A/B is an internal url.

This rule applies recursively, so the urls A/B/C, A/B, and A would be merged into a site with root page A and internal pages A/B and A/B/C.

This rule can fail — two URLs that belong to the same site will not be merged if no common ancestor in the directory structure (the "real" root page) has been linked to, and two URLs from distinct sites can be merged, (e.g., if there are links to two distinct sites hosted on www.geocities.com and to www.geocities.com itself). We are refining this rule with

[1] Notice that our notions of site and clan graph are interdependent: a site is defined in terms of links from within the graph, and the graph is constructed by following links from sites.

site-splitting heuristics based on the idea that when some "internal" pages are linked to significantly more often than is the (supposed) root page, then the heavily linked-to internal pages may be separate sites. And we are considering site-merging heuristics based on the idea that if (supposedly) distinct sites point to many of the same pages in the same domain, they may be part of the same site.

We also must decide whether a link from a page is within the site or to another site. We classify links based on their relationship to the root page of the site. If a link is contained within the directory that contains the root page, then we classify it as internal; otherwise, we classify it as a link to an external site. Internal links are added to a site-internal queue of candidate pages to be fetched.

Finally, we must specify how many pages to fetch from a site, i.e., what it means to expand the site,. The primary reason for fetching pages is to find links to other sites, which are the building blocks of the clan graph. For this purpose, finding a site's index page presumably would yield most or all such links, so we could stop expanding the site then. Indeed, we try to find index pages first by sorting pages on the site-internal queue by name, preferring pages whose names contain words like "links", "pages", "sites", "web", and "internet".

However, there is another reason to fetch pages, namely to build a profile that can be used to evaluate a site. Factors like site size (in pages) and amount of content (text, images, audio files) are important. The more pages we fetch, the more accurate a site profile we can create. Therefore, to serve both goals, we introduce a parameter P (default = 25) that controls how many pages to fetch from a site.

ANALYZING CLAN GRAPHS

After constructing a clan graph, we analyze it to extract additional structure to aid user comprehension. We first compute structural properties of sites; for each site, we tally the number of 2-clans it is a member of and the number of in and out links. Combining this structural information with site profile data like size (in pages) and the amount and type (text, audio files, images) of content makes it possible to distinguish "official" sites (for a TV show, for example), which tend to have lots of content and in-links and few or no out-links, from index sites, which tend to have little content and lots of out-links. Thus, users don't get stuck following links from one index site to another, never getting to the content that they really want.

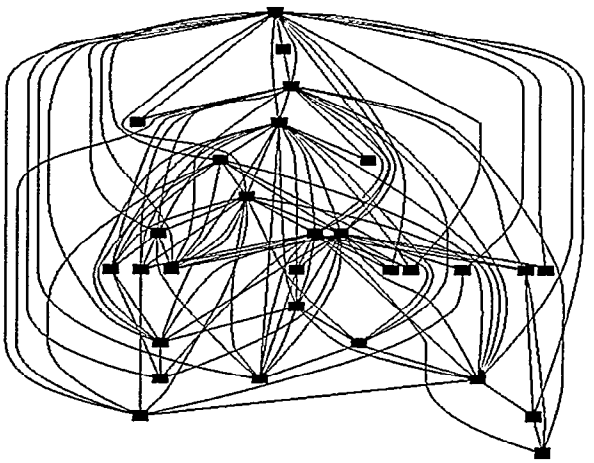
We also identify internal pages of a site that multiple external sites have linked to. By providing direct access to these pages, we create "shortcuts" to places the topic community found worth endorsing. These can be considerable aids to navigation.

Finally, we analyze the text associated with the hyperlinks to each site. Much of the time the text is either the site title or a close variation. However, sometimes it is a useful alternative description of what the site is good for. We are experimenting with techniques to identify useful descriptions and use them in the interface.

VISUALIZING CLAN GRAPHS

The structure of clan graphs that we have observed in the web is complicated and not easy to visualize or understand. For example, figure 2 is a direct node/edge representation of the clan graph for the Television show "Roar" observed in August of 1997. The drawing was produced by a sophisticated graph layout tool, *dot*, which minimizes edge crossings, yet the drawing still is complicated. The clutter of edge crossings, edge angles and local node constellations divert visual attention to non-significant graphic elements. A viewer can identify some nodes of high and low degree, but the layout reveals no overall pattern. It is virtually impossible to visually discern central and peripheral sites.

Figure 2: Graph view with least edge crossings



For the purpose of revealing node degree, simply collapsing the graph structure into a list of nodes ordered by degree is a better interface. The ordered list form of figure 3 makes it easy for users to compare node degree and check quantities. Note that the eighth site in the list ("Universal Studios") contains substructure, i.e., an internal page that was linked to by multiple sites. The list view is quick and easy to produce but still hides many important properties of sites and the graph. The list view is linear, so it easily communicates only one dimension. It is textual, so it cannot exploit graphical display properties, either images from the sites or the use of color, position, shape, etc. to communicate site properties. It is static, so there is no dynamic focusing, no hiding and revealing of structure.

We wanted users to see the results of our clan graph analysis in terms of the graph itself. These results include site centrality/peripherality, in-link to out-link ratio, patterns of inter-site links, and how sites rank in terms of properties such as size, number of images, audio and download files.

Figure 3: HTML view of clan graph

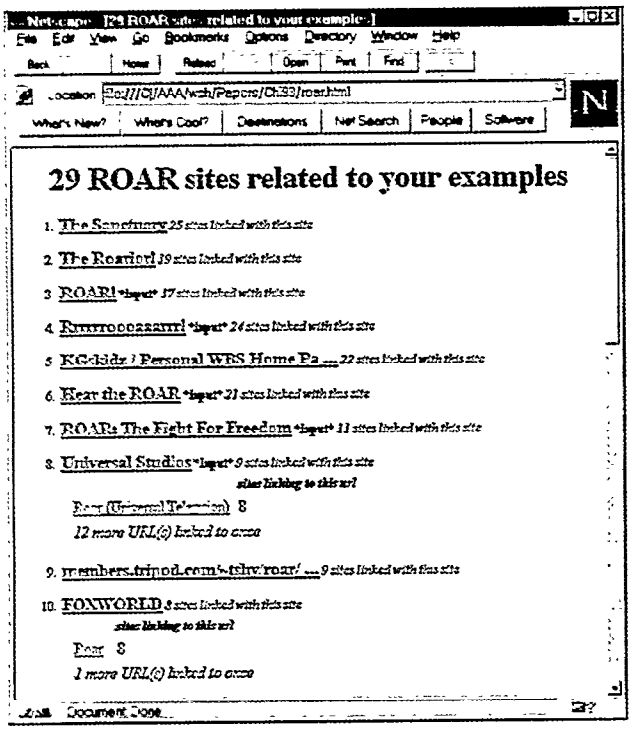
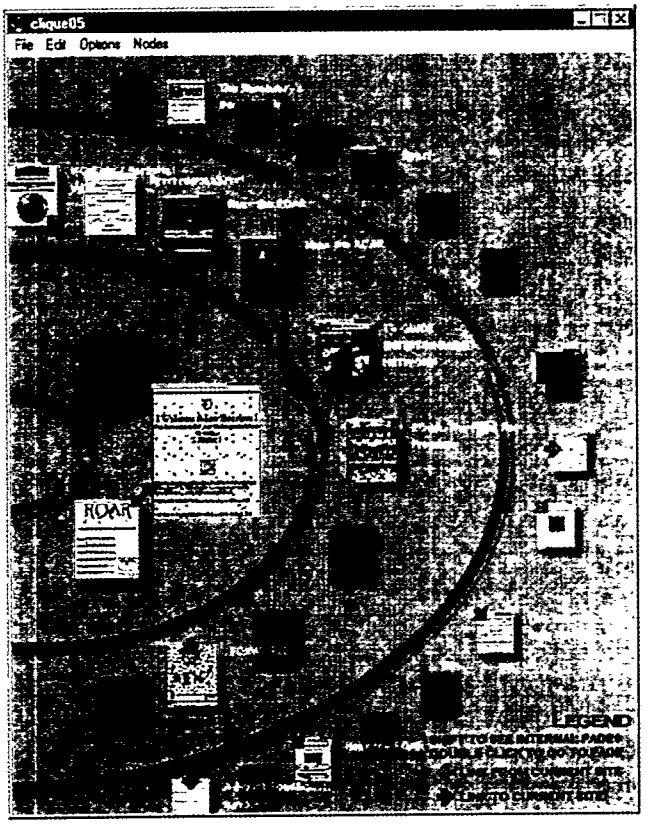


Figure 4: Auditorium view of clan graph



The Auditorium view: how it satisfies our design goals

To meet these goals, we have iterated cycles of design and usability testing, arriving at a design that we call the *auditorium seating visualization.* The metaphor is to the arrangement of seating in an auditorium: row upon row curved around a center stage. Figure 4 and color plate 1 show the auditorium seating visualization of the clan graph for the television show "Roar!". Thumbnails of site "front door" pages serve as iconic representations of sites. The auditorium seating visualization is dynamic. By moving the mouse over a site thumbnail, users switch from a general view of the graph to a view focused around the indicated site. Figure 4 shows the visualization in the site-focused mode. The thumbnail of the focused site is enlarged, and green "in" arrows and red "out" arrows appear on sites that the focused site is linked with. Other unlinked sites are blanked, but their drop shadows are left to note their positions. As the result of user experience with many design versions, we came to use a large number of techniques to communicate information necessary to satisfy user needs. We now discuss these in detail. Please refer to figure 4 or color plate 1 to identify the graphic elements discussed.

- *Concentric semi-circles* of sites group sites into equivalence classes from most to least important on some user-settable property. By default, sites are assigned to rows based on the number of in-links, so the closer to the center a site is, the more of its "peer sites" have linked to it.
- *Dynamic ordering within semi-circles.* Our original design used concentric circles instead of semi-circles. However, user feedback showed the desirability of ordering sites within each row, and while circles "wrap around", the properties important for ordering (such as number of in or out links and amount of content) do not. Semi-circles, on the other hand, with their definite end points, were suitable for our purpose.

  An important distinction users made was between index sites and content sites. Allowing dynamic ordering of sites within a row by properties like number and proportion of in and out links, and amount of content (audio files, images, or all types of content) makes these distinctions apparent. By default, we order sites within rows by the amount of content, so sites with lots of content appear at the top of each row.
- *Graded colored bands* aid in interpreting the values of the within-row ordering property of sites. Bands are graded from bright red to bright green, with the color break occurring at the median value of the ordering property. For example, if sites are ordered by the proportion of in links to out links, the break point is a visual cutoff between sites that serve more as indices and sites that serve more as content repositories.
- *Hiding graph spaghetti* — We wanted to reveal the fine structure of inter-site links without producing visual spaghetti as in figure 2. Users typically focused

either on all the links from or to a single site or traced the edge between two sites. We designed to support those two visual tasks while removing as many distracting visual elements as possible. We did this with "one-site at a time" dynamic presentation of graph structure. Users move the mouse cursor over a site to focus on it, and only links from or to the focused site are shown. To further reduce clutter, we do not draw complete links between sites, since they draw too much user attention to uninformative crossings and edge angles. Instead, we represent links with small in and out arrows.

- *Linked views* The auditorium view is linked to a web browser; clicking on a thumbnail drives the browser to that web site.
- *Progressive revelation* of greater detail. While a site is in focus, holding down the shift key reveals any internal pages of that site that are linked to by other sites. These are pages that the author of the linking site found worthy of special attention. The link text often is more informative in these cases
- *Thumbnail representations* reveal quite a bit of information about sites. Overall design and color scheme can be seen. Ratio of text to graphics on the front door page tells users something about what to expect from a site. Saturated color, positioning and shape of banner ads reveal their presence in thumbnails. If a user has browsed a site previously, a thumbnail usually is sufficient to identify the site.

Early user testing highlighted for us the necessity of relevance feedback, leading to construction of a new observed clan graph. Users can judge sites as on-topic (good) or off-topic (bad). On-topic sites are added to the original seed set, and off-topic sites are added to a stop list. Thus, users can nudge the graph into a somewhat different area, moving it closer to the ideal topic they have in mind.

FUTURE WORK: EVALUATION AND DEPLOYMENT

We must verify experimentally that the NK local clan graph is a useful construct. A graph should contain mostly on-topic sites, and the quality of the graph should not be too dependent on precisely which sites are selected as seeds. Our informal inspections of dozens of graphs show these conditions to be satisfied, but clearly we need more systematic evaluation. To that end, we did a pilot study on the topic of the rock group The Grateful Dead. We used 63 URLs obtained from Yahoo as a starting point for our experiment. We randomly divided these URLs into sets of size 5, 10, and 20. We used these as seed sets for our clan graph construction algorithm, also experimenting with different values of K. Analysis of the results so far has confirmed some of our intuitions. First, larger seed sets (size 10 or 20) tend to result in graphs that better cover a topic than do smaller seed sets (size 5). Since the Web contains many index pages, it is easy to obtain a sufficient number of seeds on many topics. Second, increasing the parameter K results in smaller, more tightly focused graphs, while decreasing K leads to larger, but perhaps not as accurate graphs. Third, sites with large numbers of in-links almost always are discovered by the clan construction algorithm regardless of the sites in the seed set. Therefore, the algorithm does not appear overly sensitive to the choice of seeds. Finally, when we ranked sites within a graph by in-degree, the top ranked sites (i.e., those most cited by their "peers") always were on-topic. We did find that "the topic" may be somewhat broader than we initially had supposed. For example, many Grateful Dead sites link to The Electronic Frontier Foundation and various tape-trading and tape-tracking sites. Although these sites are not about the Grateful Dead per se, clearly they are part of what the online Grateful Dead community considers important and relevant. This community is defined by but not limited to interest in The Grateful Dead. We are continuing our evaluation work, both analyzing additional topics and quantifying the tentative conclusions we have drawn so far.

We are extending the interface to give users more control during the graph construction process, allowing them to intervene early if they find some sites particular interesting (or not), thus influencing subsequent sites that are added to the graph. We also are considering methods to scale our visualization. Currently, it can handle around 35-40 sites. We would like to scale it up to at least 100 sites and are confident that techniques like fisheye views and zooming will get us there. Finally, one of our colleagues, Brian Amento, is preparing to carry out formal user studies of the auditorium visualization and a dynamic text-table interface to the same data (i.e., a clan graph). We are seeking experimental evidence of the utility of the clan graph information structure and the relative utility and usability of the auditorium visualization and the best dynamic textual interface we can design.

Finally, we are making our system robust enough for widespread use. We will first open it up for use within our laboratory. After any fixes and enhancements this leads to, we intend to distribute the system freely, thus enabling anyone to create collections of online documents on topics they are interested in. We will put up a server where people can publish and retrieve collections. It is our hypothesis that relatively few people will choose to build collections, but many will want to view and interact with collections someone else has built. By distributing our software and maintaining a server, we will be able to test this hypothesis, and, in general, to investigate the social nature and social roles of communities that organize their interests around online information resources.

CONCLUSIONS

The goal of the work reported here is to help people find and manage collections of documents related to topics they care about. We offer a novel information structure, the clan graph, to formalize the notion of a topically related

collection of interlinked documents. We present an algorithm to construct a clan graph from a set of seed documents. The algorithm also tackles the hard problem "what is an online document?": it aggregates individual web pages (URLs) into sites (multimedia documents) based on the context of links from other documents. Finally, we introduce and illustrate the auditorium visualization. It gives a graphical overview of the most important several dozen sites for a topic, lets users explore structural relationships between sites and the internal structure of individual sites, and allows dynamic sorting to aid users in understanding the structural role a site plays within the community of related sites. We are moving from informal to formal evaluations of both our algorithms and interface and are making our implementation robust enough to be freely distributed and used.

ACKNOWLEDGMENTS

We thank Brian Amento, Josh Creter and Peter Ju for their system implementation work. We also thank Harley Manning, Steve Whittaker, Lynn Cherny, and Julia Hirschberg for many useful design discussions.

REFERENCES

1. Baldonado, M.Q.W., and Winograd, T. An Information-Exploration Interface Supporting the Contextual Evolution of a User's Interests, in *Proceedings of CHI'97* (Atlanta GA, March 1997), ACM Press, 11-18.
2. Botafogo, R.A., Rivlin, E., and Shneiderman, B. Structural Analysis of Hypertexts: Identifying Hierarchies and Useful Metrics. *ACM Transactions on Information Systems 10, 2*, 142-180.
3. Card, S.K., Robertson, G.C., and York, W. The WebBook and the Web Forager: An Information Workspace for the World-Wide Web, in *Proceedings of CHI'96* (Vancouver BC, April 1996), ACM Press, 111-117.
4. Carrière, J., and Kazman R. WebQuery: Searching and Visualizing the Web through Connectivity, in *Proceedings of WWW6* (Santa Clara CA, April 1997).
5. *Communications of the ACM,* Special issue on Recommender Systems, 40, 3 (March 1997). Resnick, P., and Varian, H.R., guest editors.
6. Furnas, G.W. Effective View Navigation, in *Proceedings of CHI'97* (Atlanta GA, March 1997), ACM Press, 367-374.
7. Garfield, E. *Citation Indexing.* ISI Press, Philadelphia, PA, 1979.
8. Hill, W.C., Stead, L., Rosenstein, M. and Furnas, G. Recommending and Evaluating Choices in a Virtual Community of Use, in *Proceedings of CHI'95* (Denver CO, May 1995), ACM Press, 194-201.
9. Jackson, M.H. Assessing the Structure of Communication on the World Wide Web. *Journal of Computer-Mediated Communication , 3, 1*, June 1997
10. Kandogan, E., and Shneiderman, B. Elastic Windows: A Hierarchical Multi-Window World-Wide Web Browser, in *Proceedings of UIST'97* (forthcoming), preprint at http://www.cs.umd.edu/users/kandogan/papers/uist97/paper.html
11. Kleinberg, J.M. Authoritative Sources in a Hyperlinked Environment, in *Proceedings of 1998 ACM-SIAM Symposium on Discrete Algorithms* (forthcoming).
12. Lamping, J., Rao, R., and Pirolli, P. A Focus + Context Technique Based on Hyperbolic Geometry for Visualizing Large Hierarchies, in *Proceedings of CHI'95* (Denver CO, May 1995), ACM Press, 401-408.
13. Mackinlay, J.D., Rao, R., and Card, S.K. An Organic User Interface for Searching Citation Links, in *Proceedings of CHI'95* (Denver CO, May 1995), ACM Press, 67-73.
14. Maarek Y.S., Jacovi, M., Shtalhaim, M., Ur, S., Zernik, D., and Ben Shaul, I.Z. WebCutter: A System for Dynamic and Tailorable Site Mapping, in *Proceedings of WWW6* (Santa Clara CA, April 1997).
15. Mukherjea, S., and Foley, J. D. Visualizing the World-Wide Web with the navigational view finder. *Computer Networks and ISDN Systems 27, 1*, (1995), 1075-1087.
16. Mukherjea, S., Foley, J.D., and Hudson, S. Visualizing Complex Hypermedia Networks through Multiple Hierarchical Views, in *Proceedings of CHI'95* (Denver CO, May 1995), ACM Press, 331-337.
17. Pirolli, P., Pitkow, J., and Rao, R. Silk from a Sow's Ear: Extracting Usable Structures from the Web, in *Proceedings of CHI'96* (Vancouver BC, April 1996), ACM Press, 118-125.
18. Pirolli, P., Schank, P., Hearst, M., and Diehl, Scatter/Gather Browsing Communicates the Topic Structure of a Very Large Text Collection, in *Proceedings of CHI'96* (Vancouver BC, April 1996), ACM Press, 213-220.
19. Pitkow, J., and Pirolli, P. Life, Death, and Lawfulness on the Electronic Frontier, in *Proceedings of CHI'97* (Atlanta GA, March 1997), ACM Press, 383-390.
20. Scott, J. *Social Network Analysis: A Handbook.* SAGE Publications, London, 1991.
21. Terveen, L.G., Hill, W.C., Amento, B., McDonald, D., and Creter, J. Building Task-Specific Interfaces to High Volume Conversational Data, in *Proceedings of CHI'97* (Atlanta GA, March 1997), ACM Press, 226-233.
22. *What is twUrl?* http://www.roir.com/whatis.htm