

Data Visualization, Indexing and Mining Engine  
– A Parallel Computing Architecture for Information Processing Over  
the Internet

Xiannong Meng, Zhixiang Chen, Richard H. Fowler, Richard K. Fox, Wendy A. Lawrence-Fowler  
Department of Computer Science  
The University of Texas - Pan American  
1201 W. University Drive  
Edinburg, TX 78539-2999

February 1998

# Contents

<b>1</b>	<b>Introduction</b>	<b>4</b>
<b>2</b>	<b>A Parallel Computing Architecture for Information Processing Over the Internet</b>	<b>5</b>
2.1	An Information Service Model . . . . .	5
2.2	System Architecture . . . . .	5
<b>3</b>	<b>Document Explorer – Information Retrieval and Visualization Tools for the WWW</b>	<b>7</b>
3.1	Extracting And Organizing WWW Semantic Content . . . . .	8
3.1.1	Deriving the Term Associations . . . . .	8
3.1.2	Pathfinder Networks: A Technique For Network Abstraction . . . . .	9
3.2	Spatial Representation Of the System’s Networks . . . . .	10
3.3	Display And Interaction Mechanisms . . . . .	11
3.3.1	Overviews, orientation, and network abstraction . . . . .	11
3.3.2	Other Navigation And Orientation Tools . . . . .	12
3.3.3	Head-tracked Stereoscopic Display . . . . .	12
<b>4</b>	<b>Web Access of Geographic Information System Data</b>	<b>13</b>
4.1	Functions Provided by GIS2WEB . . . . .	13
4.2	Structure . . . . .	14
<b>5</b>	<b>ParaCrawler — Parallel Web Searching With Machine Learning</b>	<b>15</b>
5.1	A Machine Learning Approach Towards Web Searching . . . . .	15
<b>6</b>	<b>DUSIE — Interactive Content-based Web Structuring</b>	<b>16</b>
6.1	Creating the Index . . . . .	17
6.2	A Query Tool for Information Retrieval . . . . .	18
<b>7</b>	<b>Conclusion And Future Work</b>	<b>19</b>

## List of Figures

1	System Architecture of DaVIME . . . . .	6
2	Software Structure of GIS2WEB . . . . .	14

## Abstract

We propose in this report DaVIME (Data Visualization, Indexing and Mining Engine), a software architecture that performs data visualization, indexing and mining in a integrated environment. DaVIME presents to the users a unified view of information service. When a user issues an information service request, DaVIME calls upon appropriate software components (agents) to provide the service requested. Depending on the type of the service requested, DaVIME may call one or more agents into action to satisfy the user request. DaVIME is an open, extensible architecture that allows researchers and developers to add software components (agents) incrementally. Currently DaVIME includes the following components. Document Explorer analyzes text information in large document collections and presents to the user a suite of visualization tools. DUSIE (Dynamic User-created Searchable Index Engine) extends the hierarchical indexing schemes currently used in popular browsers to let users build content-based searchable indexes. ParaCrawler is a parallel Web search engine which uses novice ranking and indexing algorithm to provide users with more accurate search results in shorter time. Gis2web allows users to access GIS data from the Web.

## 1 Introduction

The Internet provides great opportunities, yet poses great challenges. Opportunities range from building a virtual supercomputer from home[68, 16], to accessing digital libraries from classrooms[47] and searching for any piece of knowledge, e.g. the largest prime number[51, 33]. A great many challenges exist, including security, scalability, resource discovery, information extraction, and coordination of parallel computing activities. Research efforts underway to address these problems can be divided into two major categories: computing over the Internet and information engineering over the Internet.

Research efforts in computing over the Internet use the network to access available computing power of a large number of computers. Issues studied include sharing computing power in a heterogeneous environment, distributing tasks to individual computers, coordinating the results, and developing incentive schemes for sharing the excessive resources. The general model is that the clients needing extra computation or storage will access available resources offered by host computers. The key feature here is that the clients do not access data, only the computing power or storage capacity, on the hosts. Examples in this category include Javelin[16] where clients and hosts register with brokers that help coordinate the work, ParaWeb[12] where clients use internet and intranet as a part of their computing infrastructure in a seamless fashion, and ATLAS[6] where hosts actively take work away from overloaded clients.

Information engineering involves search, retrieval, indexing, categorizing, and visualizing various types of information accessed across the Internet. The Internet's massive amount of typically unstructured and often natural language based data requires approaches different than used in most database applications[26]. The challenges are to guide users in formulating queries, to accurately locate relevant information among hosts, to properly index and rank the information, and to present it to users usefully. Harvest[13] is one of the pioneering and successful Internet information access and discovery systems. Harvest servers run on the client and index information in locally stored files. When a service is requested, the server reports the result through a pre-defined network interface. MetaCrawler[64] uses existing Internet search engines. It sends user requests to search engines such as Yahoo, Lycos and others. Results are analysed, filtered, ranked and sent back to users. MLDB[38] is designed for resource and knowledge discovery in global information systems using a multi-layered database approach. The information is generalized and transformed one layer at a time, the lowest layer being the raw data and

the highest layer being the well organized, fully usable information for the user.

Current research either focus on the information engineering side such as search, categorizing, visualizing certain pieces of information, or concentrate on the sharing excessive computing power over the Internet to speed up some specialized computation. While exciting in concepts, feasible implementations are few, usable tools are scarce. This is partially due to the fact that it is extremely complicated to effectively use the excessive computing power and storage capacity over the Internet without sacrificing security, efficient management, and scalable performance. Instead of an all-out attack on all the problems, we concentrate on building infrastructure and tools that can be used in Internet as well as intranet, making more efficient use of locally campus wide available computing power and storage capacity.

## 2 A Parallel Computing Architecture for Information Processing Over the Internet

We view information engineering as various concerted operations on a collection of information. These operations include categorization, indexing, ranking, visualization, annotation, storage, and transmission.

We propose DaVIME (for Data Visualization, Indexing, and Mining Engine), an architecture that can effectively provide fast, usable information service from the Internet. Our goal is to provide users with a seamless Internet-based information service using search, categorization, indexing, and visualization techniques in a cooperative computing environment.

### 2.1 An Information Service Model

DaVIME takes a general information service approach. Users will request information services such as search, indexing, or visualization. DaVIME provides these services in a fast and accurate manner, using its various agents. As with any information service system, it is vitally important that the user formulate proper queries. In practice, a large portion of an application's effort can go into properly formulating the problem (asking the right question) rather than optimizing the algorithmic details of a particular data method[27]. We will use AI and visualization techniques in query refinement. Parallel search schemes are used to retrieve information. Search results are sorted and ranked, and presented to users through visualization tools.

### 2.2 System Architecture

Properly servicing user information requests is at the core of the research. Our design goal is to have an open, extensible architecture that provides a comprehensive information service for users and allows incremental development for researchers. The system is divided into four major components, the User Interface Coordinator (UIC), the Data Resource Coordinator (DRC), the Computing Resource Coordinator (CRC), and various Extensible SoftBots (ESB) (agents). An overview is given in Figure 1.

**UIC** Users request information service through the User Interface Coordinator. This module is responsible for accepting and refining user requests and then sending the requests to appropriate agents

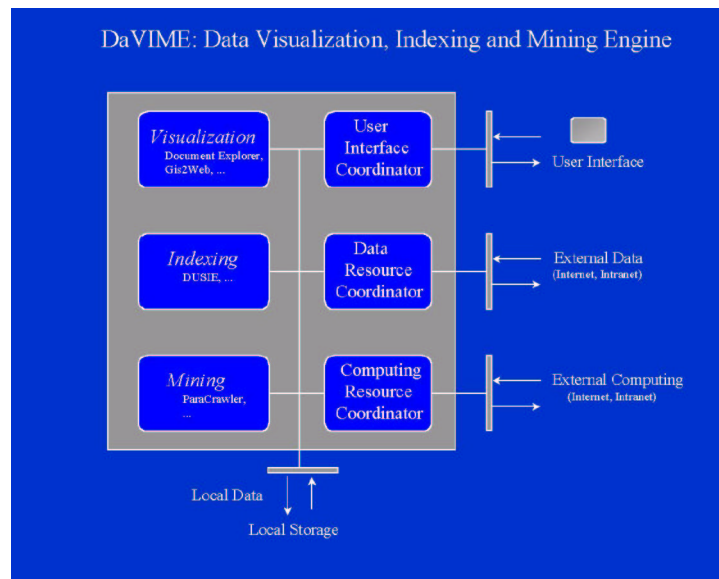


Figure 1: System Architecture of DaVIME

of DaVIME for service.

**DRC** The Data Resource Coordinator handles data exchange among ESB components and provides ESB components with internal and external data. When an ESB component needs data from other component or from Internet, DRC checks the data format and retrieves data for the ESB component from appropriate source through the Internet.

**CRC** The DaVIME architecture allows the ESBs access external computing resources (CPU cycles, storage) from the Internet or intranets. CRC registers resource requests from internal DaVIME components, finds available hosts on the network that are willing to share resources, and requests services from these hosts. In this sense, the CRC is also acting as brokers between DaVIME components and external computing and storage resources.

**ESB** Once the user request is accepted, various agents will act on the requests and provide the user with a visualized response. The ESBs are a collection of various information processing software that can be expanded incrementally. Currently it includes the following components.

**Document Explorer** The Document Explorer[28] analyzes text information in large natural language document collections. Using various text analysis tools, network-based clustering is performed. A suite of visualization tools are available for query formulation and presentation of results.

**Dynamic User Searchable Index Engine** User-configurable indexing techniques[32] are supplied which extend the hierarchical indexing schemes currently in widespread use.

**ParaCrawler** ParaCrawler is a parallel search engine. The search scheme used will be very similar to that of MetaCrawler[64] where the user request is forwarded to various existing search engines. The results returned from these search engines are digested and presented to the user. The indexing and ranking schemes will be different from what the MetaCrawler used.

**Gis2Web** Gis2Web[50] allows users to access GIS (Geographic Information System) data from the Web. Though most information available on the WWW is un-structured, GIS data has a very rigid internal structure because the GIS software that provides these data organizes these information in a data base structure. Gis2Web is to retrieve these information from existing GIS and present to users in Web suitable forms.

In the following sections, we will discuss in details these components of DaVIME.

### **3 Document Explorer – Information Retrieval and Visualization Tools for the WWW**

Many of the challenges of information access on the Internet are issues common to all forms of information retrieval. These issues include difficulties in query formulation such as use of indexing vocabularies, indexing indeterminacy, and users' inability to completely specify information needs[41]. Retrieving information that meets users' information needs is an iterative process, and techniques which explicitly incorporate users' judgments, such as relevance feedback[48], provide means to automate some aspects of user guided retrieval. It is also clear that mechanisms providing alternative paths of access to information can enhance retrieval effectiveness[7].

One promising approach for enhancing information retrieval through the Internet is visualization to facilitate users' perception of document relation structure. A number of systems have been developed to provide visually based browsing mechanisms for traversing the link structure of Internet documents[49, 2, 40, 53], typically in a three dimensional or distorted space. Unfortunately, link connectivity shows relatively little about the topic relations among documents across the Internet. Systems which provide facilities for organizing and viewing document organizations based on semantic content are an alternative. Mukherjea, Foley, and Hudson[52] describe a system which operates on the semantic content of WWW documents to form visually displayed hierarchies. VR-VIBE[11] fixes WWW documents at locations in three-dimensional space by allowing users to interactively position keywords on a pyramid. Gershon et al.[35] describe a system which allows users to view an organization of documents visited as a hierarchy of links and also to construct a separate hierarchy based on their own needs.

Document Explorer[30] is a system for visualization of WWW content structure. Visualization, browsing, and query formulation mechanisms are based on documents' semantic content. These mechanisms complement text and link based search by supplying a visual search and query formulation environment using semantic associations among documents. The user can view and interact with visual representations of WWW document relations to traverse this derived document space. The relationships among individual keywords in the documents are also represented visually to support query formulation by direct manipulation of content words in the document set. Finally, the system provides direct manipulation facilities for forming queries to identify starting points in the network of documents or use conventional vectorbased retrieval mechanisms. A suite of visualization tools for interaction and display are used to supply orientation cues and navigation mechanisms to assist users in browsing the document and keyword networks, as well as in the formulation of queries by direct manipulation of system objects.

### 3.1 Extracting And Organizing WWW Semantic Content

Document Explorer's principal visualizations are network displays based on documents' keyword lists. The lists can be provided by automatic content extraction tools, such as Harvest[13] or derived within the system. Keyword lists for each document are used to determine the associations among documents and among terms. The statistical text analyses rely on recovering conceptual information from natural language by considering the frequency and co-occurrence of words. This basic approach has been used in a wide range of contexts and its utility and limitations are well-known[59].

The visual environment for exploration and direct manipulation uses the same visual representation for query, associative thesaurus, and document content to facilitate query revision by direct manipulation of system objects. The representations underlying the visual displays are minimum cost networks derived from measures of term and document associations. The associations are derived from natural language text for queries, single documents, and associative term thesauri. The document collection is represented as a network of documents based on interdocument similarity.

#### 3.1.1 Deriving the Term Associations

For each set of WWW documents the system uses a separate set of terms formed from the most frequently occurring word stems, excluding function words. For some forms of retrieval this simple procedure suffers from the limitation that frequently occurring terms have relatively little value for discriminating among documents[65]. However, one function of the associative thesaurus is to give a picture of all of the concepts in a document set. The most frequently occurring terms tend to be general terms that provide useful information about the domain of the document collection.

To derive the distances between terms used to construct networks, text is analyzed by first finding the sequence of term stems present in the text. This sequence is used to assign distances between terms based on lexical distance and co-occurrence in syntactic units with a metric similar to that used by Belkin and Kwasnik[10]. Term pair similarity is calculated as the sum of values added when terms are adjacent, or occur in the same sentence, paragraph or document. These similarities provide the associations used in deriving the networks displayed by the system.

There are three reasons for using statistically-based associative structures in an interactive browsing system. One reason follows from the view that information retrieval systems should supply the user with a variety of tools and retrieval techniques. Statistically-based associative information structures provide one class of retrieval tools that can complement other retrieval aids. For example, an associative thesaurus based on term co-occurrence in documents presents a structure of term relationships quite different than presented in a thesaurus showing term hierarchies. The associative thesaurus can encourage browsing and exploration, as well as bring the user's own associations into play. For information needs in which the user is not familiar with the domain, and indeed may not even know what his or her information needs are, the associative structures provide one means to explore and gain information to better define the information need.

A second reason for using statistically-based associative structures is the desire to have a representation that can be derived automatically in an interactive system, rather than through knowledge-engineering efforts such as are required for most deep representations. Associative structures have been used effectively as one component of hybrid systems incorporating both deep and shallow representations[18].

The final reason is the desire to provide a common visual representation for retrieval tools. Networks are naturally represented visually and can provide a common representation for the system's several visualization components.

### 3.1.2 Pathfinder Networks: A Technique For Network Abstraction

The associative networks used in the system are Pathfinder networks (PfNets)[20]. The Pathfinder algorithm was developed to model semantic memory in humans and to provide a paradigm for scaling psychological similarity data[62]. A number of psychological and design studies have compared PfNets with other scaling techniques and found that they provide a useful tool for revealing conceptual structure[17, 61].

The PfNets representations underlying the system's network displays are minimum cost networks derived from measures of term and document associations. The network of documents is based on interdocument similarity, as measured by co-occurrence of keywords between document pairs. For the network of terms, or associative term thesaurus, the visual representation of the user's query, and single document representations the associations are derived from text with association measured by keyword co-occurrence and lexical distance within documents. PfNets can be conceptualized as path length limited minimum cost networks. Algorithms to derive minimum cost spanning trees (MCSTs) have only the constraints that the network is connected and cost, as measured by the sum of link weights, is a minimum. For PfNets, an additional constraint is added. Not only must the graph be connected and minimum cost, but also the longest path length to connect node pairs, as measured by number of links, is less than some criterion. To derive a PfNets direct distances between each pair of nodes are compared with indirect distances, and a direct link between two nodes is included in the PfNet unless the data contain a shorter path satisfying the constraint of maximum path length.

In constructing a PfNets two parameters are incorporated:  $r$  determines path weight according to the Minkowski  $r$ -metric and  $q$  specifies the maximum number of edges considered in finding a minimum cost path between entities. As either parameter is manipulated, edges in a less complex network form a subset of the edges in a more complex network. Thus, the algorithm generates two families of networks, controlled by  $r$  and  $q$ . The least complex network is obtained with  $r = \text{infinity}$  and  $q = n-1$ , where  $n$  is the total number of nodes in the network. The containment property has in practice provided a particularly useful technique for systematically varying network density to provide both relatively sparse networks (the union of MCSTs with  $r = \infty$  and  $q = n - 1$ ) for global navigation, as well as more dense networks for local inspection.

Reducing the complexity of network representations is a central objective in many efforts in visualizing Internet information structures. Complexity can be reduced by transforming a network of document connectivity to a hierarchy by removing links. Compared to the more general problem of representing directed graphs, visualization of hierarchies is relatively well developed. This sort of transformation is used by Mukherjea et al.[10] for WWW documents. In the Document Explorer nodes in the complete network are identified for display and navigation in a fashion similar to the identification of cluster centroids in single link clustering. These document nodes are continuously displayed in the overview diagram and the three-dimensional space within which the user navigates.

Document Explorer uses networks of differing densities to provide separate views of document and term interrelationships. The least dense network, which allows paths of any length in satisfying the minimum cost criterion, is useful for global navigation and orientation. Conceptually, it shows the

strongest relations among elements. As such, it is effective in supplying views for global navigation and structure perception. To provide a more detailed view of relationships the system also maintains a PfNet of elements in which the maximum path length is relatively small, creating a complementary network with many more links. This more dense network display is most beneficial when viewing a small set of elements and serves as a sort of magnifying glass for revealing relations among elements not shown in the sparse network.

### 3.2 Spatial Representation Of the System's Networks

A number of data presentation problems require the drawing or display of graphs, and interest in computer-based visualization has increased attention to methodologies for graph display in three dimensions. For visual presentations graph layout criteria center on how quickly and clearly the meaning of the diagram is conveyed to the viewer, the readability of the graph. Graph drawing algorithms have as their goal the layout and presentation of an inherently mathematical entity in a manner which meets various criteria for human observation. The aesthetics of a layout determine its readability and can be formulated as optimization goals for the drawing algorithm[21]. For example, the display of symmetry and minimization of the number of edge crossings in two-dimensional drawings are fundamental aesthetics for visual presentations.

Considering the wide application of graph structures in display, there are relatively few algorithms for drawing general undirected graphs[66]. This is due in part to the inability to specify the aesthetic criteria individuals use in understanding graphs[22]. Nonetheless, for certain restricted classes of graphs in which graph-theoretic expressions of aesthetic criteria can be specified, satisfactory algorithms have been developed[8, 46, 58]. Some of the aesthetics for drawings of general undirected graphs are symmetry, minimization of edge crossings and bends in edges, uniform edge lengths, and uniform node distribution. These aesthetics are such that optimality of one may prevent optimality in others. Additionally, graph layout algorithms in general can be viewed as optimization problems and are typically NP-complete or NP-hard. These two observations suggest a heuristic approach to general graph drawing for many applications.

The spring embedder algorithm[23] is a heuristic approach to graph drawing based on a physical system. This algorithm simulates a mechanical system in which a graph's edges are replaced by springs and nodes are replaced by rings connecting edges, or springs, incident on a node. From the initial configuration of ring positions, the system oscillates until it stabilizes at a minimum-energy configuration. Among the parameters that control the forces acting on the rings and causing their movement are spring length, spring stiffness, spring type, and initial configuration. This a very general heuristic which can be combined with other algorithms[24] to provide approximate solutions for competing aesthetics.

The spatial representations of the Document Explorer's networks are designed to facilitate users' perception of network structure. Network nodes are positioned in three dimensions using a graph layout algorithm[44] based on the spring metaphor which is similar to Kamada and Kawai's[42] two-dimensional network layout algorithm. As with other spring embedder algorithms, nodes are treated as connectors and spring length and strength among connectors is derived from network link distances. Nodes are allowed to vary in three dimensions and iteratively positioned at the points which minimize energy in the system of springs. Varying spring length and strength allows layouts which are useful for user interaction and visually reveal clustering and connectivity among nodes.

### 3.3 Display And Interaction Mechanisms

One of the central challenges in the display of large information spaces is to overcome users' feelings of disorientation, the feeling of being "lost in hyperspace". In the Document Explorer feelings of disorientation are attenuated in part by the layout of network nodes to facilitate perception of global structure coupled with overview diagrams which track the users current viewing position. Additionally, facilities to interactively vary network density provide global orientation while examining local detail. Finally, navigation tools including bookmarks, anchors, and signposts supply mechanisms allowing users to control the course of navigation and facilitate way finding in the large document and term spaces.

#### 3.3.1 Overviews, orientation, and network abstraction

The size and density of the Internet document network requires viewing and navigation tools that allow users to perceive the overall structure of document relations, explore smaller regions in detail, and select and view individual documents. Display and interaction mechanisms in the Document Explorer supply orientation and overview of the global structure of document associations, together with navigation and retrieval tools for exploring local detail. Overview diagrams[3] display a small number of nodes selected to provide information about the organization of the complete networks. These nodes provide orientation by serving as landmarks to assist the user in knowing what part of the network is currently being viewed. Additionally, the nodes provide entry points for traversing the network.

In the Internet document collections we have used, PfNets derived for associative thesauri and networks of documents have a characteristic structure. There tend to be a small number of nodes that have many nodes directly connected and there are relatively short paths between these highly connected nodes, i.e., there are relatively few high degree nodes and network diameter is small. This network form suggests a criterion for selecting nodes to include in overview diagrams. The system's overview diagrams include those nodes of highest degree in the complete network. The term nodes selected for associative thesauri overviews tend to be general terms that provide a guide to the content of the database, and document nodes selected to appear in overviews tend to be general articles about a content area. The overview nodes are landmarks in that they supply information about both the content and structure of the database. As the user changes viewpoint in the main viewing window, and thus the portion of the network which is viewed in detail, the overview diagram tracks the overview nodes which are visible in the view volume of the detailed view. This helps attenuate disorientation by providing the context for the individual network nodes which are in view.

Other display mechanisms for changing network views are also designed to facilitate orientation to the overall structure while examining local detail. The availability of two separate networks which differ in density, or number of links, provides a form of network abstraction enhancing users' perceptual extraction of network structure, as well as supplying a mechanism for navigation. The most sparse network displayed by the system is essentially a tree, thus having the fewest links necessary to have all nodes connected. It is this sparse network which is used to provide the spring analogs used to position nodes in three-dimensional space. The arrangement of nodes on this basis in practice supplies a characteristic clustering of nodes in which structure is relatively easy to perceive. Yet, much of the utility of browsing in the derived semantic space of documents comes from exploring relations which are relatively weak. These relations are captured in the more dense network. The user typically identifies a single document or document region by browsing or traversing links from a known document in the sparse network, and then selects to have the weaker links of a node or node set become visible. These

links are identified by a differing color and the point at which the selection is made is identified by a marker, or anchor, in the network which is relatively large and can be easily zoomed back to during the course of exploration.

### **3.3.2 Other Navigation And Orientation Tools**

Several navigational tools supply mechanisms to return users to previous viewing points or provide information about the current viewing position's context. Visual bookmarks can be set by the user at any point while moving through the network. The bookmark is displayed as an annotated snapshot of the user's view. Selecting the bookmark returns the user to the viewpoint at that time the bookmark was set. Visual anchors, displayed as colored arrows, can be set at any time and remain visible from long viewing distances. Selecting an anchor also returns the user to the viewpoint at the point the anchor was set.

Signposts provide a technique to supply context information about a network region when it is being viewed in detail. As mentioned above, users can select nodes in the overview windows to navigate to a viewpoint near the selected overview node, and this facility is often used as the initial method of exploring the large network. In practice users quickly become familiar with the relatively small set of overview document labels and terms. Signposts exploit the users familiarity with the overview nodes in a mechanism which provides global context while viewing a detailed local region. Creating a signpost displays a three-dimensional arrow, similar to the visual form of an anchor, but with each point of the arrow labeled with the nearest overview node. The labeled points of the arrow show the direction of the closest overview node. The same orientation information about the location of the detailed view in relation to the complete network is, of course, available to the user by inspecting the overview window to see which portion of the network is being viewed in detail, i.e., discerning the location of the local detail by examining the global context. Yet, the signpost provides a complementary type of information. Information about the global context, the location of the overview nodes, presented within the detailed view.

Additional display and orientation facilities are available. Link and node colorings can be set to indicate parts of the network of high similarity to the user's query. A backtracking facility allows users to retrace the path followed to arrive at a particular viewpoint. Users can select a node which represents a single document or term and expand or collapse connected nodes. Again, color is used to mark the course of browsing and network exploration. The user can enter the associative thesaurus from any node displayed, e.g., from the query or an individual document. Similarly, any document, e.g., from the results of a search, can serve as an entry point into the network of documents. All navigation maintains fluid movement in the space, always zooming, rather than jumping, to new viewpoints.

### **3.3.3 Head-tracked Stereoscopic Display**

We have also explored display and interaction using head-tracked stereoscopic display of the networks. For the sorts of network displays in the Document Explorer there is an increase in users' abilities to perceive structure and perform three-dimensional interaction tasks, as found in a number of recent studies of stereoscopic network viewing, both with and without head position tracking[69, 5]. Recent attention to immersive interfaces has focused on head mounted displays (HMDs). Such displays typically provide the user a wide field of stereoscopic view and use head position tracking to determine view.

Though HMDs can be quite effective in evoking the feeling of presence necessary for total immersion, the apparatus and restricted view preclude their use in most environments.

Head-tracked stereoscopic display (HTS) using LCD shutter glasses affords a compromise in which many of the perceptual and performance advantages of fully immersive interfaces obtain, but the user is free to interact less encumbered by the display apparatus, and information can be displayed with resolution sufficient for text-based tasks. It supplies a relatively immersive interface, yet allows the user to work comfortably at a desk and switch among tasks. In HTS three-dimensional objects appear to be positioned in a volume bounded by the screen perimeter and extending from somewhat in front of the screen to somewhat behind the screen. Much of the utility of three-dimensional display is obtained from the user's active interaction which changes the viewpoint and allows the user to confirm or disconfirm perceptual hypotheses. In HTS simple head movements easily and naturally change viewpoint to "look around" the objects. This viewing paradigm has been called "fish-tank virtual reality" and human factors experiments indicate enhanced perception and interaction in three dimensional tasks for HTS viewing compared to flat screen presentations[5].

The Document Explorer provides stereoscopic viewing together with untethered head position tracking for HTS viewing. The enhanced three-dimensional perception and interaction supplied by this display addresses the needs in our system to aid the user in perceiving structure and interacting with the system's large networks. The system also provides sufficient display resolution and user controlled switching from three to two dimensions for text-based tasks typical of document retrieval. In using the various system displays the user moves between stereoscopic three-dimensional displays for some tasks, e.g., viewing the document collections and term networks, and two-dimensional viewing, such as reading a document abstract, without removing any apparatus.

## 4 Web Access of Geographic Information System Data

Geographic Information Systems (GIS) such as Arc/Info[67] and GRASS[34] usually contain rich and diverse information such as census data, municipal data, business data, highway networks, and geological data. Information is stored in formats including tables, images, and text. Typical GIS software maintains, manipulates, and displays this information in dynamic and graphical ways. The traditional mode of operation is to have the data accessible from within a specific GIS application on a single platform. With the recent wide-spread acceptance and availability of World Wide Web technology, it is natural to look toward building linkages between the two distinct worlds of GIS and the WWW in order to make the wealth of information available in the GIS world publicly available through the WWW. Gis2web[50] is such a system intended to bridge the gap between existing GIS software and the WWW. Gis2web accesses distributed datasets in different Arc/Info formats, converts the data to forms appropriate for WWW distribution, and provides access through a gis2web server. Gis2web allows distributed GIS to dynamically update their databases and have this information available to the WWW through automatic retrieval by gis2web. Users access GIS data interactively through gis2web over the Internet as if they were using a local GIS.

### 4.1 Functions Provided by GIS2WEB

Gis2web provides the following functions.

- Gis2web periodically extracts geographic information from the GIS database and stores the information on local disk. The type and the amount of information to extract is based on the access statistics. Most frequently referenced information will be extracted and stored. Others will be retrieved on demand to save disk space and network traffic.
- Gis2web periodically collects this information from the GIS server and sends it to the WWW server for processing. The GIS server and the WWW server can be at different server, so far as they are connected by the Internet.
- Once received the data sent from the GIS server, gis2web parses the information and converts them to the formats that can be recognized by the WWW server.
- Gis2web communicates with the WWW server to display the information at user's request.

## 4.2 Structure

Gis2web consists of a number of different components. These components act as independent processes. They communicate to each other using Internet protocols. There are five major components. The GISextractor extracts information from the GIS server upon requests. The Messenger accepts raw GIS information sent from the GIS server by the GISextractor. The Parser parses the GIS information into HTML format and stores them on local disk. The Gatekeeper acts as an interface between the gis2web and local WWW server. The Coordinator is the main gis2web which coordinates rest of the functional components. Figure 2 shows the overall structure of gis2web. The following is a more detailed description of each of the components.

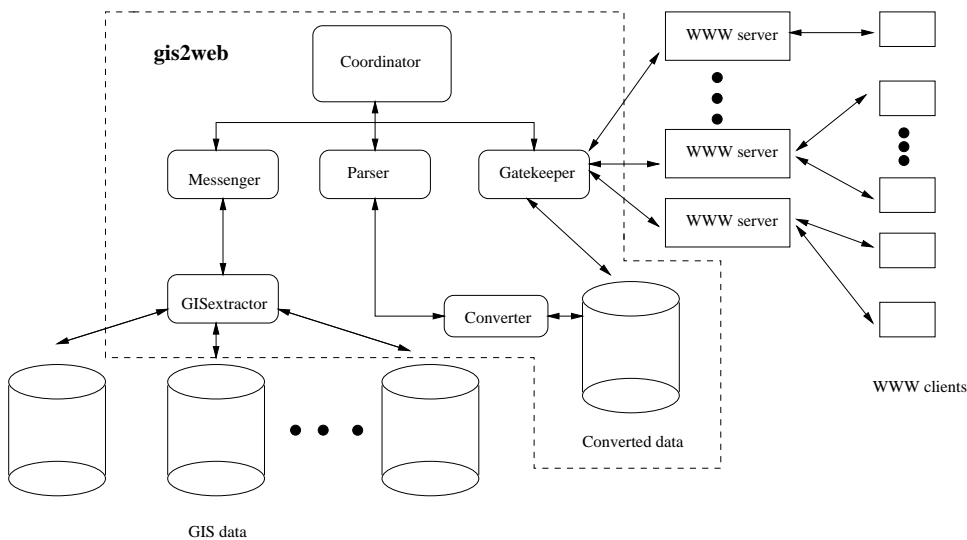


Figure 2: Software Structure of GIS2WEB

**GISextractor:** The *GISextractor* works closely with the GIS software, Arc/View. The purpose of this component is to extract information from the Arc/View database and send it to the main gis2web. Because this component interact with GIS information, it needs to understand the format and structure of the GIS information and it has to be able to communicate with the rest of the

world using standard protocol. For these reasons, the component is written in Arc/View native programming language *Avenue* [25].

**Messenger:** The *Messenger* is responsible for communicating with the GIS server and accepting information sent by the *GISextractor*. The information can be in various formats, texts, tables, or images. The *messenger* simply takes the data from the *GISextractor* as a binary stream along with their headers for data formats) and stores them on local disk in separate files.

**Parser:** The *Parser* converts the GIS information received by the *Messenger* into formats that can be recognized by the WWW server. This includes converting plain text into HTML, converting Arc/View tables into HTML compatible tables, and converting images to GIF or other formats when necessary. The *Parser* also builds necessary hyper-links into the text where needed.

**Gatekeeper:** The *Gatekeeper* is an interface between the local WWW server and gis2web. Essentially, it takes the user request and retrieves the information from the files generated by the *Parser* and display them in proper format on the user screen. This component is in a CGI script format in order to communicate among the components properly.

**Coordinator:** The main program of the gis2web coordinates all the activities among the different components.

## 5 ParaCrawler — Parallel Web Searching With Machine Learning

ParaCrawler is a Web search engine that has combined search, indexing, and ranking capability. We will use a search mechanism similar to that of MetaCrawler[64]. ParaCrawler takes a user request then re-sends it to a number of well-known search services on the Internet in parallel by starting one process for each target search engine. Each of these processes is responsible for working with these pre-defined search engines. Each process will do the following.

1. Each process contacts a pre-specified search engine and sends the user request to it. Because each search engine has different query and response format the individual process has to know the correct formats in order to communicate with these search engines.
2. Once a process establishes a connection with the search engine, it will get search results back from the search engine. The process can receive a large volume of data as result.
3. The process then will use machine learning approaches to locate the most plausible answers and present them to the user.

### 5.1 A Machine Learning Approach Towards Web Searching

In the first generation of text search robots Lycos and others engines extracted keywords using standard algorithms based on word placement, word frequencies and other statistics. These indices did not provide more advanced tools for detecting word sense, synonyms, or word meaning. Despite these drawbacks, users eagerly flocked to use these indices to navigate and sift through a burgeoning growth of the Web documents. Subsequent generations of text search “robots” have been built on this humble foundation[63].

Machine learning approaches (see Knoblock[43]) faced the problem of labeling the samples. Techniques such as uncertainty sampling[45] reduce the amount of labeled data needed, but do not eliminate the labeling problem. Clustering techniques do not require labeled inputs, and have been applied successfully to large collections of documents[15]. Indeed, the Web offers fertile ground for document clustering research. However, because clustering techniques take weaker (unlabeled) inputs than other data mining techniques, they produce weaker (unlabeled) output.

From the machine learning point of view, one needs a sufficiently large unbiased samples of labeled examples to identify a target concept. When we use machine learning theory to attack the problem of the Web searching, a major obstacle is that data is abundant on the Web but unlabeled. To solve the labeling problem, we use the following approach: We allow the user to train a machine learning algorithm to search the Web. That is, when a machine learning algorithm identifies a concept and shows it to the user, the users can label it as a positive example when it is the desired information, otherwise the user can label it as a negative example. After receiving input from the user, the algorithm interactively refines its search and finds a new concept for the user. This process will repeat until the information the user wants has been found.

Our approach is a dialog between the user and the search algorithm (the machine learning algorithm). We allow the user to label a guessed concept for the search algorithm. As we will describe, a target concept (the desired information, or the desired home page) is represented by an axis-parallel rectangle over a discretized vector space, the existing machine learning theory regarding rectangles[4] can be applied to the Web search. However, those machine learning algorithms might not be able to work fast enough for practical use. We need to investigate other solutions. For example, we might be able to allow the user to provide more feedback than just labeling a guessed concept. We might also be able to allow the search algorithm to do parallel searching on all the dimensions at the same time. First, we will consider how to represent documents over the Web. We will use a vector of features to represent a document. For example, an image document can be represented by a vector of image statistics: color, texture, etc. The vector space dimension would be between 50 to 500. Secondly, we will show that any home page (or any image document) can be represented by an axis-parallel rectangle in the above established vector space. Then, we will design machine learning algorithms to identify rectangles in the vector space under different environments. we will show that those algorithms are, in theory, good enough to be implemented as a software “robot” for searching the Web. That is, they are guaranteed to find the desired information and work fast.

## 6 DUSIE — Interactive Content-based Web Structuring

Facilities supporting active engagement by hypertext readers are among the functions characteristic of second-generation hypertext systems, facilities not yet in widespread use in the Web. In describing issues to be addressed by hypertext systems Halasz[37] identifies both dynamic (virtual) structures and extensibility (tailorable) as important components. Users should be able to move through the system according to their needs without spatial or conceptual disorientation. Content-based structuring of hypertext document content was suggested by Bruza[14]. The approach uses a two-level architecture for hypertext documents where the top level “hyperindex” contains index information and the bottom level “hyperbase” contains content nodes and links. The hyperindex consists of a set of indices which are themselves linked together. When an index term describing the required information is found, the objects from the underlying hyperbase are retrieved for examination.

The Dynamic User created Searchable Index Engine (DUSIE)[32] is a user-created dynamic open indexing system, which can connect to other information resources and allows individual annotation, provides the sort of functionality suggested above for augmenting Web structure. The goal of the user-created index is to reduce cognitive overhead by allowing users to define an index using their own terms, based on their understanding. Each index term contains semantic information relevant to the user. The indexing system keeps track of nodes and links to conceptual materials, thus providing a navigational tool through conceptual space. With its additional ability to cross-reference, the index is often more relevant to the user than an index created by the author.

In the simplest terms DUSIE provides users facilities to construct an index for Web documents. Users define a list of index terms, make annotations to the terms, and link related concepts. In the context of the educational setting in which DUSIE was developed, a primary function of the indexing process is to provide a set of terms for each document that reflects the user's conceptual domain and can be used efficiently and effectively to retrieve information from the document. More generally, the system's functionality supplies mechanisms to create 1) new nodes through the annotation facility, 2) links to points in existing Web documents for the new nodes, and 3) alternative links for existing Web nodes through the addition of terms linked to points in existing documents.

DUSIE provides content retrieval in a manner similar to Bruza's two-level architecture for hypertext documents through a top level index information and a bottom level containing content nodes and links. In the DUSIE implementation the hyperindex is a set of indices linked together. When an index term describing the required information is found, the objects from the underlying hypertext document base are retrieved for examination. Retrieving information in this manner is retrieval through navigation. After selecting an index term, the information associated with the hypertext base link is displayed in the browser. The annotation can also be displayed in the browser. Information may also be retrieved using the related concepts entries, facilitating the retrieval of semantically related materials. The user can move directly to the desired information as well as access related information without having to remember locations, links or other structural details. As a reflection of the users conceptual domain, the index terms might be considered metanodes which are linked to other metanodes via terms in the related concepts term set. In this way the index system expresses local relations among clusters of nodes.

## 6.1 Creating the Index

While Web documents use a static and explicit model of hypertext, i.e., nodes, links, and link markers are fully enumerated during creation, DUSIE allows the user to create a virtual structure or model of hypertext through a user defined indexing scheme. The virtual structure, or user defined index, has two levels: the index term level with annotation capabilities and the hypertext base level. The index is made up of a set of index entries. Each index entry consists of a term descriptor or keyword supplied by the user, a locator (like an HTML anchor), an annotation area, and a set of related term descriptors. The term descriptors are as general or as specific as desired by the user. Any particular index term provides a focus for the set of related concept descriptors and so a broadening or refinement of the concept represented by the focus index term. The focus term and the set of related concepts then creates a structure that can be used to support query by navigation. In turn, these indices provide immediate access to required information without navigating through the document space.

When using DUSIE, an index term is first defined by the user, and a new index term node is created

and inserted into the index level. A link is made from the new node to the location in the hypertext document specified by the user. The node is also linked to an annotation block, providing space for the user to make an annotation about the term. After a new index term is defined, the user can define relationships between the index term and other index entries. If a relationship is defined, a link is created from the focus index term node to the related entry node. This creates a set of index entry nodes linked together. The assumption is that the links are created because the user perceives a relationship between the index terms represented by the nodes. As the users perception of the concept materials change, the user can modify the index term network by adding or deleting links between nodes. These user defined relationships among entries serve somewhat as a user defined thesaurus of index terms, even though the terms are not distinguished as synonymous or subordinate, rather just as related concepts. This facility allows the set of index entry nodes to be used later in navigation through the hypertext, as well as for more explicit retrieval of information. The user can make an annotation for each term entry in the index. The annotation might contain explanatory notes, materials to extend the original document content, or even a part of the actual document text. The annotation can be modified at any time as users knowledge or perceptions change. Though the indexing scheme incorporates a two level model, the implementation of DUSIE uses a three level architecture: information source (database), index application which creates the user defined conceptual structure, and user-interface client (Java virtual machine). This architecture was chosen to simplify modification of system components, particularly the user interface. The database stores the user created index structures on a file server. The index application retrieves information from the database and creates the user defined structure, while Java and Javascript are used to support the user interface. From the users perspective, the Java-based application looks and feels as though it is a part of the PDP (Programming Design and Programming) hypertext document. Upon invoking the index option from the PDP hypertext document, the applet is downloaded and the application executed. The users information is retrieved, the index structure is created and finally the users index and the commands to modify, save, and exit the indexing engine are displayed.

## 6.2 A Query Tool for Information Retrieval

DUSIE provides content retrieval in a manner similar to Bruzas two-level architecture for hypertext documents through a top level index information and a bottom level containing content nodes and links. In the DUSIE implementation the hyperindex is a set of indices linked together. When an index term describing the required information is found, the objects from the underlying hypertext document base are retrieved for examination. Retrieving information in this manner is retrieval through navigation. After selecting an index term, the information associated with the hypertext base link is displayed in the browser. The annotation can also be displayed in the browser. Information may also be retrieved using the related concepts entries, facilitating the retrieval of semantically related materials. The user can move directly to the desired information as well as access related information without having to remember locations, links or other structural details. As a reflection of the users conceptual domain, the index terms might be considered metanodes which are linked to other metanodes via terms in the related concepts term set. In this way the index system expresses local relations among clusters of nodes.

## 7 Conclusion And Future Work

We propose in this report DaVIME (Data Visualization, Indexing and Mining Engine), a software architecture that does data visualization, indexing and mining in a integrated environment. DaVIME presents to the users a unified view of information service. When a user issues a information service request, DaVIME calls upon appropriate software component (agent) to provide the service requested. Depending on the type of the service requested, DaVIME may call one or more softbot into action to satisfy the user request. DaVIME is an open, extensible architecture that allows researchers and developers to add software components (agents) incrementally. Currently DaVIME includes the following components. Document Explorer analyzes text information in large document collections and present to the user with a suite of visualization tools. DUSIE extends the hierarchical indexing schemes currently used in popular browsers to let user build content-based, searchable indexes. ParaCrawler is a parallel Web search engine which uses novice ranking and indexing algorithm to provide users with more accurate search results in shorter time. Gis2web allows users to access GIS data from the Web.

We will investigate many of the implementation issues from the design. We will also make the coordination among different softbots more harmonic, seamless. The behavior of the system, as well as that of various individual components when they actually open service to the Internet will be of interest.

## References

- [1] Andrews, K. and Kappe, E. 1994. "Soaring through hyperspace: A snapshot of Hyper-G and its Harmony client". *Proceedings of Eurographics Symposium of Multimedia/Hypermedia in Open Distributed Environments*, 181-191. Berlin: Springer Verlag.
- [2] Andrews, K. (1995.) "Visualising cyberspace: Information visualisation in the Harmony Internet browser." *Proceedings of Information Visualization*, 97-104. Los Alamitos, CA: IEEE.
- [3] Apperly, M. D., Tzavaras, I., and Spence, R. (1982.) "A bifocal display technique for data presentation". *Proceedings of Eurographics '82*, 27-43.
- [4] Armstrong, R., Freitag, D., Joachims, T. and Mitchell, T. , "Webwatcher: A learning apprentice for the world wide web." In *Working notes of AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments*, 1995, AAAI Press, Stanford University.
- [5] Arthur, K. W., Booth, K. S., and Ware, C. (1993.) "Evaluating 3D task performance for fish tank virtual worlds". *ACM Transactions on Information Systems*, 11(3), 239-265.
- [6] Baldeschwiler, J.E., Blumofe, R.D., and Brewer, E.A., "ATLAS: An infrastructure for Global Computing", *Proceedings of the Seventh ACM SIGOPS European Workshop on System Support for Worldwide Applications*, September 1996
- [7] Bates, M. J. 1986. "Subject access in online catalogs: A design model." *Journal of the American Society for Information Science*, 37(6), 357-386.
- [8] Batini, C., Nardelli, E., and Tamassia, R. (1986.) "A layout algorithm for data-flow diagrams". *IEEE Transactions on Software Engineering*, 12(4), 538-546.
- [9] Beeri, C. and Kornatzky, Y. (1990). "A Logical Query Language for Hypertext Systems" *Proceedings of the European Conference on Hypertext*, ECHT'90.

- [10] Belkin, N. J. and Kwasnik, B. H. (1986.) "Using structural representations of anomalous states of knowledge for choosing document retrieval strategies." *Proceedings of ACM SIGIR*, 11-22. New York: ACM.
- [11] Benford, S., Snowden, S., Greenhalgh, C., Ingram, R., Knox, I., and Brown, C. (1995.) "VR-VIBE: A virtual environment for co-operative information retrieval." *Eurographics '95*, 349-360.
- [12] Brecht, T., Sandhu, H., Shan, M., and Talbot, J., "ParaWeb: Towards World-Wide Supercomputing", *Proceedings of the Seventh ACM SIGOPS Europe Workshop on System Support for Worldwide Applications*, September 1996
- [13] Brown, C.M., Danzig, P.B., Hardy, D., Manber, U. and Schwartz, M.F., "The Harvest Information Discovery and Access System", in *Proceedings of the Second International World Wide Web Conference*, 1994, pp. 763-771.
- [14] Bruza, Peter D. (1990) "Hyperindices: A Novel Aid for Searching Hypermedia." *Proceedings of the European Conference on Hypertext*, ECHT'90.
- [15] Chen, Z. and Maass, W., "On-line learning of rectangles and unions of rectangles," *Machine Learning*, Special Issue of the Fifth ACM Annual Conference on Computational Learning Theory, 17, pages 201-223, 1994.
- [16] Christiansen, B., Cappello, P., Ionescu, M.F., Neary, M.O., Schausser, K.E., and Wu, D., "Javelin: Internet-Based Parallel Computing Using Java", 1997 ACM Workshop on Java for Science and Engineering Computation, June 1997
- [17] Cooke, N. M., Durso, F. T., and Schvaneveldt, R. W. (1986.) "Recall and measures of memory organization". *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 12(4), 538-549.
- [18] Croft, W. B. and Thompson, R. H. (1987.) "I3R: A new approach to the design of document retrieval systems". *Journal of the American Society for Information Science*, 38, 236-261.
- [19] Cutting, D., Karger, J. and Turkey, J. "Scatter/Gather: A cluster-based approach to browsing large document collections", *Proceedings of the Fifteenth International Conference on Research and Development in Information Retrieval*, 1992, pages 318 to 329.
- [20] Dearholt, D. W. and Schvaneveldt, R. W. (1990.) "Properties of Pathfinder networks". In R. W. Schvaneveldt (Ed.), *Pathfinder associative networks: Studies in knowledge organization*, 1-30. Norwood, NJ: Ablex.
- [21] Di Battista, G., Eades, P., Tamassia, R., and Tollis, I G. (1994). "Algorithms for drawing graphs: An annotated bibliography". *Computational Geometry*, 4, 235-282.
- [22] Eades, P. and Xuemin, L. (1989.) "How to draw a directed graph". IEEE Workshop on Visual Languages, 13-17. Los Alamitos, CA: IEEE.
- [23] Eades, P. (1984.) "A heuristic for graph drawing". *Congressus Numerantium*, 42, 149-160.
- [24] Esposito, C. (1988.) "Graph graphics: Theory and practice". *Computer Mathematics Applications*, 15(4), 247-253.
- [25] ESRI Educational Services, *Programming with Avenue*, Environment Systems Research Institute, Inc. 1995

- [26] Etzioni, O., "The World-Wide Web: Quagmire or Gold Mine?", *Communications of the ACM*, 39(11), November 1996
- [27] Fayyad, U., Piatetsky-Shapiro, G., and Smyth, P., "The KDD Process for Extracting Useful Knowledge from Volumes of Data", *Communications of the ACM*, 39(11), November 1996
- [28] Fowler, R. H., Fowler, W. A. L., Williams, J. L., "Document Explorer Visualizations of WWW Document and Term Spaces", Technical Report NAG9-551, 96-6, Department Of Computer Science, The University of Texas - Pan American, 1996
- [29] Fowler, W.A.L., and Fowler, R.H. (1996). "Networks, Workstations, Multimedia, and Electronic Communities: Creating a University Learning Environment." *Proceedings of ED-TELECOM, World Conference on Educational Telecommunications 1996*, 103-108.
- [30] Fowler, R.H., Fowler, W.A.L., and Williams, J.L. (1996). "3D Visualization of the WWW Semantic Content for Browsing and Query Formulation." *WebNet 96 - World Conference of the Web Society Proceedings*, Charlottesville, VA. 147-152.
- [31] Fowler, W.A.L. and Fowler, R.H. (1993). "A hypertext-based approach to computer science education unifying programming principles." *Journal of Educational Multimedia and Hypermedia*, 2(4),443-441.
- [32] Fowler, W. A. L., Fowler, R. H., Williams, J. L., Palacios, J. X. R., and Palacios, J. X., "DUSIE: Augmenting a Static Web with User Defined Content and Link Structure", Submitted for publication.
- [33] GIMPS Sets Another Record! —  $2^{2976221} - 1$  is prime.  
<<http://www.utm.edu/research/primes/notes/2976221/>>
- [34] Geographic Resources Analysis Support System,  
<<http://www.cecer.army.mil/grass/GRASS.main.html>>
- [35] Gershon, N. D., LeVesseur, J., Winstead, J., Croall, J., Pernick, A., and Ruh, W. (1995.) "Visualizing Internet resources." *Proceedings of Information Visualization*, 122-128. Los Alamitos, CA: IEEE.
- [36] Gershon, N. D. 1994. (Panel chair.) "Information visualization: The next frontier". *ACM SIGGRAPH '94 Conference Proceedings*, 485-486. New York: ACM.
- [37] Halasz, F. (1988). "Reflections on Notecards: Seven Issues for Next Generation Hypermedia Systems." *Communications of the ACM*, 31, 836-852.
- [38] Han, J., Zaiane, O.R. and Fu, Y., "Resource and Knowledge Discovery in Global Information System: A Scalable Multiple Layered Database Approach", *Proceedings of a Forum on Research and Technology Advances in Digital Libraries (ADL'95)*, McLean, VA, May 1995
- [39] Hemmje, M., Kunkel, C., and Willett, A. (1994.) "LyberWorld - A visualization user interface supporting fulltext retrieval." *Proceedings of ACM SIGIR*, 249-259. New York: ACM.
- [40] Hendley, R. J., Drew, N. S., Wood, A. M., and Beale, R. (1995.) "Narcissus: Visualizing information." *Proceedings of Information Visualization*, 90-97. Los Alamitos, CA: IEEE.
- [41] Ingwerson, P. and Wormell, I. 1986. "Improved subject access, browsing and scanning mechanisms in modern online IR." the Proceedings of ACM SIGIR, 68-76. New York: ACM.

- [42] Kamada, T. and Kawai, S. (1989.) "An algorithm for drawing general undirected graphs". *Information Processing Letters*, 31, 7-15.
- [43] Knoblock, C. and Levy, A. Working notes of AAAI Spring Symposium on Information Gathering from Heterogeneous, Distributed Environments, 1995, AAAI Press, Stanford University.
- [44] Kumar, A. and Fowler, R. H. (1994.) "A spring modeling algorithm to position nodes of an undirected graph in three dimensions", Technical Report NAG9-551-4, Department of Computer Science, University of Texas - Pan American, Edinburg. Available at <<http://www.cs.panam.edu/info-vis>>.
- [45] Lewis, D. and Gale, W. "Training text classifiers by uncertainty sampling", *Proceedings of the Seventeenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1994.
- [46] Makinen, E. (1988.) "On circular layouts". *International Journal of Computing Mathematics*, 24, 29-37.
- [47] Marchionini, G. and Maurer, H., "The Roles of Digital Libraries in Teaching and Learning", *Communications of the ACM*, 38(4), April 1995
- [48] Maron, M. E. and Kuhn, J. L. 1960. "On relevance, probabilistic indexing, and information retrieval". *Journal of the Association for Computing Machinery*, 7(3), 216-244.
- [49] McCahill, M. P. and Erickson, T. (1995.) "Design for a 3D spatial user interface for Internet Gopher." *Proceedings of ED-MEDIA 95*, 39-44. Charlottesville, VA: ACE.
- [50] Meng, X., Fowler, R. and Rieken, E. (1997) "Bridging the gap between GIS and WWW", Poster presentation at the Sixth International World Wide Web Conference, April 7-11, 1997, Santa Clara, California.
- [51] Merssene Primes: History, Theorem and Lists.  
<<http://www.utm.edu/research/primes/merssene.shtml>>
- [52] Mukherjea, S. Foley, J. D., and Hudson, S. (1995.) "Visualizing complex hypermedia networks through multiple hierarchical views." *Proceedings of CHI '95*, 331-337. New York: ACM.
- [53] Munzner, T. and Burchard, P. (1996.) "Visualizing the structure of the World Wide Web in 3D hyperbolic space." Available at <<http://www.geom.umn.edu/docs/webogl/>>, The Geometry Center, University of Minnesota.
- [54] Nuthall, G. and Alton-Lee, A. (1993). "Predicting learning from student experience of teaching: A theory of student knowledge in classrooms." *American Education Research Journal*, 30, 799-840.
- [55] Perkowitz, M., Doorenbos, R., Etzioni, O., Weld, D. "Learning to understand information on the Internet: An example-based approach", *Journal of Intelligent Information Systems*, Vol. 8, No. 2, March/April, 1997.
- [56] Presidential Task-force on Psychology in Education (1993). Learner-Centered Psychological Principles: Guidelines for School Redesign and Reform.
- [57] Robertson, G. G., Card, S. K., and Mackinlay, J. D. (1993.) "Information visualization using 3D interactive animation". *Communications of the ACM*, 36(4), 57-71.

- [58] Rowe, L. A., Davis, M., Messinger, E., Meyer, C., Spirakis, C., and Tuan, A. (1987.) "A browser for directed graphs". *Software Practice and Experience*, 17(1), 61-76.
- [59] Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. New York: Addison-Wesley.
- [60] Schatz, B. R. and Hardin, J, B. 1994. "NCSA Mosaic and the World-Wide Web: Global hypermedia protocols for the Internet", *Science*, 265, 895-901.
- [61] Schvaneveldt, R. W. (Ed.), (1990.) *Pathfinder associative networks: Studies in knowledge organization*. Norwood, NJ: Ablex.
- [62] Schvaneveldt, R. W., Dearholt, D. W., & Durso, F. T. 1989. Network structures in proximity data. In G. H. Bower (Ed.), *The psychology of learning and motivation: Advances in research and theory*, 249-284. New York: Academic Press.
- [63] Sclaroff, S., Taycher L. and Cascia, M. "ImageRover: A content-based image browser for the World Wide Web", *Proc. IEEE Workshop on Content-Based Access of Image and Video Libraries*, 1997.
- [64] Selberg, E., Etzioni, O. "The MetaCrawler Architecture for Resource Aggregation on the Web", *IEEE Expert 1997*
- [65] Sparck Jones, K. (1972.) "A statistical interpretation of term specificity and its application in retrieval." *Journal of Documentation*, 28(1), 11-20.
- [66] Tamassia, R., Di Battista, G., and Battini, C. (1988.) "Automatic graph drawing and readability of diagrams". *IEEE Transactions on Systems, Man, and Cybernetics*, 18(1), 61-79.
- [67] *Understanding GIS - The Arc/Info Method*, Environmental Systems Research Institute, Inc., 1993
- [68] Vanhelsuwe, L., "Create Your Own Supercomputer With Java" *JavaWorld*, 2(1), January 1997
- [69] Ware, C. and Franck, G. (1996.) "Evaluating stereo and motion cues for visualizing information nets in three dimensions". *ACM Transactions on Graphics*, 15(2), 121-140.
- [70] Weingrad, P., Hay, K.E., Jackson, S., Boyle, R.A., Guzdial, M. and Soloway, E. (1993). "Student Composition of Multimedia Documents: A Preliminary Study. " *Proceedings of ED-MEDIA 93-World Conference on Educational Multimedia and Hypermedia*, 541-548.