

EXHIBIT A-6

Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					<p>describe methods and apparatuses for navigating in a multidimensional space, but do not indicate that the navigation takes place in a “graphical” multidimensional space. Instead, the claims involve displaying a single portion of text-based data (a textual, not pictorial, display) on an axis and displaying a second related point on an axis related to the first point. There is nothing in these claims to indicate that these axes are displayed graphically. In fact, when read in light of the remainder of the specification and Figures 18–20 and 22 it is apparent that these claims pertain to primarily textual displays of information.</p> <p>The remaining aspects of the invention are not targeted toward a computer display, and therefore do not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
8.	Brief Description of the Drawings	Col. 8, Ins. 17-39.	No	No	This text briefly describes Figures 1–17. The text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display, either standing alone, or in combination with those Figures. For further analysis of the Figures and accompanying text, see below.
9.	Brief Description of the	Col. 8, Ins. 40-	No	Yes*	This text briefly describes Figures 18–23. The text

Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
	Drawings	59.			does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display, either standing alone, or in combination with those Figures. For further analysis of the Figures and accompanying text, see below.
10.	Detailed Description	Col. 8, ln. 60- Col. 9, ln. 5.	No	No	This part of the specification is the introduction to the more detailed description of the embodiments. This introductory language lists three problems purportedly solved by the invention, but does not mention any specifics of a computer display, such as a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.
11.	Detailed Description: Overview of Embodiments	Col. 9, ln. 6- Col. 12, ln. 14.	No	No	This part of the specification lists the purported advantages of the invention. This text describes difficulties of the “conventional” means of presenting Australian legislation to users. The text goes on to describe “key aspects” of the invention, including how the text-based data is divided into “optimum storage units,” how it is marked up using a markup language such as SGML, how it is encoded with links and attributes, and how database technologies are used in the storage and management of the SGML data. The specification states that “[t]he embodiments of the

Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					<p>invention organize, process and present information in a way that is significantly different than conventional structures, processes and presentation.” However, this part of the specification does not describe or disclose what that purportedly novel “presentation” is, or what it looks like.</p> <p>This section of the specification does not contain any description of a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
12.	Detailed Description: Overview of Embodiments	Col. 12, lns. 15-35.	No	No	See below, comments pertaining to Figures 1–3.
13.	Detailed Description: First Embodiment	Col. 12, ln. 36- Col. 14, ln. 46.	No	No	<p>This section of the specification describes the first embodiment. It states that the first embodiment has six dimensions: location (its address); type (its genesis); jurisdiction (its class); subject (its content description); depth (extend of content); and time (the point in time at which the information is viewed).</p> <p>The specification describes encoding this data with a markup language such as SGML or XML using a format described in a DTD (Document Type Definition) and possibly a SSM (Style Sheet Mechanism). The description of this markup language does not include any description of what the computer display for the encoded data would look like.</p> <p>This section also describes the process of dividing a document into more than one portion of text-based</p>

Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					<p>data, and re-storing the portions when they change. This description does not include any description of a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p> <p>Finally, this section states that the commercial software programs Folio Views or Dynatext might be used to display, store, and retrieve the text-based data. This part of the specification does not describe a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display. Moreover, I am not aware that either Folio Views or Dynatext provides a user with the ability to view the stored information in a graphical representation of a multidimensional space.</p>
14.	Detailed Description: First Embodiment	Col. 14, ln. 47- Col. 18, ln. 44.	No	No	See below, comments pertaining to Figure 4–17.
15.	Detailed Description: Second Embodiment	Col. 18, ln. 45- Col. 19, ln. 24.	No	No	<p>The embodiment described in this section also uses a markup language to encode text-based data, and involves dividing documents into portions of text-based data. It describes storing this data in a relational database.</p> <p>There is no information disclosed in this section about a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space</p>

Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					on a video display.
16.	Detailed Description: Second Embodiment	Col. 19, ln. 25- Col. 21, ln. 34.	No	Yes*	<p>Although this part of the specification falls under the heading of “Second Embodiment,” it appears to be describing a different topic altogether. It is unique to the '228 patent, and purports to describe a “method, apparatus and computer program product for navigating in a multidimensional space containing an electronic publication formed from predefined portions of text-based data encoded using a markup language”</p> <p>This section describes how the inventors envisioned displaying text-based data to a user. One of the portions of text-base data is displayed in a “first display region” on a computer screen, called a “content frame.” Then, a “view into a primary axis” is displayed in a second area called a “reference frame.” Finally, a title bar indicating the current view is displayed in the “reference anchor.” This reference anchor contains tools for navigating the displayed axis.</p> <p>The specification also describes displaying a second axis, which could represent any other attribute, such as time, amending legislation, case law, annotations, or subject index. However, the patent states that the method only allows “<u>one</u> of a number of potential axes (associated with the base node) to be selected and subsequently navigated.” (Col. 20, lns. 4-6). That navigation is accomplished “by means of activating ‘links’ in the displayed base node,” which causes the</p>

Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					<p>screen to be redrawn and display a new member of the selected axis.</p> <p>That is the method that the '228 patent describes for navigating the axes in the multidimensional space. This method for navigating does not purport to include displaying a graphical representation of a multidimensional space. Instead, it describes a display where one portion of text-based data is displayed at a time, and where one link provides the user the ability to select a new portion of text-based data based on its attributes.</p> <p>This section also provides definitions for several terms used in the '228 patent. None of these definitions disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
17.	Detailed Description: Multi-axis Viewing Interface	Col. 21, ln. 35- Col. 22, ln. 25.	No	Yes*	<p>This part of the specification describes a node with seven associated axes. These axes are “sequential,” “hierarchical,” “temporal,” “source,” “case law,” “annotations,” and “subject.”</p> <p>This part of the specification describes the types of information stored in the multidimensional space. However, this section concludes by saying that “the problem is how to access this data in a meaningful way.” Before launching into a description of Figures 18–30, this section concludes by saying that a “two frame screen” is provided to “provide an untrained</p>

Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					<p>user with full multi-axis access to MALT type data whilst maintaining a consistent look and feel throughout.”</p> <p>This “two frame screen,” as is shown in the Figures and described in the accompanying text is not a graphical representation of a multidimensional space. Instead, as discussed below it is a primarily textual display.</p>
18.	Detailed Description: Multi-axis Viewing Interface	Col. 22, ln. 26- Col. 24, ln. 4.	No	Yes*	See below, comments pertaining to Figures 18–20.
19.	Detailed Description: Higher Level Scoping	Col. 24, ln. 5- Col. 29, ln. 13.	No	Yes*	<p>This part of the specification does not pertain to a user interface or any display of a multidimensional space. Instead, this part of the specification describes a convention for storing the text-based data of a multidimensional space in a set of flat file records. This part of the specification also provides some Document Type Definitions (including Tables 2–8) describing how to encode the stored portions of text-based data with a markup language, such as SGML. In short, nothing in this part of the specification describes a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
20.	Detailed Description: Higher Level Scoping	Col. 29, ln. 14- Col. 30, ln. 50.	No	Yes*	See below, comments pertaining to Figures 21–22.

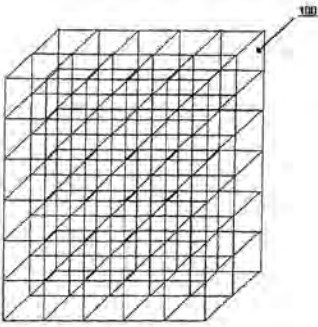
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
21.	Detailed Description: Higher Level Scoping	Col. 30, Ins. 51-59.	No	Yes*	These lines of the specification state that the text-based data may be encoded with a markup language, such as XML and stored as records in a database. This part of the specification does not describe any sort of display, and does not describe a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.
22.	Detailed Description: Higher Level Scoping	Col. 30, In. 60- Col. 32, In. 45.	No	Yes*	See below, comments pertaining to Figure 23.
23.	Detailed Description	Col. 32, Ins. 46-67.	No	Yes*	This is merely the conclusion of the specification. It does not disclose or describe any new information, and does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.
24.	Appendix A	Scattered between Col. 2, In. 5 and In. 47; Cols. 33-44.	No	No	This Appendix shows "Example Research." It is purely textual. It does not depict any information on a computer screen. This Appendix is apparently an attempt to demonstrate the difficulty of researching Australian law in January 1997. It does not depict a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.
25.	Appendix B	Col. 12, Ins. 58-64; Cols.	No	No	This Appendix is a Document Type Definition for Acts and Regulations. According to the specification, in

Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
		45-68.			<p>this invention “[a] DTD is used to define the structure of publication data, preferably being legislation, down to a comprehensive level. This is done by using information coded in conjunction with any one of a number of off-the-shelf, free-text retrieval software packages (eg, Folio Views or Dynatext) to deliver the information to the end user.”</p> <p>This DTD does not depict a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display. Instead, it merely provides rules and guidelines for inserting the markup language into the text-based data.</p>
26.	Appendix C	Col. 12, Ins. 48-57; Cols. 69-106.	No	No	<p>This Appendix is the “Relational Database Specifications,” which defines the structure of the database for one of the embodiments. The database structure does not describe or define a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display. It simply shows how data might be stored within a particular database design.</p>
27.	Appendix D	Col. 14, Ins. 32-34; Cols. 107-150.	No	No	<p>This Appendix is a “Keying Guide for Australian Legislation Documents.” It provides guidelines for how markup language information and attributes should be inserted into the text-based data. This Appendix does not depict or describe a graphical representation of a multidimensional space or the</p>

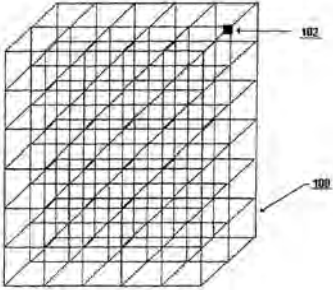
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.
28.	Appendix E	Cols. 151-162.	No	No	This Appendix is a Document Type Definition (DTD) for the second embodiment. Like the other DTDs described in the patent, this DTD does not describe or disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.
29.	<p>Figure 1</p>  <p>Fig. 1</p>	Col. 8, Ins. 22-23; Col. 12, Ins. 15-20.	No	No	<p>The specification says that Figure 1 “illustrates a grid of a multidimensional space according to the first embodiment.”</p> <p>The specification states that while this Figure “illustrates a multi-dimensional space,” which is represented by a “layered grid,” “in reality (in the case of more than two dimensions), they are at all angles and inclines.”</p> <p>In other words, the inventors chose to portray a three dimensional space with a “layered grid,” even though this depiction was not true to the reality of a multidimensional space.</p> <p>This Figure does not show a multidimensional space because a multidimensional space must be capable of or involve <u>more than</u> three dimensions. The '228 patent expressly includes seven dimensions. Moreover, this “layered grid” has very finite edges,</p>

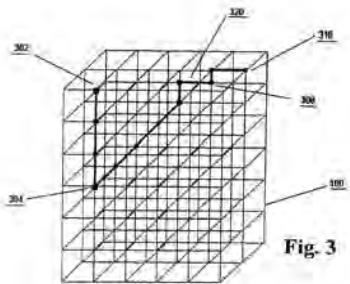
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					<p>showing that it is bounded, whereas the multidimensional space of the patents does not have boundaries.</p> <p>Furthermore, this Figure does not purport to depict a display on any video screen or computer screen, as required by the patent.</p> <p>A person of ordinary skill in the art would not understand that this Figure was intended to depict or describe a way of displaying a multidimensional space in a system for electronically publishing or searching text-based data. Instead, this appears to be a conceptual diagram meant only to help the reader of the patent understand how a three dimensional space might be visualized.</p> <p>This Figure and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display. From this Figure it is not apparent to a person of ordinary skill in the art that the inventors possessed a graphical representation of a multidimensional space including up to at least seven dimensions on a computer screen.</p>
30.	Figure 2	Col. 8, ln. 24; Col. 12, lns. 21-27.	No	No	<p>The specification says that Figure 2 “illustrates the effect of the various axes.”</p> <p>The specification shows a “node” that is located (assigned, or mapped) to a place on the layered grid. Like Figure 1, Figure 2 does not depict a graphical</p>

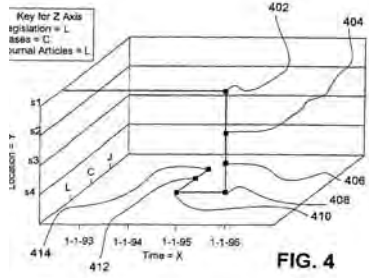
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
	 <p data-bbox="331 662 380 683">Fig. 2</p>				<p data-bbox="1312 349 1871 380">representation of a multidimensional space.</p> <p data-bbox="1312 391 1997 532">Once again, the inventors chose to portray a three dimensional space with a “layered grid,” even though this depiction was not true to the reality of a multidimensional space.</p> <p data-bbox="1312 544 1997 685">This Figure does not show a multidimensional space because a multidimensional space must be capable of or involve <u>more than</u> three dimensions. The ‘228 patent expressly includes seven dimensions.</p> <p data-bbox="1312 696 1976 837">Moreover, this “layered grid” has very finite edges, showing that it is bounded, whereas the multidimensional space of the patents does not have boundaries.</p> <p data-bbox="1312 849 1997 1214">Furthermore, this Figure does not purport to depict a display on any video screen or computer screen, as required by the patent. A person of ordinary skill in the art would not understand that this Figure was intended to depict or describe a way of displaying a multidimensional space in a system for electronically publishing or searching text-based data. Instead, this appears to be a conceptual diagram meant only to help the reader of the patent understand how a three dimensional space might be visualized.</p> <p data-bbox="1312 1226 1997 1399">This Figure and the associated text does not disclose a graphical representation of a multidimensional space. From this Figure it is not apparent to a person of ordinary skill in the art that the inventors possessed the inventive step of graphically displaying an area of up</p>

Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					to at least seven dimensions on a computer screen.
31.	<p>Figure 3</p>  <p>Fig. 3</p>	Col. 8, Ins. 25-28; Col. 12, Ins. 28-34.	No	No	<p>The specification says that Figure 3 “illustrates the mapping of various axis intersection points, or nodes, that is used to organize, present, and find information (present and past) according to the first embodiment.”</p> <p>Figure 3 shows several points assigned to locations on the layered grid, and also shows a highlighted path between these points. Like Figure 1, Figure 3 does not depict a graphical representation of a multidimensional space.</p> <p>Once again, the inventors chose to portray a three dimensional space with a “layered grid,” even though this depiction was not true to the reality of a multidimensional space.</p> <p>This Figure does not show a multidimensional space because a multidimensional space must be capable of or involve <u>more than</u> three dimensions. The '228 patent expressly describes seven dimensions.</p> <p>Moreover, this “layered grid” has very finite edges, showing that it is bounded, whereas the multidimensional space of the patents does not have boundaries.</p> <p>Furthermore, this Figure does not purport to depict a display on any video screen or computer screen, as required by the patent. A person of ordinary skill in the art would not understand that this Figure was intended to depict or describe a way of displaying a multidimensional space in a system for electronically</p>

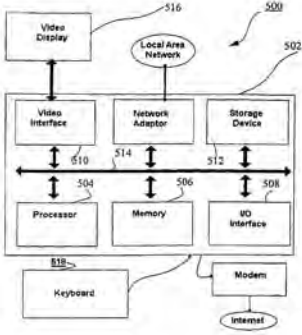
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					<p>publishing or searching text-based data. Instead, this appears to be a conceptual diagram meant only to help the reader of the patent understand how a three dimensional space might be visualized.</p> <p>Finally, this Figure does not describe or disclose any ability to link to a graphical representation of a multidimensional space populated only with points representing “related” portions of text-based data where the user can select one of the points to display the underlying text.</p> <p>This Figure and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display. From this Figure it is not apparent to a person of ordinary skill in the art that the inventors possessed a graphical representation of a multidimensional space including up to at least seven dimensions on a computer screen.</p>
32.	<p>Figure 4</p> 	<p>Col. 8, Ins. 29-30; Col. 14, ln. 47-Col. 15, ln. 45.</p>	No	No	<p>The specification says that Figure 4 “illustrates the application of legal information to mapped nodes according to the first embodiment.”</p> <p>According to the specification, “Figure 4 provides an example of how legal information is navigated by an end user.”</p> <p>The Figure shows several pieces of legislation that are located (assigned, or mapped) to a place on the layered grid depending on their attributes. Like Figure 1,</p>

Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					<p>Figure 4 does not depict a graphical representation of a multidimensional space.</p> <p>Once again, the inventors chose to portray a three dimensional space with a “layered grid,” even though this depiction was not true to the reality of a multidimensional space.</p> <p>This Figure does not show a multidimensional space because a multidimensional space must be capable of or involve <u>more than</u> three dimensions. The '228 patent expressly describes seven dimensions.</p> <p>Moreover, this “layered grid” has finite edges, showing that it is bounded, whereas the multidimensional space of the patents does not have boundaries.</p> <p>Furthermore, this Figure does not purport to depict a display on any video screen or computer screen, as required by the patent. A person of ordinary skill in the art would not understand that this Figure was intended to depict or describe a way of displaying a multidimensional space in a system for electronically publishing or searching text-based data. Instead, this appears to be a conceptual diagram meant only to help the reader of the patent understand how a three dimensional space might be visualized.</p> <p>This Figure and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display. From this Figure it is not apparent</p>


Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					to a person of ordinary skill in the art that the inventors possessed a graphical representation of a multidimensional space including up to at least seven dimensions on a computer screen.
33.	<p>Figure 5</p>  <p>Fig. 5</p>	Col. 8, Ins. 31-33; Col. 16, Ins. 12-63.	No	No	<p>The specification says that Figure 5 “is a block diagram illustrating a general purpose computer that can be used to implement the electronic publishing system according to the first embodiment.”</p> <p>This Figure simply depicts a typical general purpose computer. It does not provide any insight whatsoever into how information might be graphically displayed to an end user on the “video display” that is depicted as 516. Figure 5, and the associated text, does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
34.	Figure 6	Col. 8, Ins. 34-35; Col. 15, ln. 46-Col. 16, ln. 11.	No	No	<p>The specification says that Figure 6 “is a flow diagram illustrating the method of electronic publishing.”</p> <p>This “flow diagram” shows possible steps in a publishing process. The only component of this diagram that is related to how information is displayed on a computer screen is 624, which explains that a commercial software product, such as “Folio Views” or “Dynatext” or possibly Topic or an internet markup language, HTML, might be used as text retrieval software. However, disclosing these possible text</p>

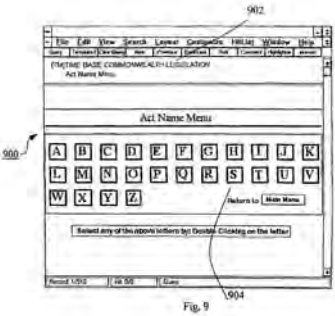
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
	<p>FIG. 6</p>				<p>retrieval mechanisms would not have caused a person of ordinary skill in the art to believe that the inventors possessed a way of graphically displaying a multidimensional space (with at least 4 dimensions and up to at least 7 dimensions) on a computer screen. I am not aware that any of those technologies permitted such a display of text-based data as of January 1997. For these reasons, Figure 6 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
35.	<p>Figure 7</p> <p>Fig. 7</p>	<p>Col. 8, Ins. 36-38; Col. 16, ln. 64- Col. 17, ln. 21.</p>	No	No	<p>The specification says that Figures 7–17 “are screen shots illustrating operation of a first embodiment as a software application executing on a general purpose computer.”</p> <p>Figures 7–17 use Folio Views as the retrieval software. According to the specification, these Figures “are screen shots illustrating navigation or movement around the information.”</p> <p>Figure 7 shows the “opening screen 700, which the end user sees when the program is started.”</p> <p>Some of the components on this Folio Views screen shot are: “drop menus that provide access to all function”; a customizable toolbar 704”; a “status bar 708.” The Start button takes a new user to a help menu. The Main Menu button takes the user to the screen shot of Figure 8.</p>


Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					<p>Although there are lines and squares on this screen shot, those are for the purpose of providing menus to the end user. As the specification states, this is a “standard windows interface featuring drop menus that provide access to all functions.”</p> <p>There is no indication whatsoever that this Figure is an inventive graphical representation of a multidimensional space. Instead, this is a typical Folio Views screen with ordinary user interface components. For these reasons, Figure 7 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
36.	<p>Figure 8</p>  <p>800</p> <p>802</p> <p>Fig. 8</p>	Col. 8, Ins. 36-38; Col. 16, Ins. 22-31.	No	No	<p>The specification says that Figures 7–17 “are screen shots illustrating operation of a first embodiment as a software application executing on a general purpose computer.”</p> <p>Figures 7–17 use Folio Views as the retrieval software. According to the specification, these Figures “are screen shots illustrating navigation or movement around the information.”</p> <p>Figure 8 shows “a main selection menu 800” where “the user can see the currency of the total information.” Here, the user can see a list of bullet points in a menu, which are jump links.</p> <p>Figure 8 has similar drop down menus and features as Figure 7, but these are not inventive, and are not a</p>


Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					<p>graphical representation of a multidimensional space. There is no indication whatsoever that this Figure is an inventive graphical representation of a multidimensional space. Instead, this is a typical Folio Views screen with ordinary user interface components. For these reasons, Figure 8 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
37.	<p>Figure 9</p>  <p>The image shows a screenshot of a software application window titled 'Act Name Menu'. The window has a standard menu bar with options like File, Edit, View, Search, Layout, Configuration, Help, and Window. Below the menu bar, there is a title bar that says 'Act Name Menu'. The main content area contains a grid of 26 letter buttons arranged in three rows: Row 1: A, B, C, D, E, F, G, H, I, J, K; Row 2: L, M, N, O, P, Q, R, S, T, U, V; Row 3: W, X, Y, Z. Below the grid, there is a text box with the prompt 'Select any of the above letters by Double-Clicking on the letter'. At the bottom of the window, there is a status bar with the text 'Home View 10:50 Done'. The number '900' is in the top left corner, and '904' is in the bottom right corner. The caption 'Fig. 9' is centered below the screenshot.</p>	<p>Col. 8, Ins. 36-38; Col. 16, Ins. 32-41.</p>	No	No	<p>The specification says that Figures 7–17 “are screen shots illustrating operation of a first embodiment as a software application executing on a general purpose computer.”</p> <p>Figures 7–17 use Folio Views as the retrieval software. According to the specification, these Figures “are screen shots illustrating navigation or movement around the information.”</p> <p>Figure 9 shows “the ‘Act Name Menu’ screen 900” where the user can see letter buttons that are actually links to Acts beginning with the corresponding letter. These letter buttons are not a graphical representation of a multidimensional space.</p> <p>Figure 9 has similar drop down menus and features as Figure 7, but these are not inventive, and are not a graphical representation of a multidimensional space. There is no indication whatsoever that this Figure is an</p>

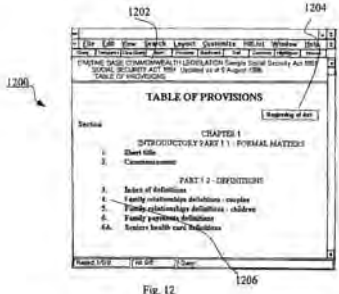
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					<p>inventive graphical representation of a multidimensional space. Instead, this is a typical Folio Views screen with ordinary user interface components. For these reasons, Figure 9 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
38.	<p>Figure 10</p>  <p>Fig. 10</p>	<p>Col. 8, Ins. 36-38; Col. 16, Ins. 42-46.</p>	<p>No</p>	<p>No</p>	<p>The specification says that Figures 7–17 “are screen shots illustrating operation of a first embodiment as a software application executing on a general purpose computer.”</p> <p>Figures 7–17 use Folio Views as the retrieval software. According to the specification, these Figures “are screen shots illustrating navigation or movement around the information.”</p> <p>Figure 10 shows a list of Acts beginning with “S.” This list is not a graphical representation of a multidimensional space.</p> <p>Figure 10 has similar drop down menus and features as Figure 7, but these are not inventive, and are not a graphical representation of a multidimensional space. There is no indication whatsoever that this Figure is an inventive graphical representation of a multidimensional space. Instead, this is a typical Folio Views screen with ordinary user interface components. For these reasons, Figure 10 and the associated text does not disclose a graphical representation of a</p>

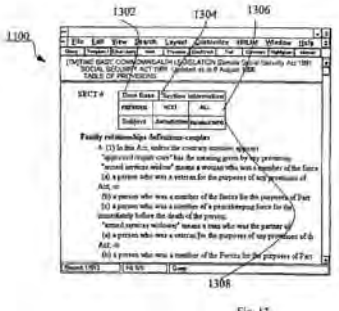
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.
39.	<p>Figure 11</p> 	Col. 8, Ins. 36-38; Col. 17, Ins. 47-54.	No	No	<p>The specification says that Figures 7–17 “are screen shots illustrating operation of a first embodiment as a software application executing on a general purpose computer.”</p> <p>Figures 7–17 use Folio Views as the retrieval software. According to the specification, these Figures “are screen shots illustrating navigation or movement around the information.”</p> <p>Figure 11 shows the current version of the Social Security Act, assuming that this is the Act selected by a user on the screen depicted in Figure 10. Figure 11 also shows a button that allows a user to link to a list of “provisions” (<i>i.e.</i>, sections) within the Act.</p> <p>Figure 11 has similar drop down menus and features as Figure 7, but these are not inventive, and are not a graphical representation of a multidimensional space. There is no indication whatsoever that this Figure is an inventive graphical representation of a multidimensional space. Instead, this is a typical Folio Views screen with ordinary user interface components. For these reasons, Figure 11 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>

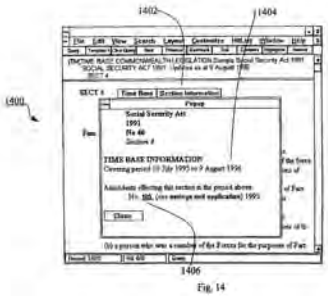
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
40.	Figure 12 	Col. 8, Ins. 36-38; Col. 17, Ins. 54-61.	No	No	<p>The specification says that Figures 7–17 “are screen shots illustrating operation of a first embodiment as a software application executing on a general purpose computer.”</p> <p>Figures 7–17 use Folio Views as the retrieval software. According to the specification, these Figures “are screen shots illustrating navigation or movement around the information.”</p> <p>Figure 12 shows the list of provisions (<i>i.e.</i>, sections) that are included in the Social Security Act being viewed in Figure 11.</p> <p>Figure 12 has similar drop down menus and features as Figure 7, but these are not inventive, and are not a graphical representation of a multidimensional space. There is no indication whatsoever that this Figure is an inventive graphical representation of a multidimensional space. Instead, this is a typical Folio Views screen with ordinary windows user interface components. For these reasons, Figure 12 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
41.	Figure 13	Col. 8, Ins. 36-38; Col. 17, ln. 62-Col. 18, ln.	No	No	<p>The specification says that Figures 7–17 “are screen shots illustrating operation of a first embodiment as a software application executing on a general purpose</p>

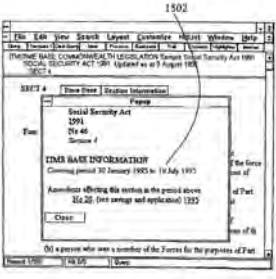
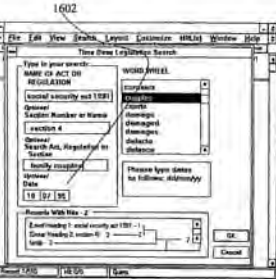
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
	 <p>1100</p> <p>1302 1304 1306</p> <p>1308</p> <p>Fig. 13</p>	13.			<p>computer.”</p> <p>Figures 7–17 use Folio Views as the retrieval software. According to the specification, these Figures “are screen shots illustrating navigation or movement around the information.”</p> <p>Figure 13 shows “the TimeBase toolbar 1302, which preferably provides eight buttons for accessing time based information.” According to the specification, this toolbar is not a standard feature of Folio Views, but was a feature programmed specifically by TimeBase.</p> <p>This screen only displays one portion, not the plurality of portions related to the current portion, and not a plurality of portions as a graphical representation of a multidimensional space. While it does contain some “buttons” on a “toolbar,” this is not a pictorial depiction of a multidimensional space, having more than three dimensions, which are axes along which, or along some combination of which, point-to-point movement is allowed. The “Previous” and “Next” buttons appear to permit a user to move from point-to-point along only one axis: the versions axis. Some of the other buttons on the toolbar simply appear to lead to either another portion of text-based data, or a list of text-based portions.</p> <p>Figure 13 has similar drop down menus and features as Figure 7, but these are not inventive, and are not a graphical representation of a multidimensional space.</p>

Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					For these reasons, Figure 13 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.
42.	Figure 14 	Col. 8, Ins. 36-38; Col. 18, Ins. 14-20.	No	No	The specification says that Figures 7–17 “are screen shots illustrating operation of a first embodiment as a software application executing on a general purpose computer.” Figures 7–17 use Folio Views as the retrieval software. According to the specification, these Figures “are screen shots illustrating navigation or movement around the information.” Figure 14 shows a screen that “appears when the user selects the Section information button,” which is button 1304 in Figure 13. This is simply a pop-up box, which contains textual information. This screen is not a graphical representation of a multidimensional space. For these reasons, Figure 14 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.
43.	Figure 15	Col. 8, Ins. 36-38; Col. 18, Ins. 21-28.	No	No	The specification says that Figures 7–17 “are screen shots illustrating operation of a first embodiment as a software application executing on a general purpose computer.”

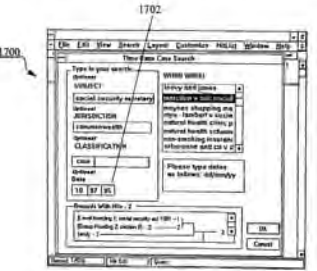
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
	 <p>Fig. 15</p>				<p>Figures 7–17 use Folio Views as the retrieval software. According to the specification, these Figures “are screen shots illustrating navigation or movement around the information.”</p> <p>Figure 15 shows “a screen 1500 which appears when the user selects the previous button” on the toolbar on Figure 13. That previous button causes this pop-up window to be displayed containing a short, textual prior version of the legislative section that was shown in Figure 13. This screen is not a graphical representation of a multidimensional space.</p> <p>For these reasons, Figure 15 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
44.	<p>Figure 16</p>  <p>Fig. 16</p>	Col. 8, Ins. 36-38; Col. 18, Ins. 29-36.	No	No	<p>The specification says that Figures 7–17 “are screen shots illustrating operation of a first embodiment as a software application executing on a general purpose computer.”</p> <p>Figures 7–17 use Folio Views as the retrieval software. According to the specification, these Figures “are screen shots illustrating navigation or movement around the information.”</p> <p>Figure 16 shows the “searching” feature provided in the first embodiment. Figure 16 is “a customized search template 1602 that includes a time base option allowing a user to search for versions of a section, for</p>


Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					<p>example.”</p> <p>This screen is not a graphical representation of a multidimensional space. While it does contain some “boxes” in which a user may enter text, and a “word wheel” showing the number of matches for the user’s search, this is not pictorial depiction of a multidimensional space, having more than three dimensions, which are axes along which, or along some combination of which, point-to-point movement is allowed. This screen shot does not depict any “dimensions.”</p> <p>Figure 16 has similar drop down menus and features as Figure 7, but these are not inventive, and are not a graphical representation of a multidimensional space. For these reasons, Figure 16 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
45.	Figure 17	Col. 8, Ins. 36-38; Col. 18, Ins. 36-44.	No	No	<p>The specification says that Figures 7–17 “are screen shots illustrating operation of a first embodiment as a software application executing on a general purpose computer.”</p> <p>Figures 7–17 use Folio Views as the retrieval software. According to the specification, these Figures “are screen shots illustrating navigation or movement around the information.”</p> <p>Figure 17 shows the “searching” feature provided in</p>

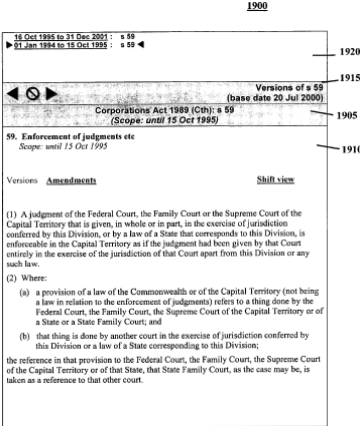
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
	 <p>Fig. 17</p>				<p>the first embodiment. Figure 17 is “a customized search template 1702 for case law which includes a time base option connecting cases to legislation at a particular date, for example.”</p> <p>This screen is not a graphical representation of a multidimensional space. While it does contain some “boxes” in which a user may enter text, and a “word wheel” showing the number of matches for the user’s search, this is not pictorial depiction of a multidimensional space, having more than three dimensions, which are axes along which, or along some combination of which, point-to-point movement is allowed. This screen shot does not depict any “dimensions.”</p> <p>Figure 17 has similar drop down menus and features as Figure 7, but these are not inventive, and are not a graphical representation of a multidimensional space. For these reasons, Figure 17 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
46.	Figure 18	Col. 8, Ins. 42-45; Col. 22, ln. 26-Col. 23, ln. 33.	No	Yes*	<p>The specification says that Figure 18 “is a screen shot of a Normal axis view of a MALT publication (with a search mode enabled) in accordance with an embodiment of the present invention.”</p> <p>Figure 18 shows “a screenshot 1800 depicting a section of legislation in accordance with an</p>

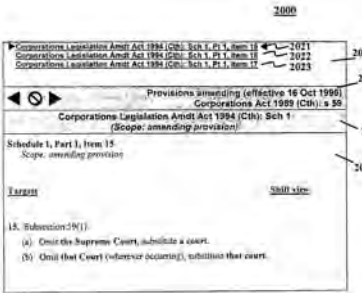
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
	 <p>Fig. 18</p>				<p>embodiment of the invention.” The “content frame” is at 1810, and it “displays the content of the current node.” The “content anchor” is shown at 1805, and it “displays the locators for the current content provision in a user friendly form.” The “reference frame” is shown at 1820, and it “contains a set of links 1821 corresponding to the members of a viewing axis associated with the current base node.” Label 1823 indicates a “reference marker” which shows the link currently selected. The buttons 1816, 1818 are for navigating the sequential axis. The buttons 1812, 1813 “allow access to the next or previous occurrence of this text [that is the search result].” Links 1807, 1808 allow the user to select different viewing axes: Versions and Amendments.</p> <p>This screen is not a graphical representation of a multidimensional space. It displays the text of a legislation section, and does not display portions of text-based data as points on a multidimensional space. The arrows (called “buttons” by the specification) are not a graphical representation of a multidimensional space. These arrows provide a user with a mechanism for moving to another portion of text-based data. They are simply icons that function as links.</p> <p>There is no indication whatsoever that this Figure is an inventive graphical representation of a multidimensional space. There is no pictorial depiction of an area not having boundaries and that has, or involves, more than three dimensions. For</p>

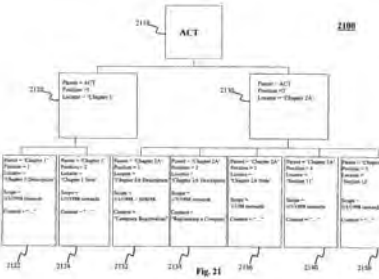
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					these reasons, Figure 18 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.
47.	<p>Figure 19</p>  <p>Fig. 19</p>	Col. 8, Ins. 46-48; Col. 23, Ins. 34-50.	No	Yes*	<p>The specification says that Figure 19 “is a screen shot of a Versions axis view of a MALT publication in accordance with an embodiment of the present invention.”</p> <p>The specification says that Figure 19 “is a Version axis view 1900 of the provision depicted in FIG. 18.” The portion of text-based data shown is an earlier version of the provision displayed in Figure 18.</p> <p>Like Figure 18, Figure 19 also shows a “content frame,” a “content anchor,” a “reference anchor,” and a “reference frame.” There is also a link to allow the user to select different viewing axis: the Amendments axis.</p> <p>This screen is not a graphical representation of a multidimensional space. It displays the text of a legislative section, and does not display portions of text-based data as points on a multidimensional space. The arrows (called “buttons” by the specification) are not a graphical representation of a multidimensional space. These arrows provide a user with a mechanism for moving to another portion of text-based data. They are simply icons that function as links.</p> <p>There is no indication whatsoever that this Figure is an</p>

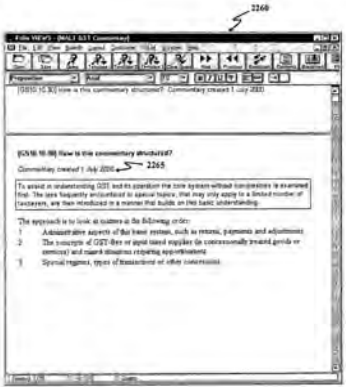
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					<p>inventive graphical representation of a multidimensional space. There is no pictorial depiction of an area not having boundaries and that has, or involves, more than three dimensions. For these reasons, Figure 19 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
48.	<p>Figure 20</p>  <p>The image shows a screenshot of a legal document viewer. At the top, there are three reference markers: 2000 (a box), 2015 (a left-pointing arrow), and 2020 (a right-pointing arrow). Below these, the text reads: "Provisions amending (effective 16 Oct 1999) Corporations Act 1989 (Cth), s 58 Corporations Legislation Amdt Act 1994 (Cth), Sch 1 (Scope: amending provision)". Underneath, it says "Schedule 1, Part 1, Item 15 - Scope: amending provision". A "Table of Contents" section is visible, with a "Skip table" link. The table lists "15. Subsection 5(1)" with two items: "(a) Omit the Supreme Court, substitute a court." and "(b) Omit the Court (reference omitted), substitute that court." At the bottom of the screenshot, the text "Fig. 20" is visible.</p>	<p>Col. 8, Ins. 49-51; Col. 23, ln. 34-Col. 24, ln. 4.</p>	<p>No</p>	<p>Yes*</p>	<p>The specification says that Figure 20 “is a screen shot of a Source axis view of a MALT publication in accordance with an embodiment of the present invention.”</p> <p>The specification says that Figure 20 is a “Source axis view 2000 of the provision under consideration.” It depicts an amending provisions rather than a portion of legislation. This Figure also has a “reference anchor” 2015, a “reference frame” 2020, a “content anchor” 2005, and a “content frame” 2010.</p> <p>This screen is not a graphical representation of a multidimensional space. It displays the text of an amending provision, and does not display portions of text-based data as points on a multidimensional space. The arrows (called “buttons” by the specification) are not a graphical representation of a multidimensional space. These arrows provide a user with a mechanism for moving to another portion of text-based data. They are simply icons that function as links.</p>


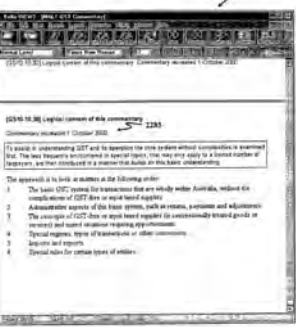
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
					<p>There is no indication whatsoever that this Figure is an inventive graphical representation of a multidimensional space. There is no pictorial depiction of an area not having boundaries and that has, or involves, more than three dimensions. For these reasons, Figure 20 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
49.	<p>Figure 21</p> 	<p>Col. 8, Ins. 52-54; Col. 29, ln. 14-Col. 30, ln. 28.</p>		<p>Yes*</p>	<p>The specification says that Figure 21 “is a block diagram representation of higher level scoping in accordance with an embodiment of the present invention.”</p> <p>This Figure is a conceptual diagram of how the “nodes” might be stored in a database. This is not meant to depict the display of an electronic system for publishing or searching text-based data.</p> <p>This does not represent a graphical representation of a multidimensional space. There are not multiple axes. There is no pictorial depiction of an area not having boundaries and that has, or involves, more than three dimensions. For these reasons, Figure 21 and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>

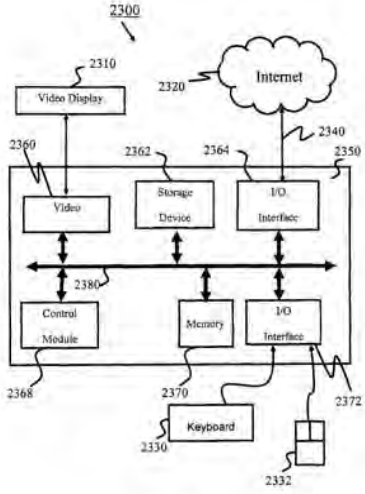
Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
50.	<p>Figure 22A</p>  <p>Fig. 22A</p>	Col. 8, Ins. 55-57; Col. 30, Ins. 29-35.	No	Yes*	<p>The specification says that Figures 22A–22C are “screen shots illustrating a commentary in accordance with a further embodiment of the invention.”</p> <p>Figure 22A is a screen shot that “shows a commentary.” This screen shot says that it is from the off-the-shelf Folio Views software product.</p> <p>This screen shot does not depict a graphical representation of a multidimensional space. It is a typical Folio Views screen, showing the contents of a portion of text-based data. For these reasons, Figure 22A and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
51.	Figure 22B	Col. 8, Ins. 55-57; Col. 30, Ins. 36-40.	No	Yes*	<p>The specification says that Figures 22A–22C are “screen shots illustrating a commentary in accordance with a further embodiment of the invention.”</p> <p>Figure 22B is a screen shot that “shows a screen shot 2270 of the commentary of FIG. 22A at a later date.” This screen shot says that it is from the off-the-shelf Folio Views software product.</p> <p>This screen shot does not depict a graphical representation of a multidimensional space. It is a typical Folio Views screen, showing the contents of a portion of text-based data. For these reasons, Figure 22B and the associated text does not disclose a graphical representation of a multidimensional space</p>

Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
	 <p>Fig. 22B</p>				<p>or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>
52.	<p>Figure 22C</p>  <p>Fig. 22C</p>	<p>Col. 8, Ins. 55-57; Col. 30, Ins. 40-50.</p>	No	Yes*	<p>The specification says that Figures 22A–22C are “screen shots illustrating a commentary in accordance with a further embodiment of the invention.”</p> <p>Figure 22C is a screen shot that “shows a screen shot 2280 of the commentary of FIGS. 22A and 22B at a yet later date.” This screen shot says that it is from the off-the-shelf Folio Views software product.</p> <p>This screen shot does not depict a graphical representation of a multidimensional space. It is a typical Folio Views screen, showing the contents of a portion of text-based data. For these reasons, Figure 22C and the associated text does not disclose a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display.</p>

Appendix 8: Detailed Analysis of '228 Patent Specification

	Section / Appendix / Figure	Text	Discloses Graphical Representation of a Multidimensional Space?	Unique to the '228 Patent?	Comments
53.	<p>Figure 23</p>  <p>FIG. 23</p>	<p>Col. 8, ln. 58; Col. 30, ln. 60 -Col. 32, ln. 45.</p>	<p>No</p>	<p>Yes*</p>	<p>The specification says that Figure 23 is “a computer program product block diagram.”</p> <p>Figure 23 is very similar to Figure 5. This Figure depicts a typical general purpose computer.</p> <p>The text describing this Figure states that “the process for navigating in a multidimensional space containing an electronic publication formed from predefined portions of text-based data encoded using a markup language can be implemented using a computer program product in conjunction with a computer system 2300 as shown in FIG. 23.”</p> <p>However, neither this Figure nor the accompanying text depicts a graphical representation of a multidimensional space or the claimed method of displaying portions as a graphical representation of the multidimensional space on a video display. It does not disclose or describe how the multiple dimensions may be pictorially depicted on the “Video Display,” or how the points might be displayed in the multidimensional space.</p>

APPENDIX 9



US006144962A

United States Patent [19]

Weinberg et al.

[11] **Patent Number:** **6,144,962**

[45] **Date of Patent:** **Nov. 7, 2000**

[54] **VISUALIZATION OF WEB SITES AND HIERARCHICAL DATA STRUCTURES**

[75] Inventors: **Amir Weinberg, Zoran; Michael Pogrebisky, Herzliya, both of Israel**

[73] Assignee: **Mercury Interactive Corporation, Sunnyvale, Calif.**

[21] Appl. No.: **08/843,265**

[22] Filed: **Apr. 11, 1997**

Related U.S. Application Data

[60] Provisional application No. 60/028,474, Oct. 15, 1996.

[51] **Int. Cl.**⁷ **G06F 17/30**

[52] **U.S. Cl.** **707/10; 707/104; 345/356; 345/357**

[58] **Field of Search** **707/10, 104; 395/200.48, 395/200.49, 200.54; 345/356, 357**

[56] References Cited

U.S. PATENT DOCUMENTS

5,295,261	3/1994	Simonetti .	
5,388,255	2/1995	Pytlík et al. .	
5,515,488	5/1996	Iloppé et al. .	
5,524,202	6/1996	Yokohama .	
5,544,310	8/1996	Forman et al. .	
5,546,529	8/1996	Bowers et al. .	
5,590,250	12/1996	Lamping et al. .	
5,619,632	4/1997	Lamping et al.	395/141
5,870,559	2/1999	Lesham et al.	395/200.54

OTHER PUBLICATIONS

Discovering Web Access Patterns and Trends by Applying OLAP and Data Mining Technology on Web Logs, Zaiane, Xim and Han, Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries (IEEE ADL '98), pp. 19-29 (1998).

Tree Visualization with Tree-Maps: 2-d Space-Filling Approach, Shneiderman, Ben, ACM Transactions on Graphics vol. 11, No. 1, Jan. 1992, pp. 92-99.

Product Brochure For Graph Layout Toolkit from Tom Sawyer Software, 8 pages (undated).

"Getting Started" Manual for Netcarta WebMapper 1.0 for Windows NT/95, dated 1996.

User's Guide for NetCarta WebMapper 1.0 for Windows NT/95, dated 1996.

(List continued on next page.)

Primary Examiner—Paul R. Lintz

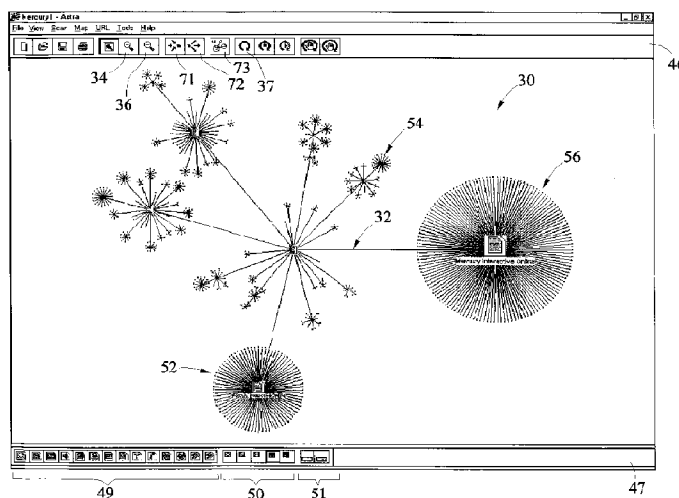
Attorney, Agent, or Firm—Knobbe, Martens, Olson & Bear LLP

[57] **ABSTRACT**

A visual Web site analysis program, implemented as a collection of software components, provides a variety of features for facilitating the analysis and management of Web sites and Web site content. A mapping component scans a Web site over a network connection and builds a site map which graphically depicts the URLs and links of the site. Site maps are generated using a unique layout and display methodology which allows the user to visualize the overall architecture of the Web site. Various map navigation and URL filtering features are provided to facilitate the task of identifying and repairing common Web site problems, such as links to missing URLs. A dynamic page scan feature enables the user to include dynamically-generated Web pages within the site map by capturing the output of a standard Web browser when a form is submitted by the user, and then automatically resubmitting this output during subsequent mappings of the site. The Web site analysis program is implemented using an extensible architecture which includes an API that allows plug-in applications to manipulate the display of the site map. Various plug-ins are provided which utilize the API to extend the functionality of the analysis program, including an action tracking plug-in which detects user activity and behavioral data (link activity levels, common site entry and exit points, etc.) from server log files and then superimposes such data onto the site map.

67 Claims, 24 Drawing Sheets

Microfiche Appendix Included
(1 Microfiche, 45 Pages)



U.S. PATENT DOCUMENTS

Product packaging (front and back) for NetCarta WebMapper 1.0, dated 1996.

“Getting Started” Guide for InContext WebAnalyzer 1.0, dated 1996.

Print-out of online help manual for InContext WebAnalyzer 1.0 product (taken from CD-ROM dated 1996).

Product packaging (front and back) for InContext WebAnalyzer 1.0, dated 1996.

Article titled “Top Tools to Manage your Web Site,” Netguide Magazine, Apr., 1997.

Selected documents from Microsoft.com Web site describing Microsoft FrontPage 1.1, downloaded and printed on Oct. 8, 1996 (5 printed pgs.).

Document titled “XSoft Licenses Information Visualization Technology to NetCarta Corp.,” dated Sep. 17, 1996, printed from NetCarta.com Website.

Press release titled “Ararat Software Announces Availability of Inwebstigator 1.0”, dated Sep. 23, 1996, printed from Ararat.com Website.

Document titled “Incontext Announces Webanalyzer for Windows 95”, dated Feb. 15, 1996.

Press release titled “Mercury Interactive Announces Industry’s First Testing Technology for the Web”, dated Jan. 1, 1996.

Press release titled “Mercury Interactive Make Stress Testing a Web Site Simple and Affordable; Astra SiteTest Generates over 4 Million Hits Using a Single Windows 95 or NT Work Station”, dated Dec. 9, 1996.

Press release titled “Out-of-Control Web Sites Made More Effective, Profitable with New Web Management Tool; Mercury Interactive’s Astra Instantly Pinpoints Problems and Displays User-Behavior Patterns for Higher Quality Web Sites”, dated Oct. 9, 1996.

Press release titled “Tool For Webmasters Provides Open API For Java, C++ and Visual Basic Developers; Mercury Interactive Announces Astra—Industry’s First, Comprehensive, Graphical Web Management Tool”, dated Oct. 9, 1996.

Press release titled “Webmasters Eager for Site Management Solutions Rally Around New Web Management Tool; Mercury Interactive Ships Astra SiteManager, Developers Begin Leveraging Open API”, dated Dec. 9, 1996.

Product brochure for Astra SiteManager, dated 1997.

Product brochure for net.Analysis product of net.Genisis (undated).

Product brochure for CyberPilot Pro Product of NetsCarta (undated).

Article entitled “Web site management betas show major advances,” dated Nov. 11, 1996 in PC Week.

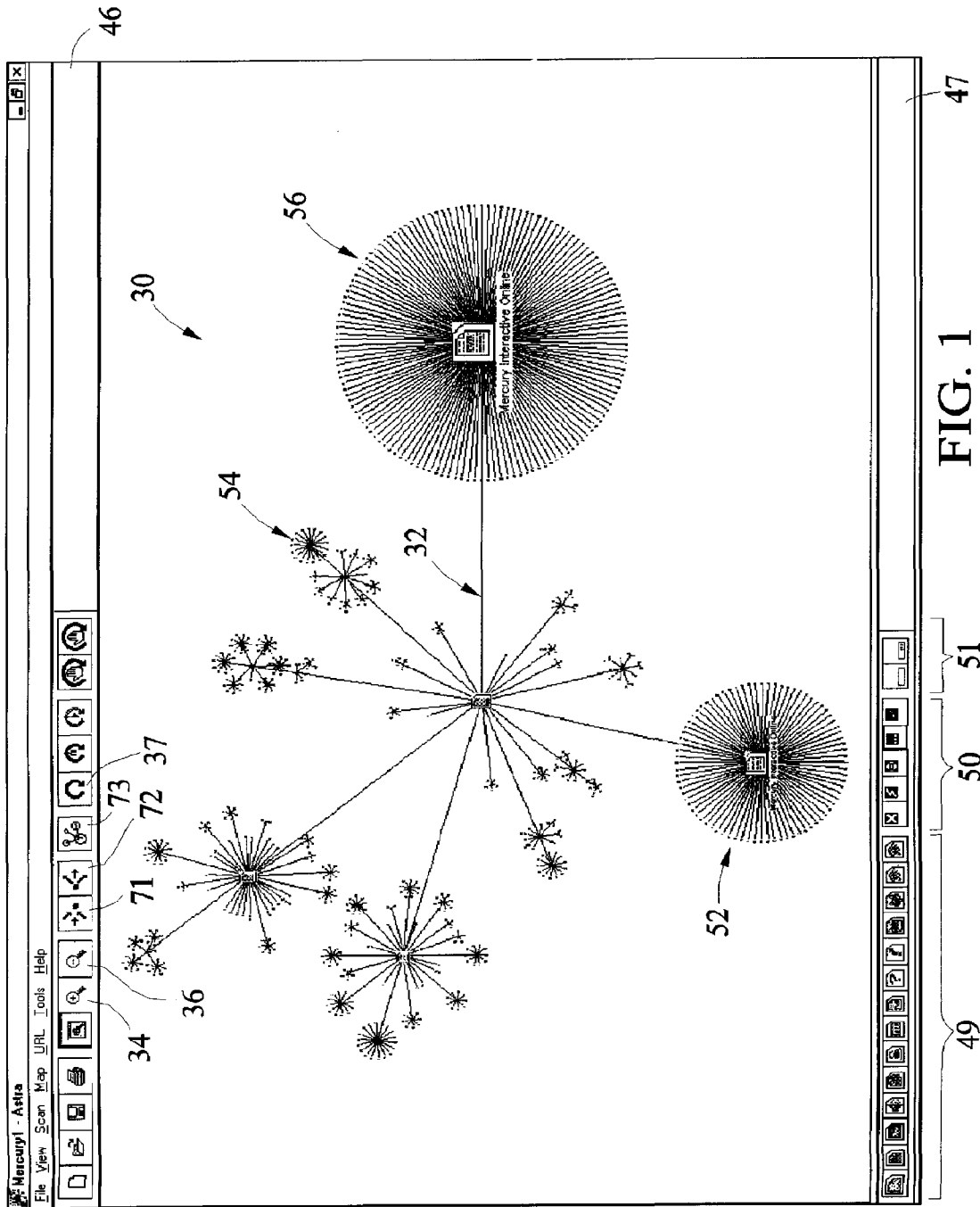


FIG. 1

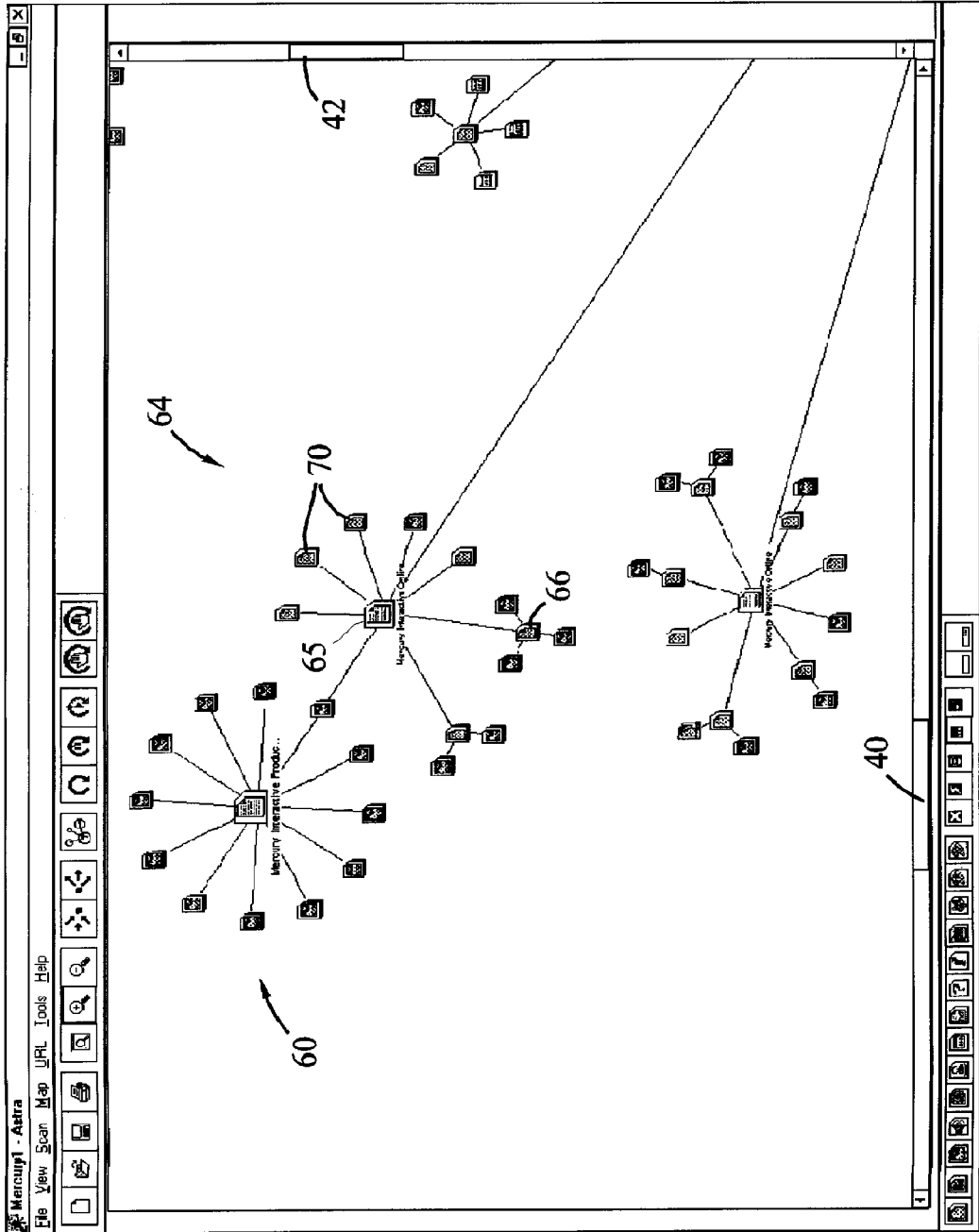


FIG. 2

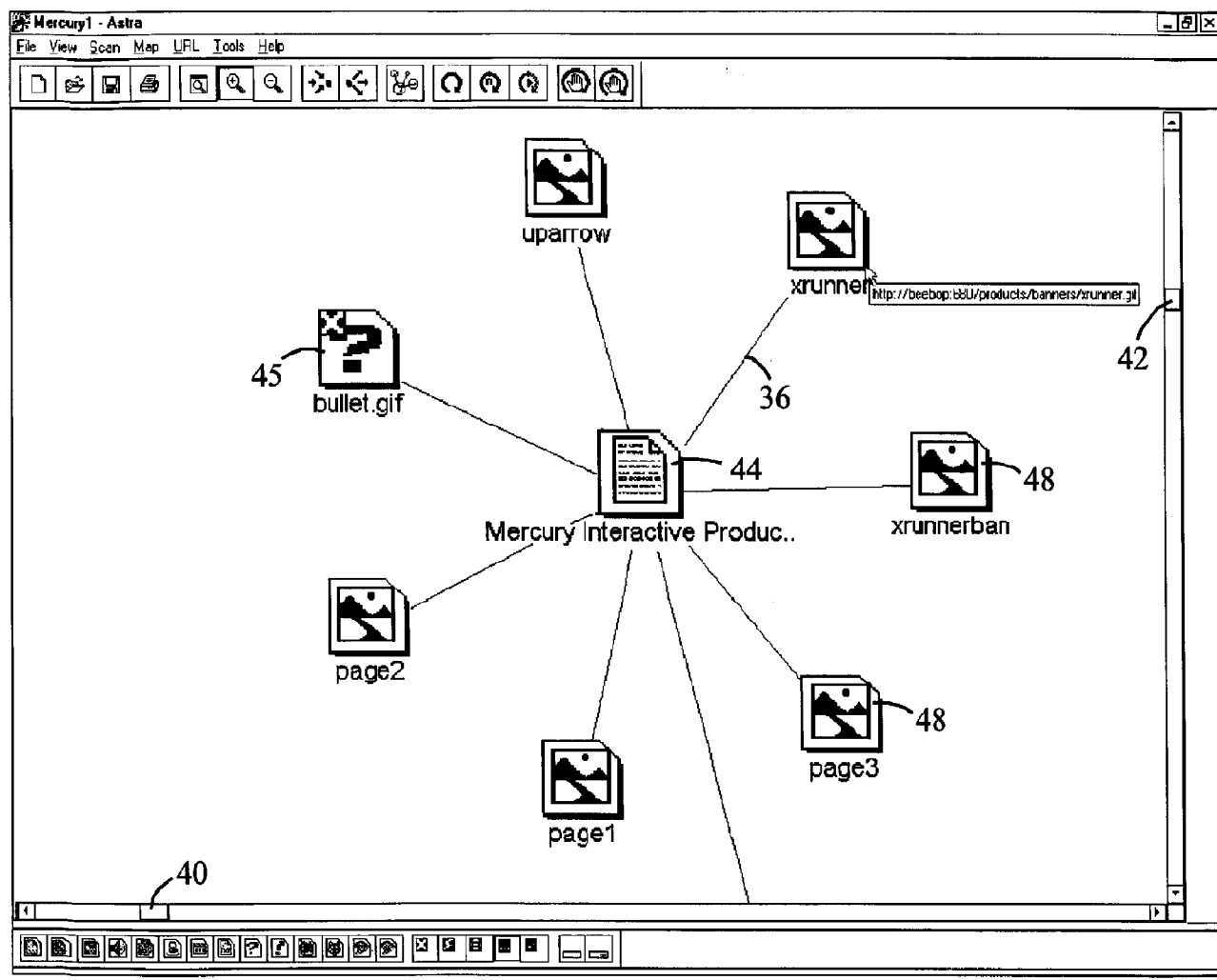


FIG. 3

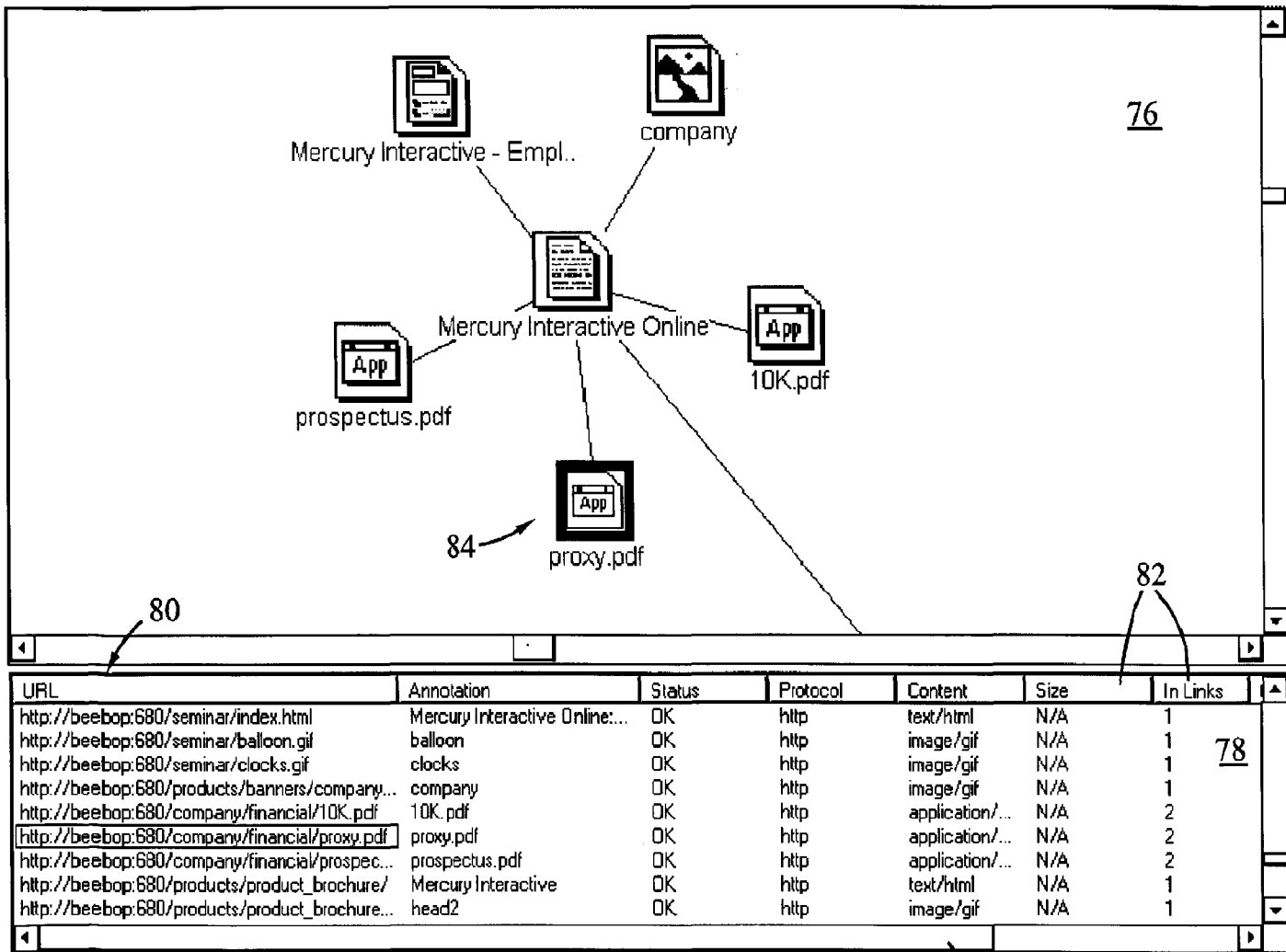


FIG. 4

77

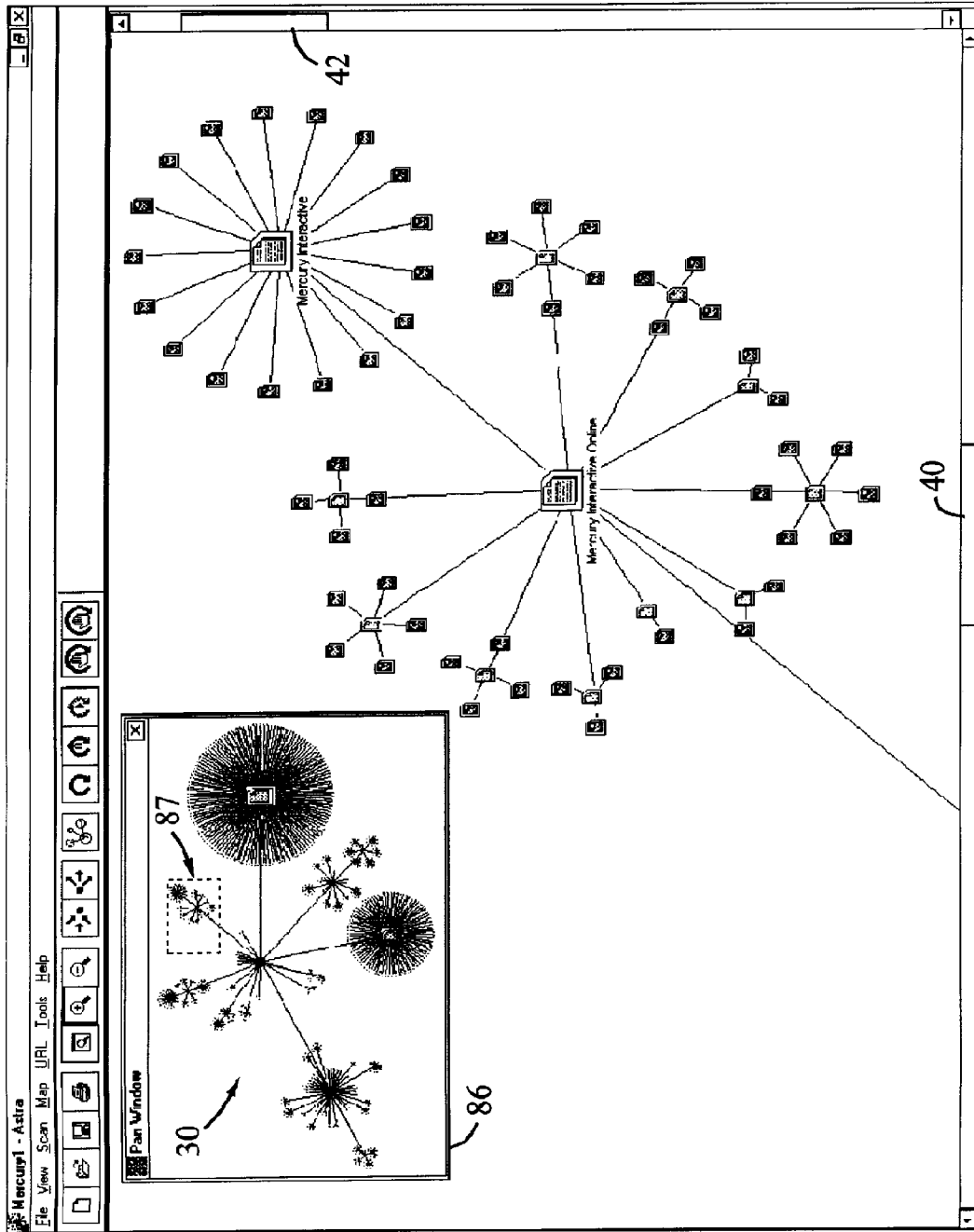


FIG. 5

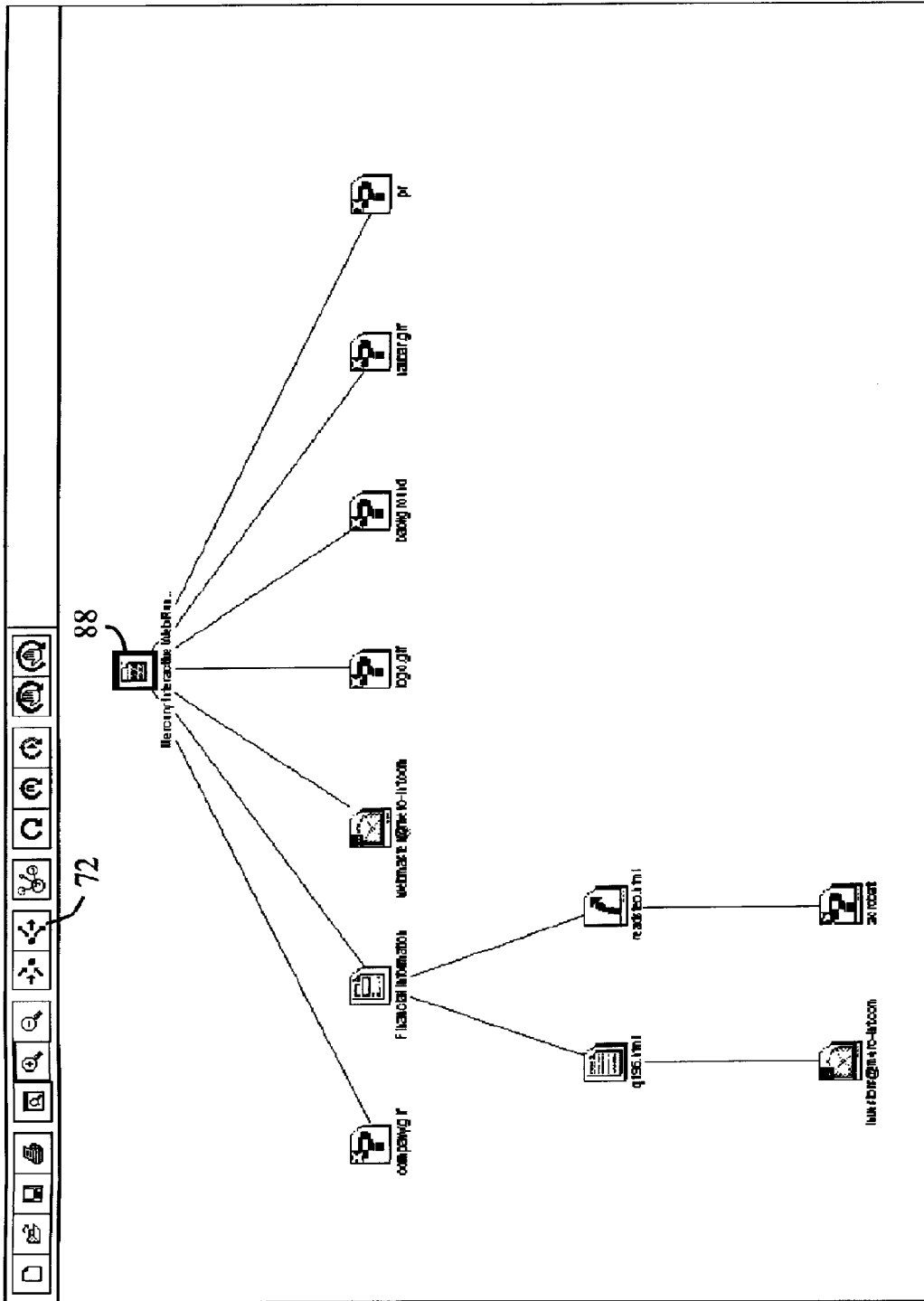


FIG. 6

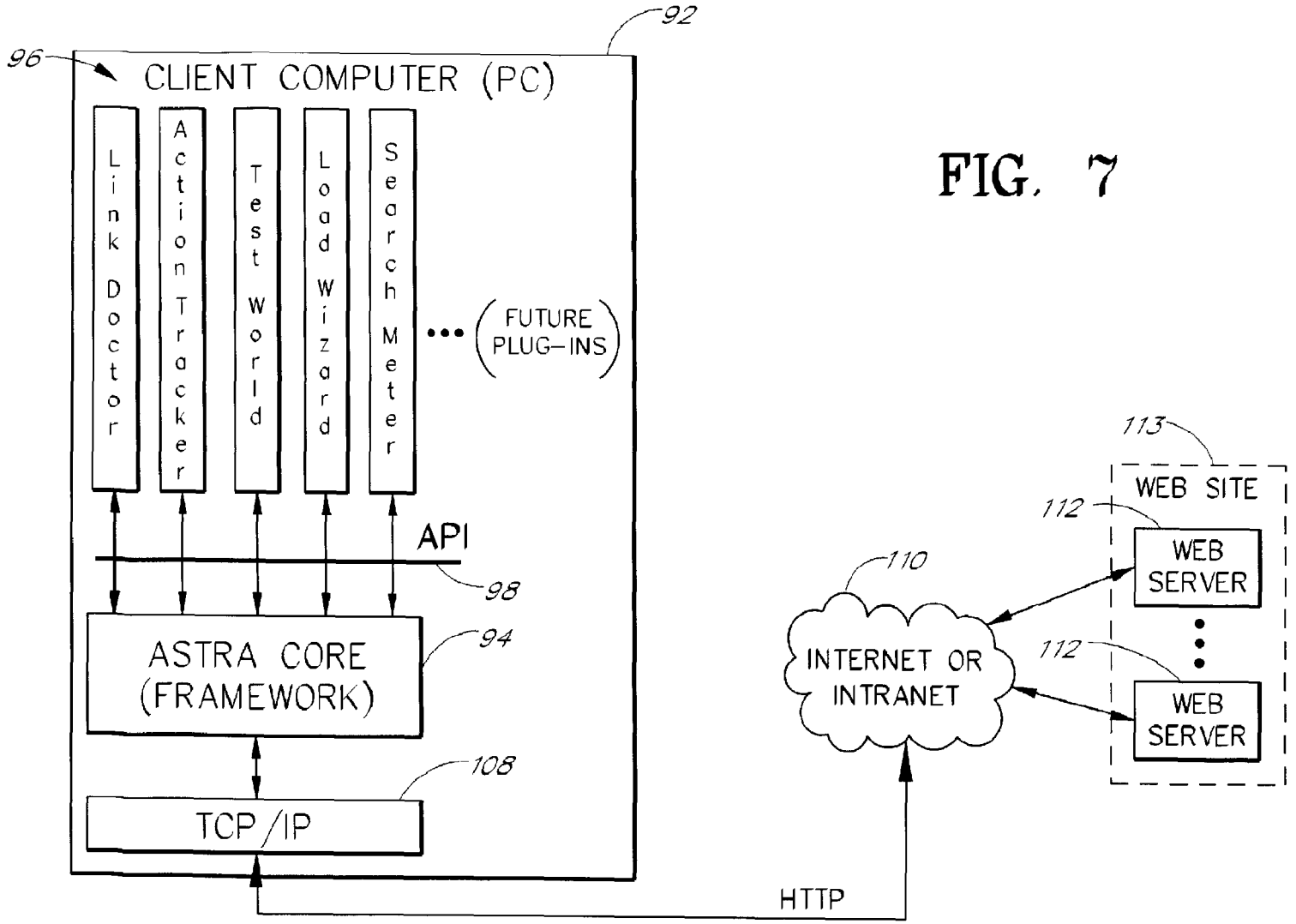


FIG. 7

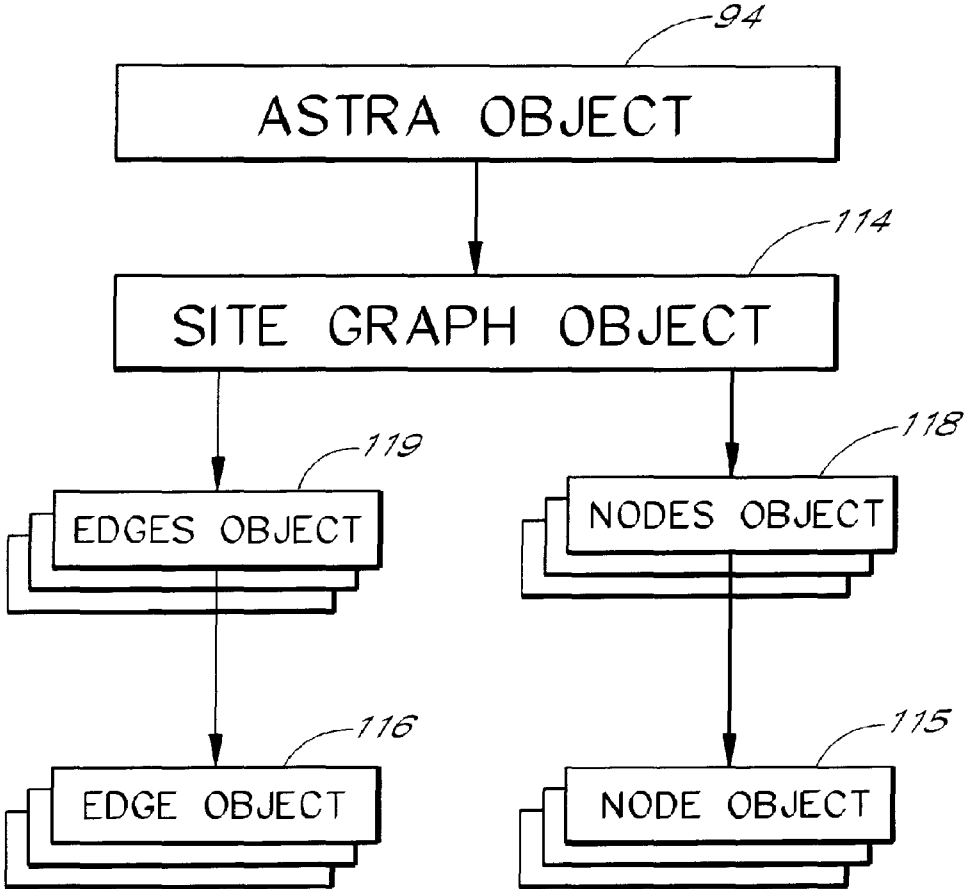


FIG. 8

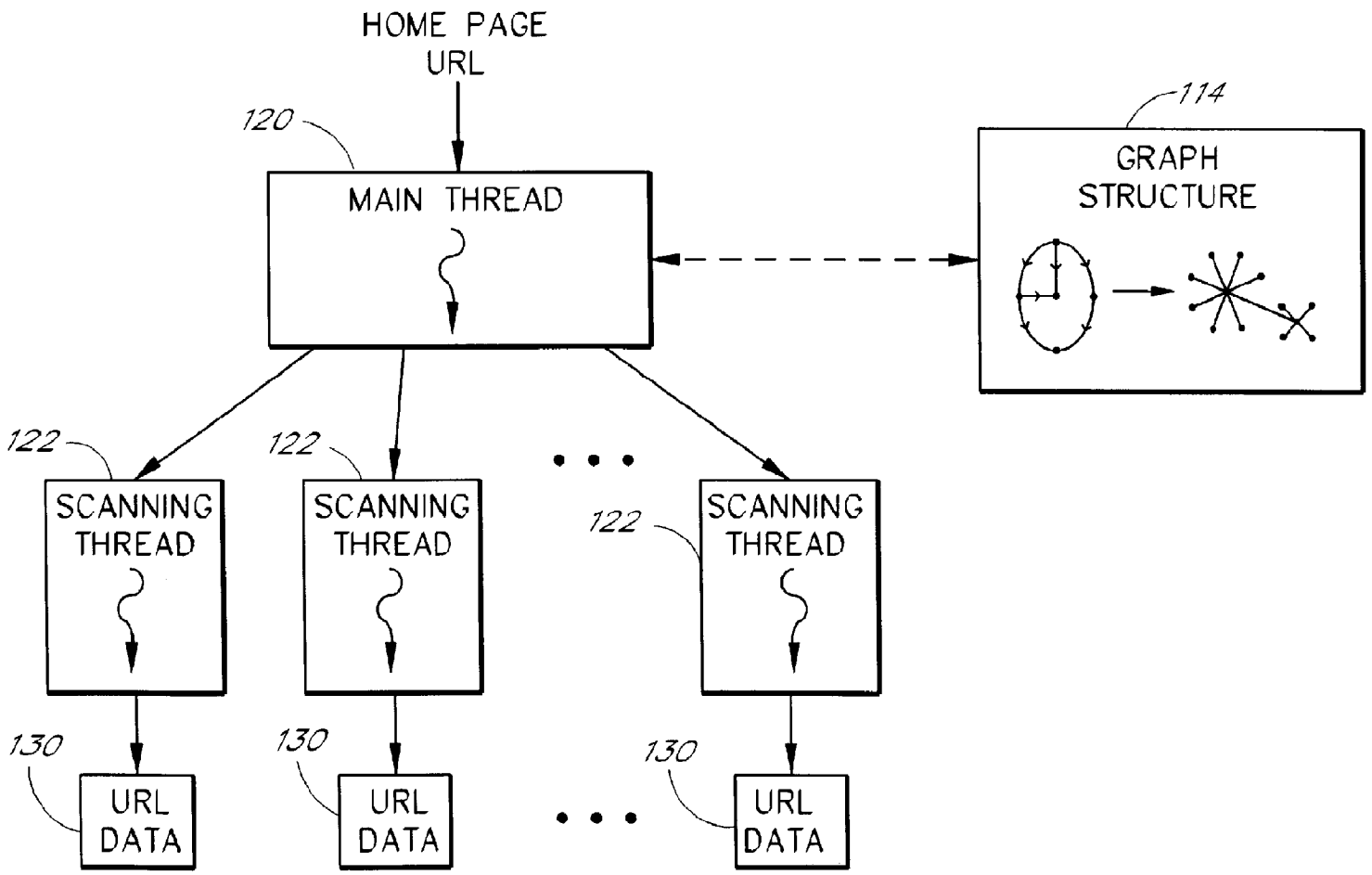


FIG. 9

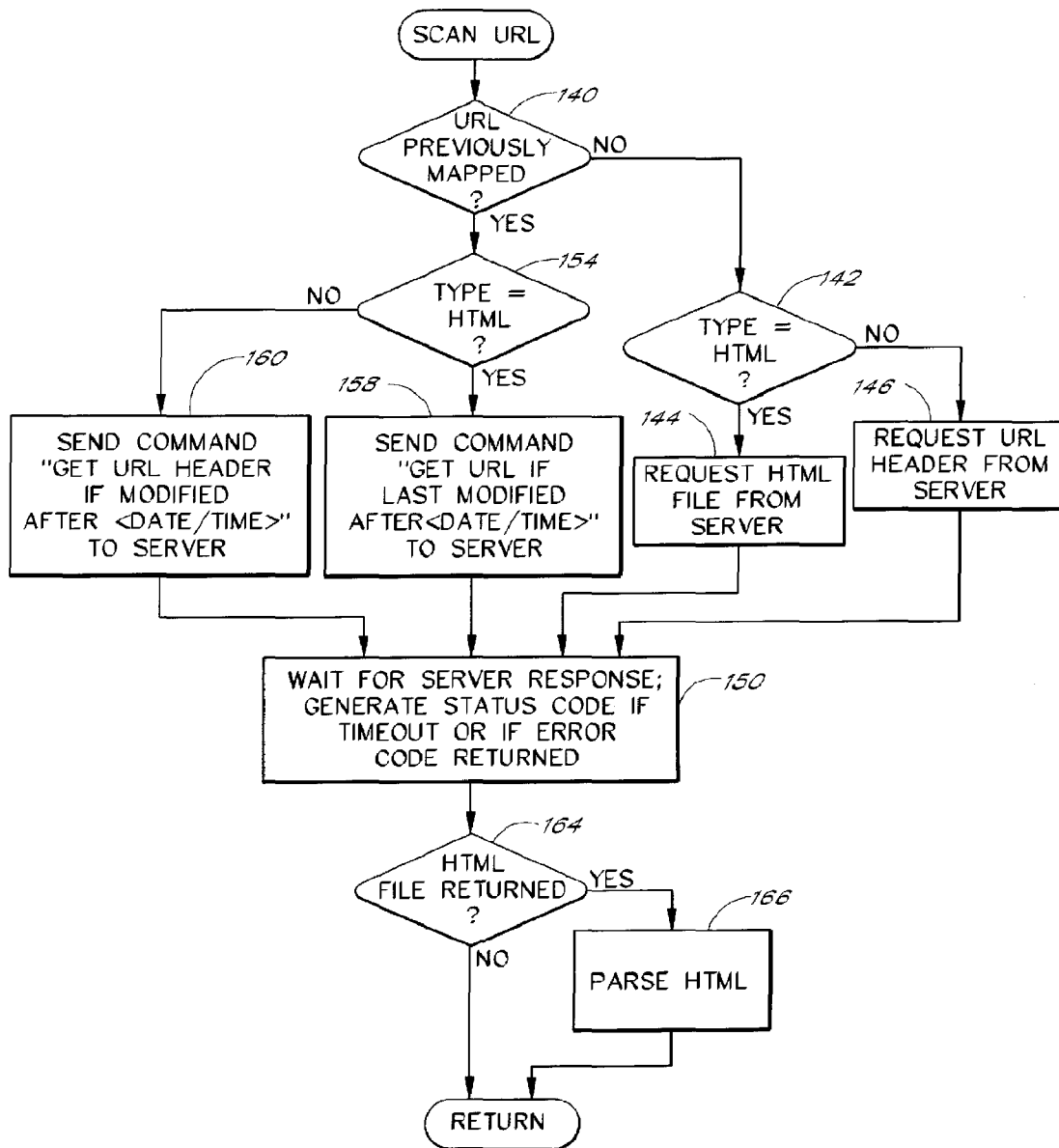


FIG. 10

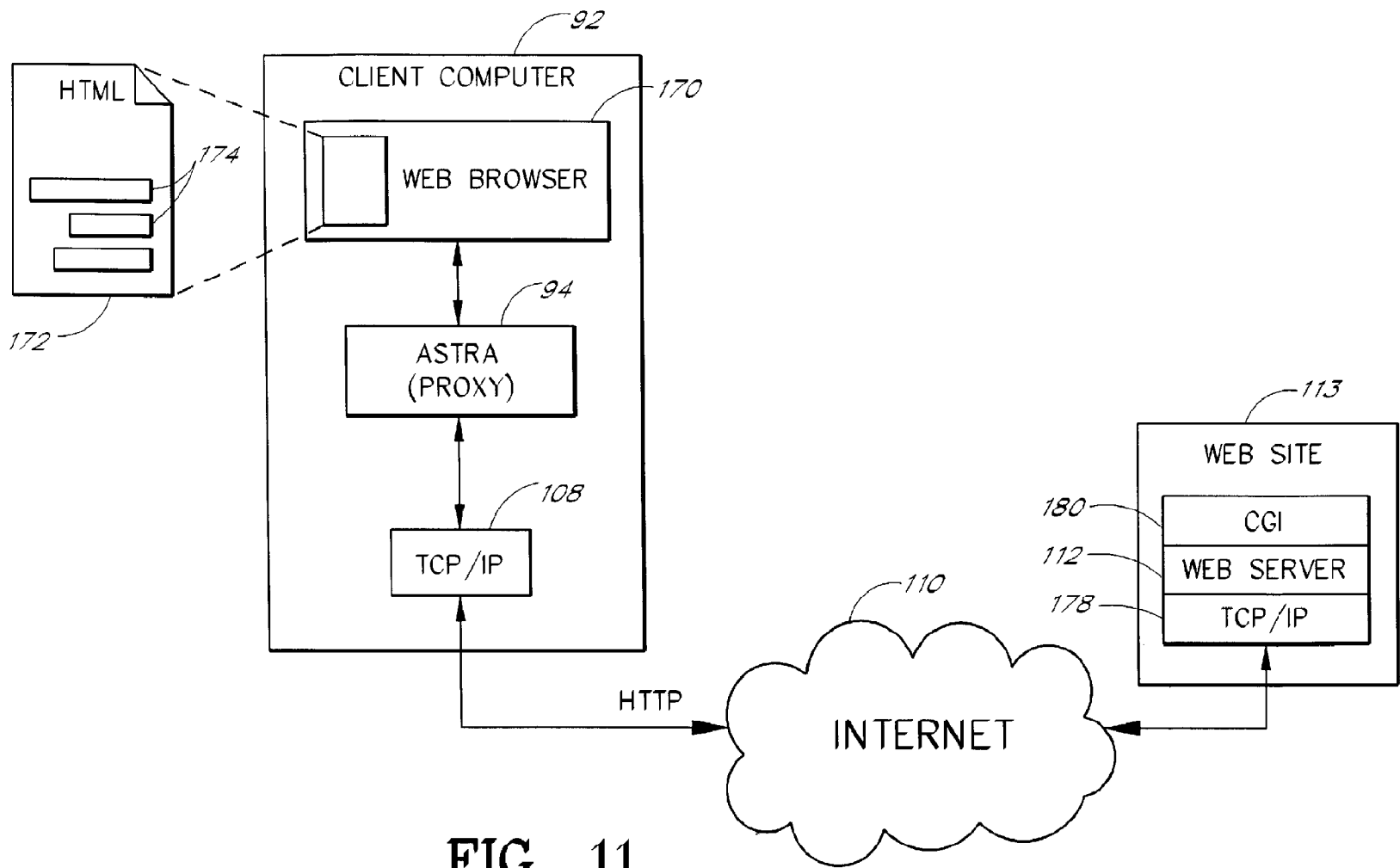


FIG. 11

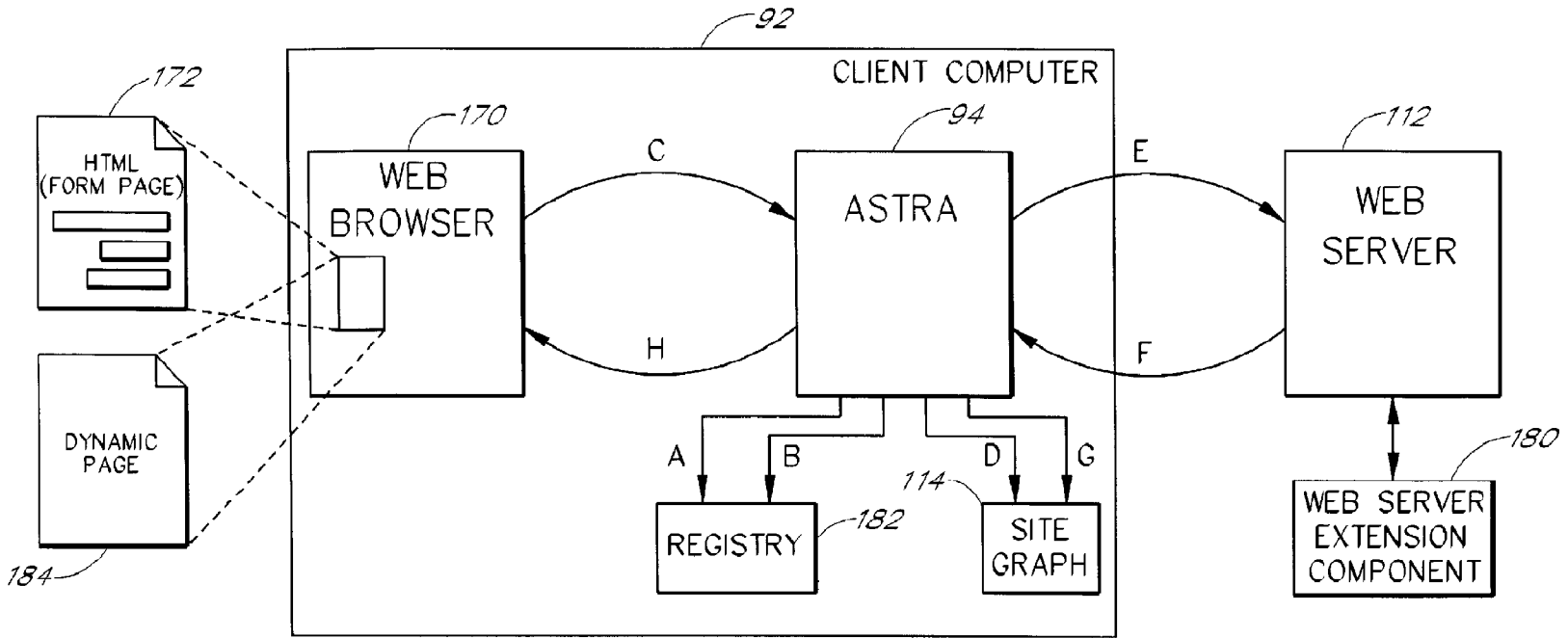


FIG. 12

- A. ASTRA MODIFIES BROWSER CONFIGURATION IN REGISTRY TO SET ASTRA AS PROXY
- B. ASTRA LAUNCHES BROWSER, AND THEN RESTORES ORIGINAL BROWSER CONFIGURATION WITHIN REGISTRY
- C. BROWSER PASSES HTTP MESSAGE TO ASTRA IN RESPONSE TO SUBMISSION OF FORM
- D. ASTRA EXTRACTS DATA SET AND STORES IN SITE GRAPH FOR FUTURE USE
- E. ASTRA FORWARDS HTTP MESSAGE TO WEB SERVER
- F. WEB SERVER RETURNS DYNAMICALLY-GENERATED WEB PAGE
- G. WEB SERVER PARSES DYNAMICALLY-GENERATED PAGE AND UPDATES SITE GRAPH
- H. ASTRA FORWARDS PAGE TO BROWSER TO CREATE IMPRESSION OF REGULAR BROWSING

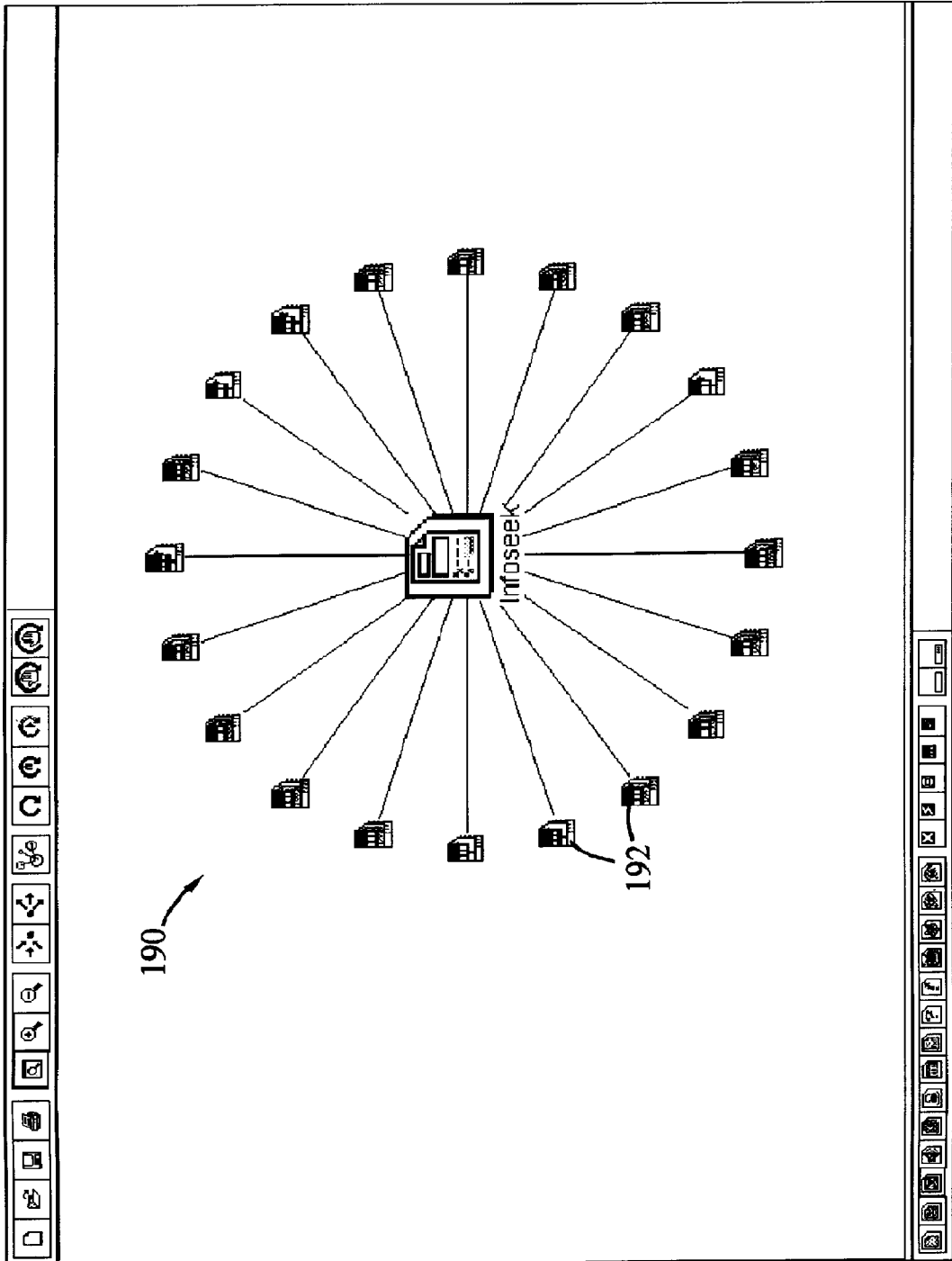


FIG. 13

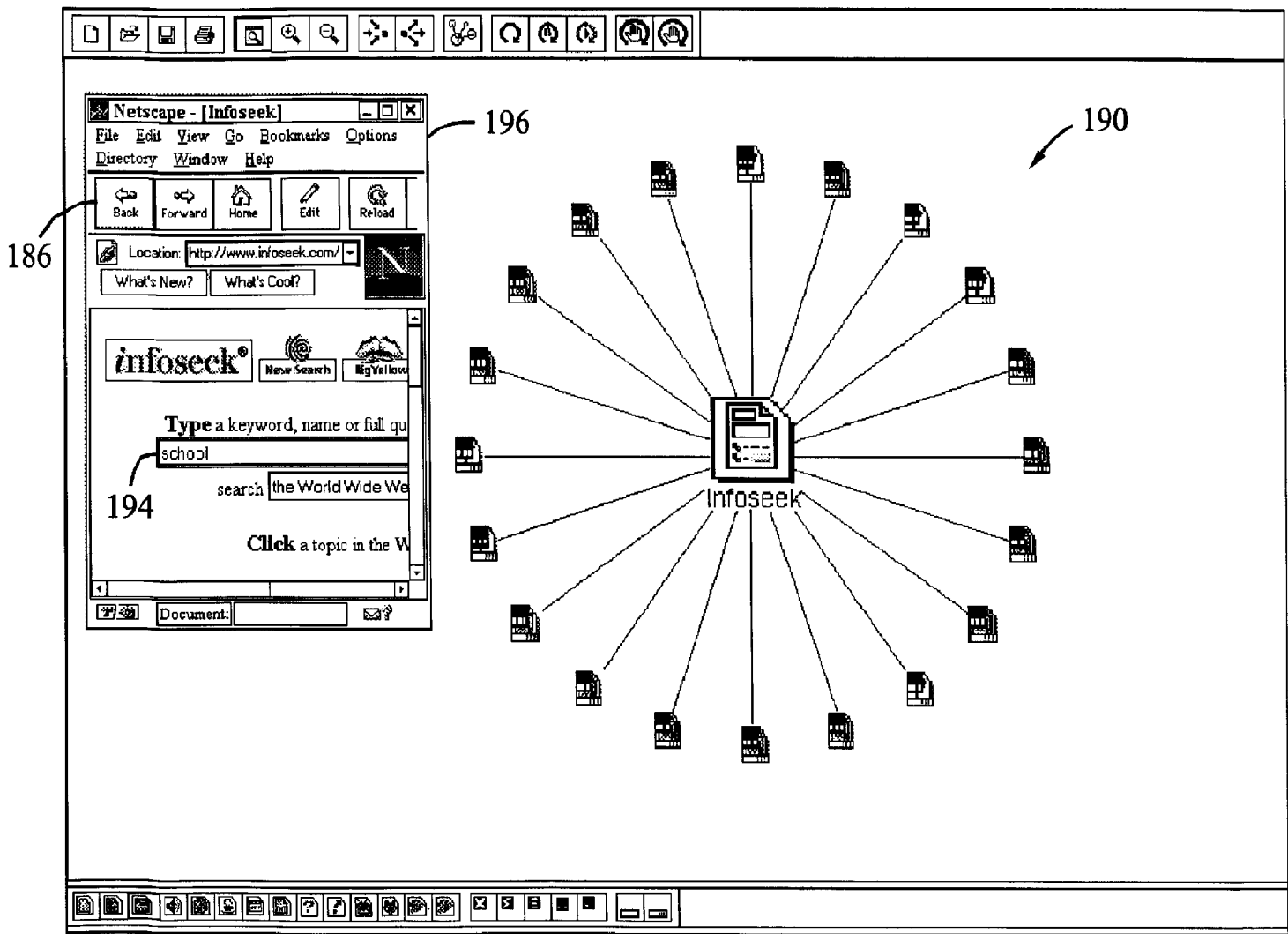


FIG. 14

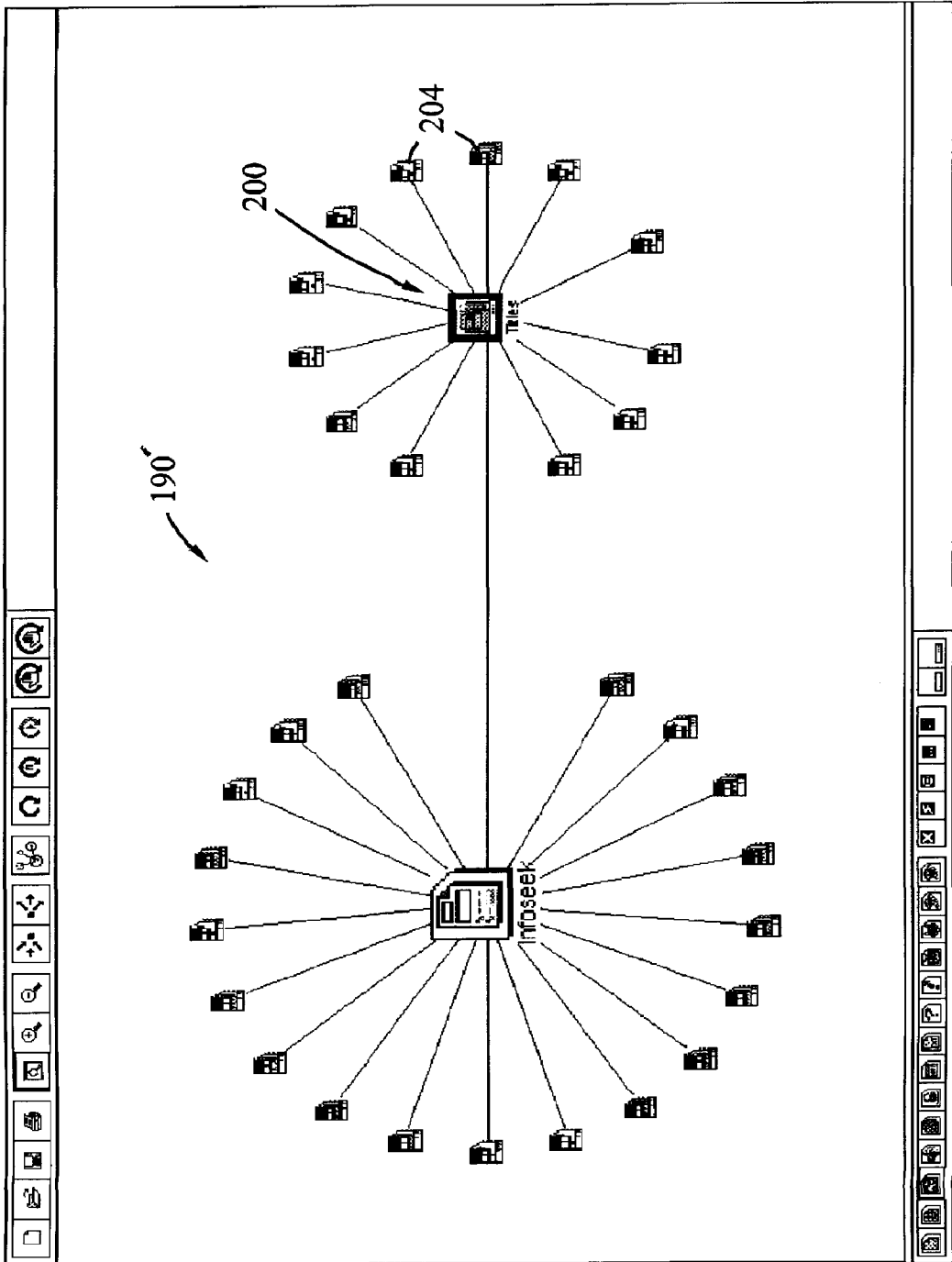


FIG. 15

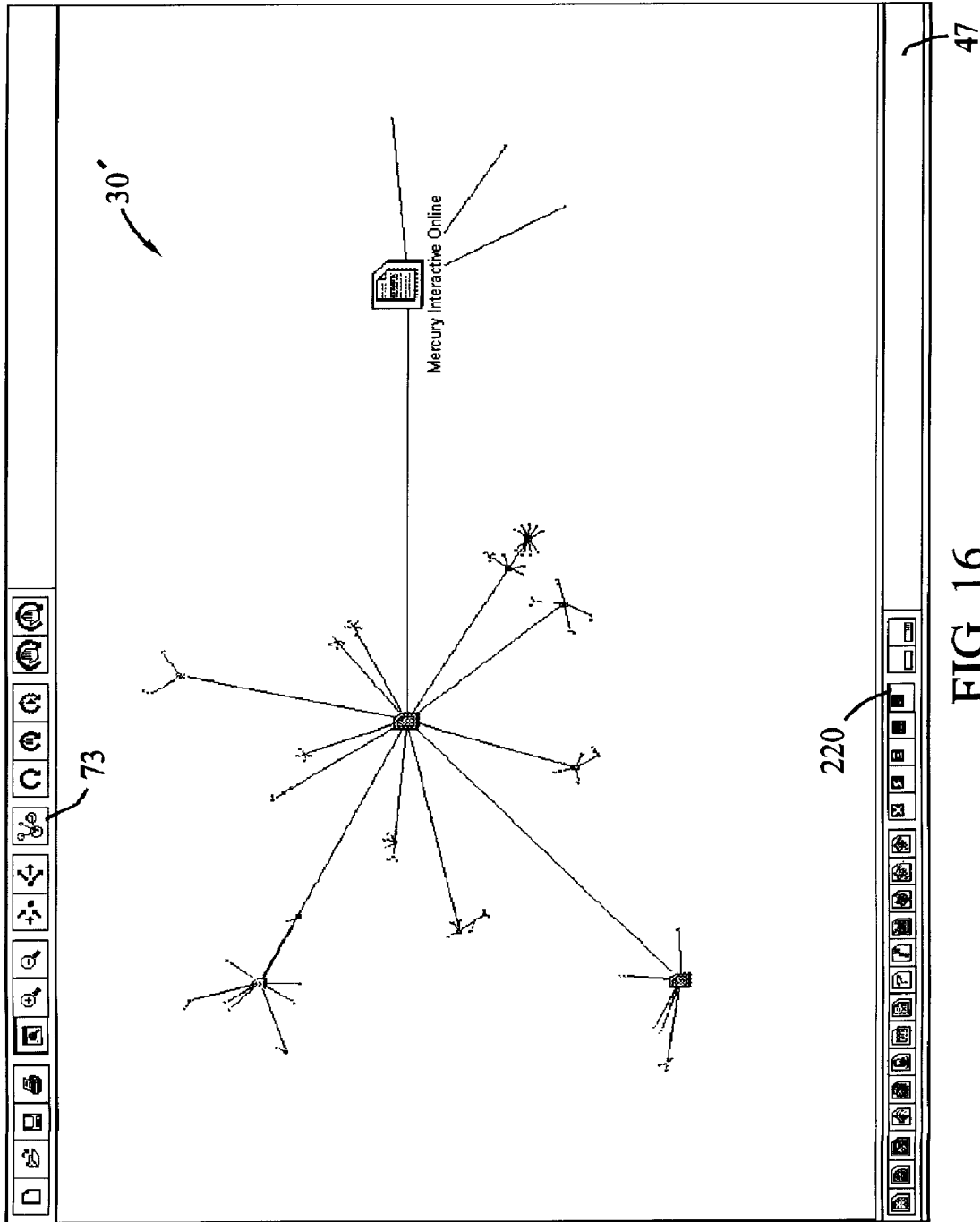
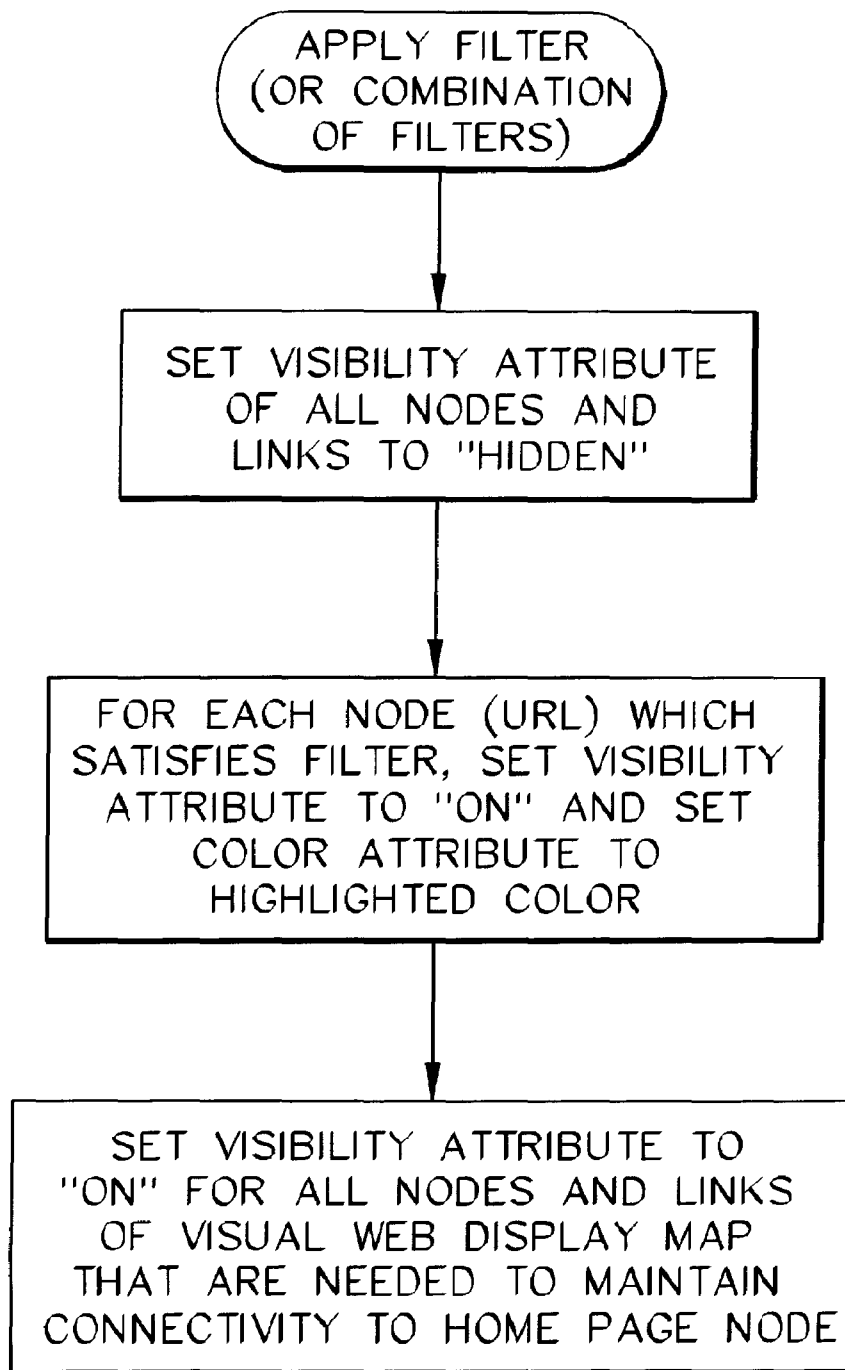


FIG. 16

**FIG. 17**

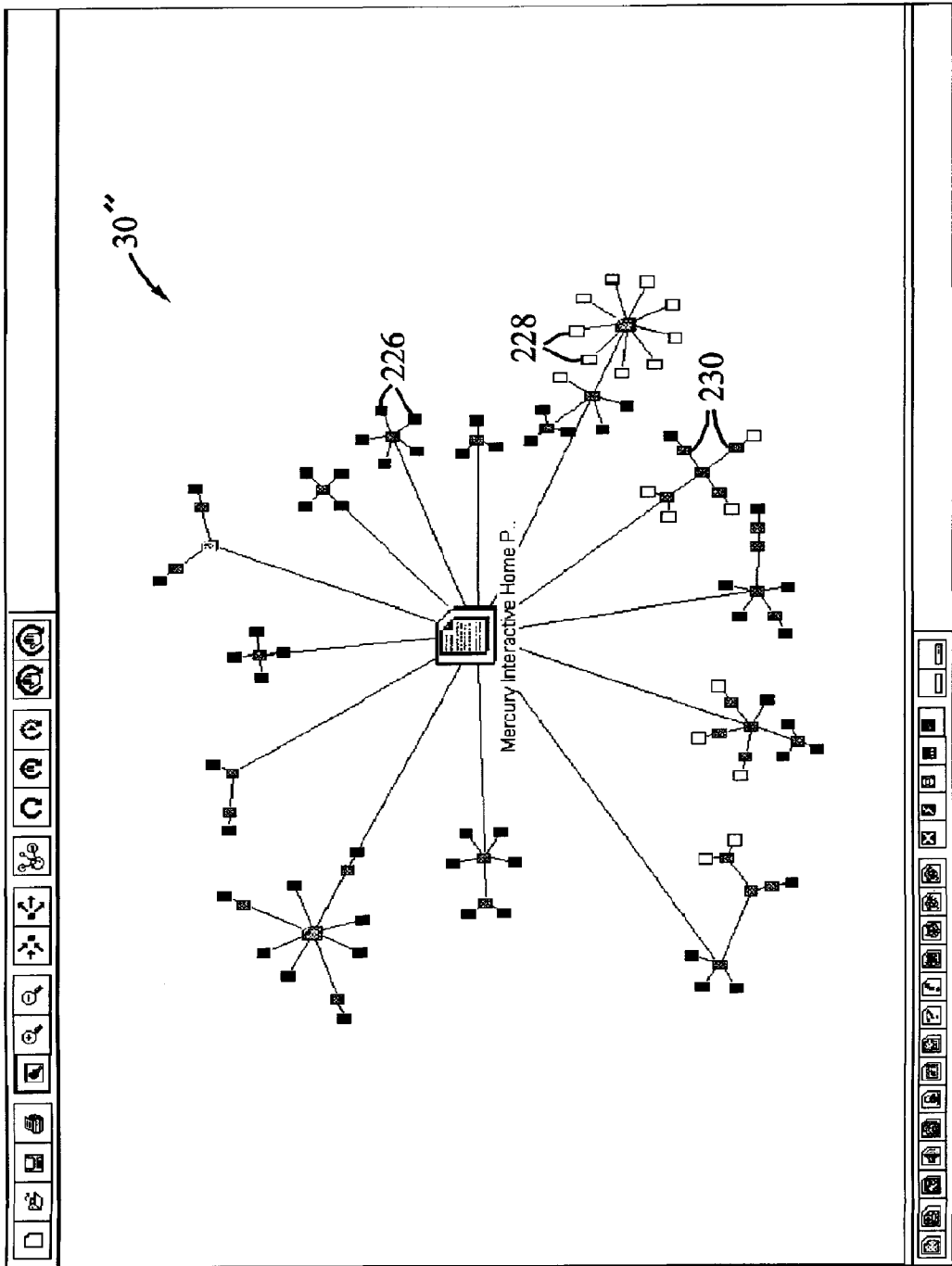


FIG. 18

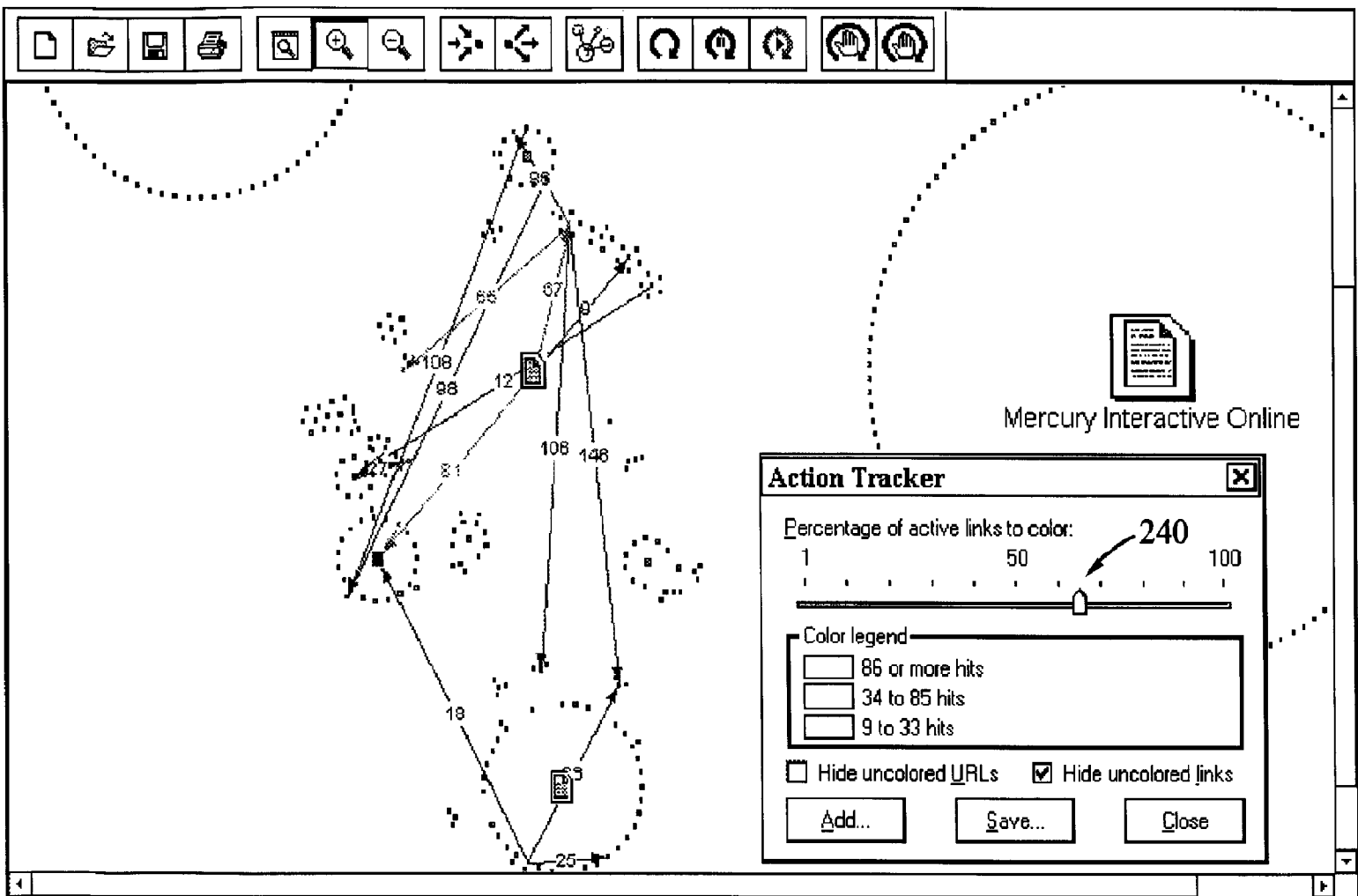


FIG. 19

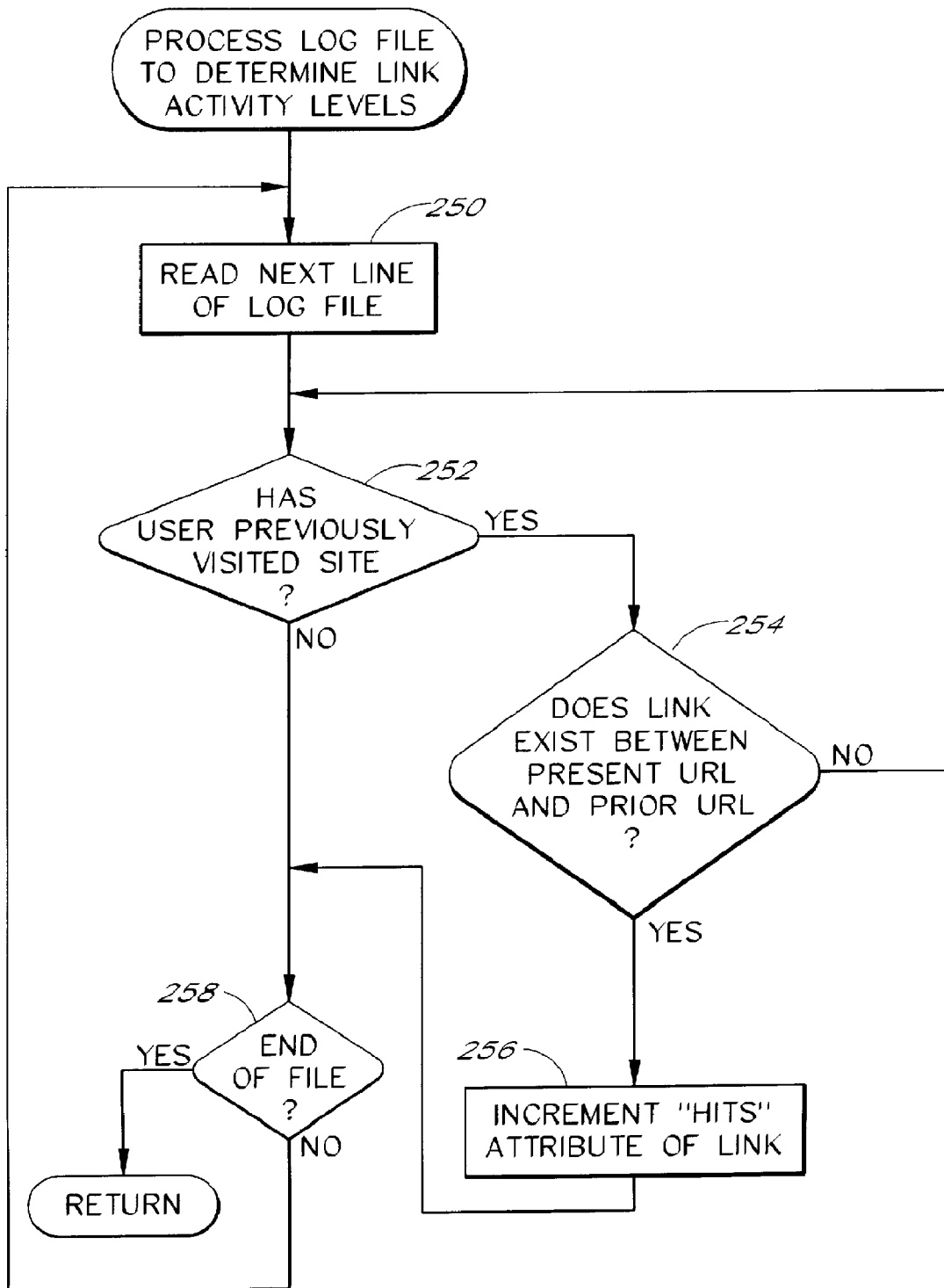


FIG. 20

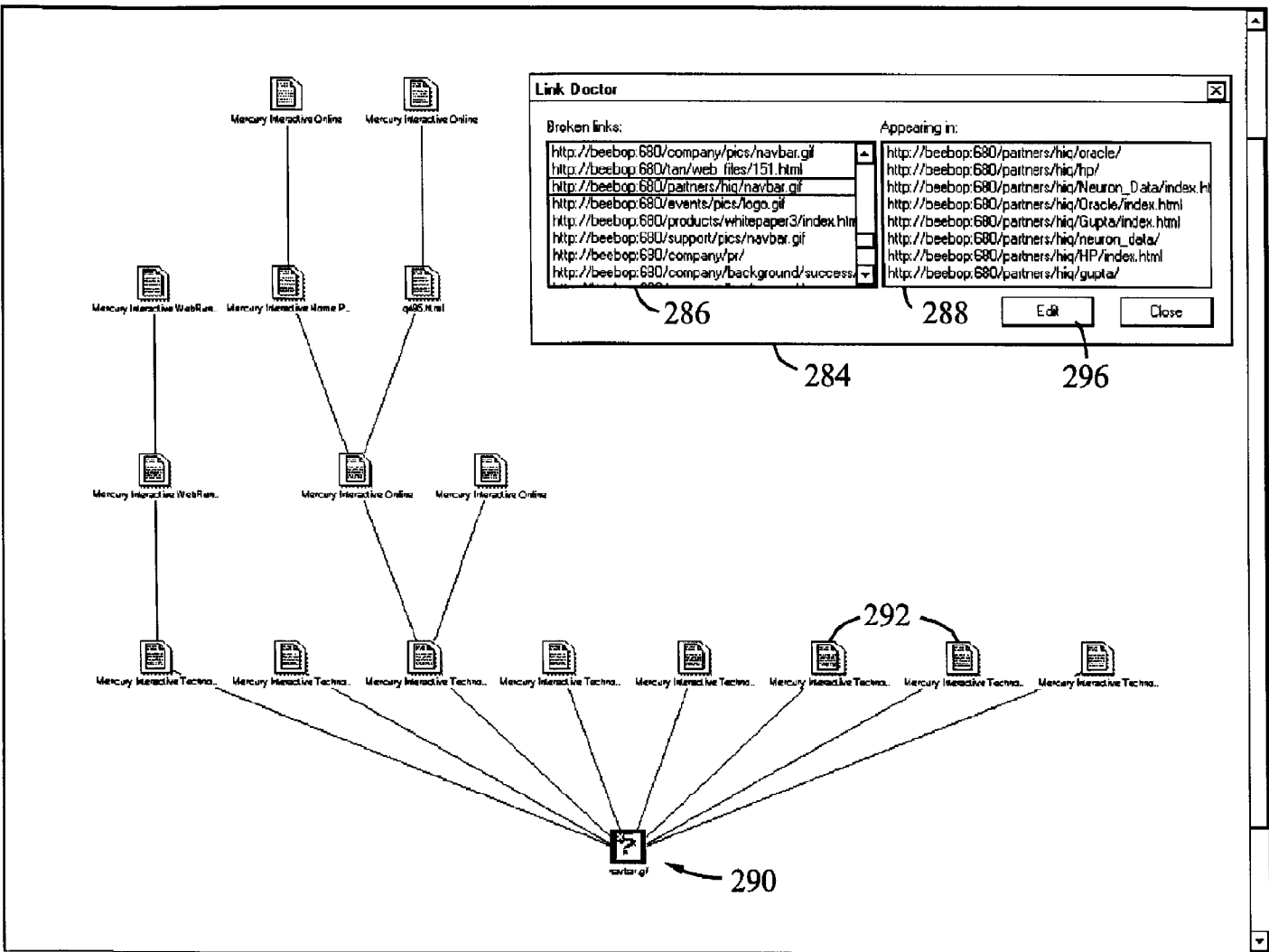


FIG. 22

FIG. 23

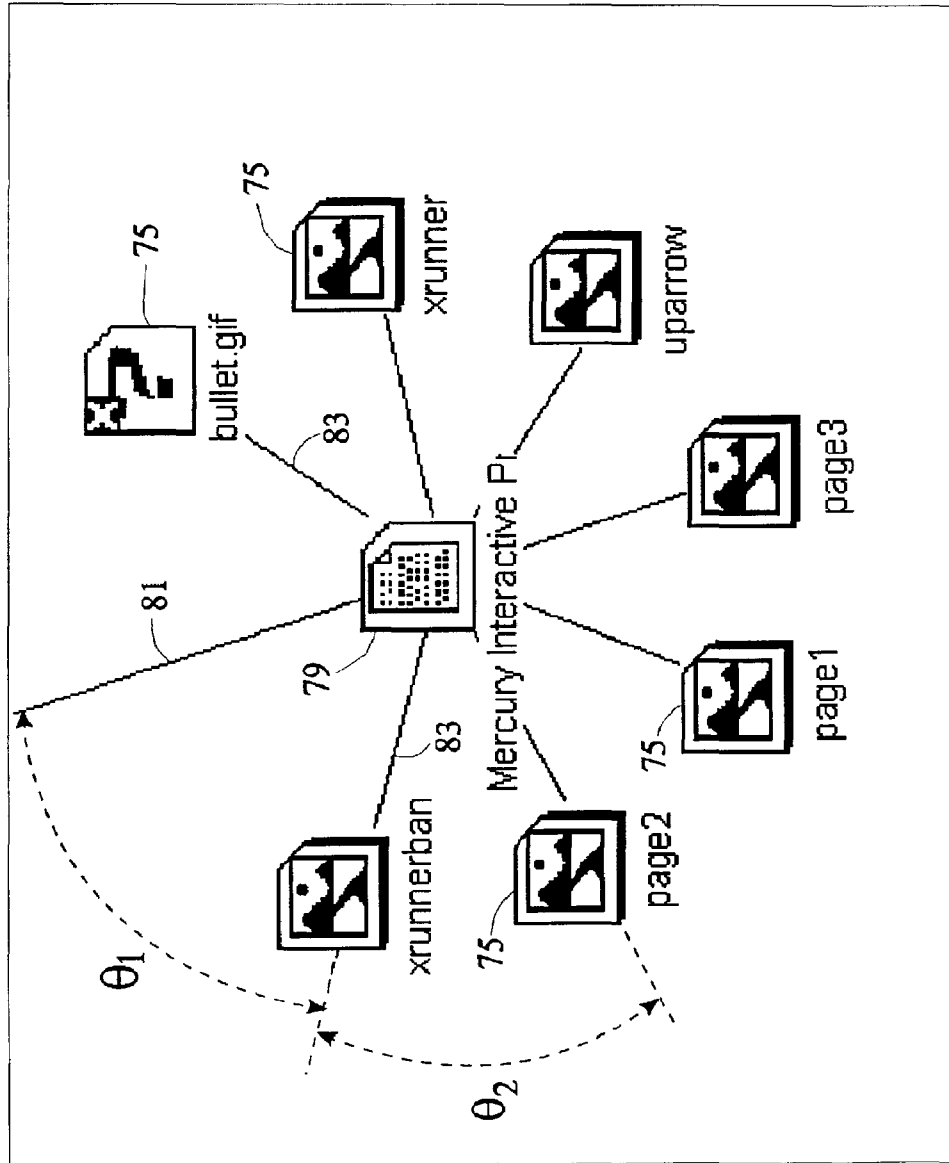
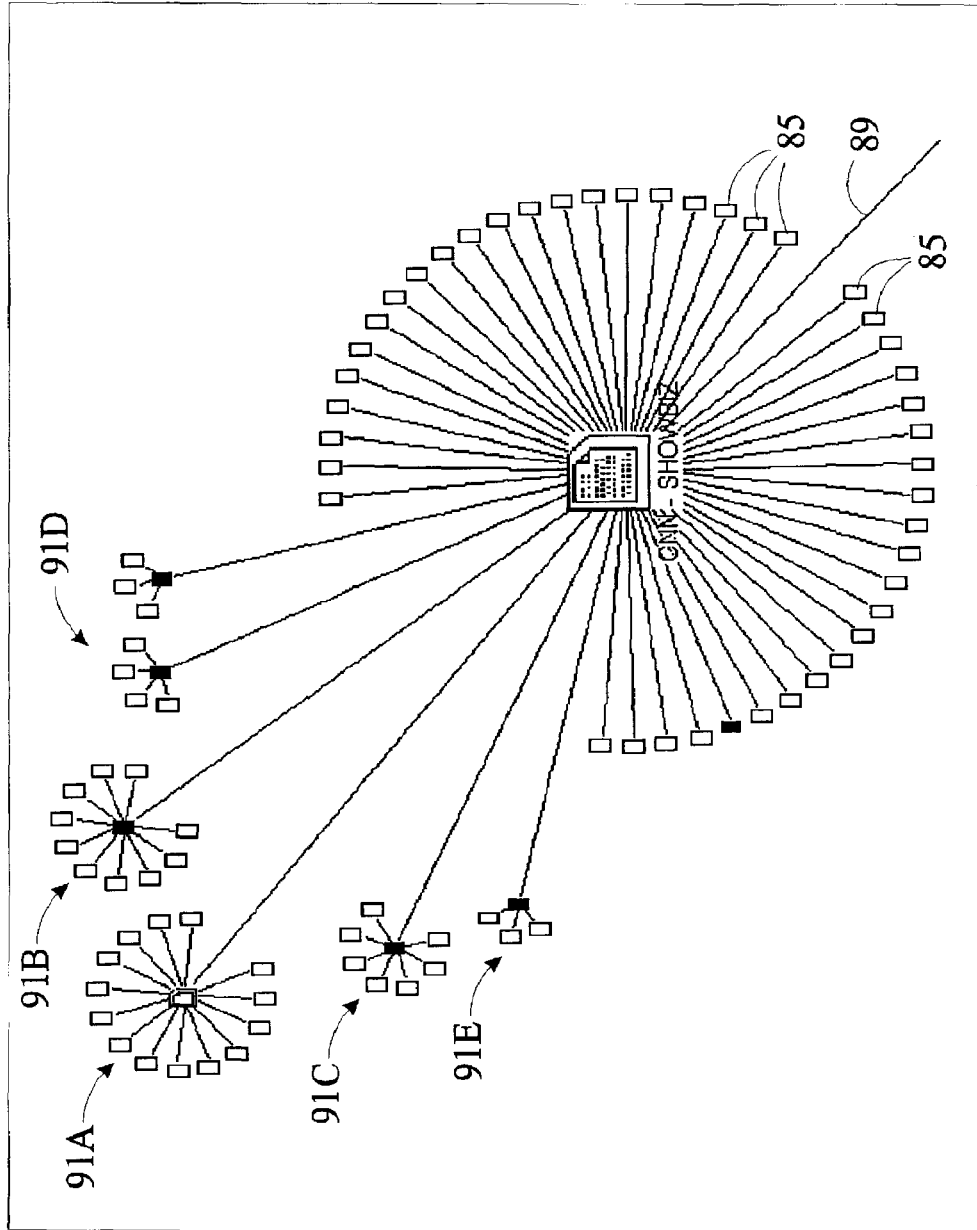


FIG. 24



VISUALIZATION OF WEB SITES AND HIERARCHICAL DATA STRUCTURES

PRIORITY CLAIM

This application claims the benefit of U.S. Provisional Application No. 60/028,474 titled SOFTWARE SYSTEM AND ASSOCIATED METHODS FOR FACILITATING THE ANALYSIS AND MANAGEMENT OF WEB SITES, filed Oct. 15, 1996, which is hereby incorporated by reference.

MICROFICHE APPENDIX

This specification includes a microfiche appendix consisting of 1 sheet with 45 frames) which contains a partial source code listing and an API (application program interface) listing of a preferred embodiment of the invention, as Appendices A and B, respectively. These materials form part of the disclosure of the specification.

FIELD OF THE INVENTION

The present invention relates generally to database management, analysis and visualization tools. More particularly, the present invention relates to software tools for facilitating the management and analysis of World Wide Web sites and other types of database systems which utilize hyperlinks to facilitate user navigation.

BACKGROUND OF THE INVENTION

With the increasing popularity and complexity of Internet and intranet applications, the task of managing Web site content and maintaining Web site effectiveness has become increasingly difficult. Company Webmasters and business managers are routinely faced with a wide array of burdensome tasks, including, for example, the identification and repair of large numbers of broken links (i.e., links to missing URLs), the monitoring and organization of large volumes of diverse, continuously-changing Web site content, and the detection and management of congested links. These problems are particularly troublesome for companies that rely on their respective Web sites to provide mission-critical information and services to customers and business partners.

Several software companies have developed software products which address some of these problems by generating graphical maps of Web site content and providing tools for navigating and managing the content displayed within the maps. Examples of such software tools include WebMapper™ from Netcarta Corporation and WebAnalyzer™ from InContext Corporation. Unfortunately, the graphical site maps generated by these products tend to be difficult to navigate, and fail to convey much of the information needed by Webmasters to effectively manage complex Web sites. As a result, many companies continue to resort to the burdensome task of manually generating large, paper-based maps of their Web sites. In addition, many of these products are only capable of mapping certain types of Web pages, and do not provide the types of analysis tools needed by Webmasters to evaluate the performance and effectiveness of Web sites.

The present invention addresses these and other limitations in existing products and technologies.

SUMMARY OF THE INVENTIVE FEATURES

In accordance with the present invention, a software package ("Web site analysis program") is provided which

includes a variety of features for facilitating the management and analysis of Web sites. In the preferred embodiment, the program runs on a network-connected PC under the Windows® 95 or Windows® NT operating system, and utilizes the standard protocols and conventions of the World Wide Web ("Web"). In other embodiments, the program may be adapted to provide for the analysis of other types of hypertextual-content sites, including sites based on non-standard protocols.

In the preferred embodiment, the program includes Web site scanning routines which use conventional webcrawling techniques to gather information about the content objects (HTML documents, GIF files, etc.) and links of a Web site via a network connection. Mapping routines of the program in-turn use this information to generate, on the computer's display screen, a graphical site map that shows the overall architecture (i.e., the structural arrangement of content objects and links) of the Web site. A user interface of the program allows the user to perform actions such as initiate and pause the scanning/mapping of a Web site, zoom in and out on portions of a site map, apply content filters to the site map to filter out content objects of specific types, and save and retrieve maps to/from disk. A map comparison tool allows the user to generate a comparison map which highlights changes that have been made to the Web site since a previous mapping of the site.

In accordance with one aspect of the invention, the Web site analysis program implements a map generation method which greatly facilitates the visualization by the user of the overall architecture of the Web site, and allows the user to navigate the map in an intuitive manner to explore the content of the Web site. To generate the site map, a structural representation of the Web site (specifying the actual arrangement of content objects and links) is initially reduced, for purposes of generating the site map, to a hierarchical tree representation in which each content object of the Web site is represented as a node of the tree. A recursive layout method is then applied which uses the parent-child node relationships, as such relationships exist within the tree, to spatially position the nodes (represented as respective icons within the map) on the display screen such that children nodes are positioned around and connected to their respective immediate parents. (This layout method can also be used to display other types of hierarchical data structures, such as the tree structure of a conventional file system.) The result is a map which comprises a hierarchical arrangement of parent-child node (icon) clusters in which parent-child relationships are immediately apparent.

As part of the layout method, the relative sizes of the node icons are preferably adjusted such that nodes with relatively large numbers of outgoing links have a relatively large icon size, and thus stand out in the map. In addition, the node and link display sizes are automatically adjusted such that the entire map is displayed on the display screen, regardless of the size of the Web site. As the user zooms in on portions of the map, additional details of the Web site's content objects are automatically revealed within the map.

In accordance with another aspect of the invention, the Web site analysis program is based on an extensible architecture that allows software components to be added that make extensive use of the program's mapping functionality. Specifically, the architecture includes an API (application program interface) which includes API procedures ("methods") that allow other applications ("plug-ins") to, among other things, manipulate the display attributes of the nodes and links within a site map. Using these methods, a plug-in application can be added which dynamically super-

imposes data onto the site map by, for example, selectively modifying display colors of nodes and links, selectively hiding nodes and links, and/or attaching alphanumeric annotations to the nodes and links. The API also includes methods for allowing plug-in components to access Web site data (both during and following the Web site scanning process) retrieved by the scanning routines, and for adding menu commands to the user interface of the main program.

In accordance with another aspect of the invention, software routines (preferably implemented within a plug-in application) are provided for processing a Web site's server access log file to generate Web site usage data, and for displaying the usage data on a site map. This usage data may, for example, be in the form of the number of "hits" per link, the number of Web site exit events per node, or the navigation paths taken by specific users ("visitors"). This usage data is preferably generated by processing the entries within the log file on a per-visitor basis to determine the probable navigation path taken by each respective visitor to the Web site. (Standard-format access log files which record each access to any page of the Web site are typically maintained by conventional Web servers.) In a preferred implementation, the usage data is then superimposed onto the site map (using the API methods) using different node and link display colors to represent different respective levels of user activity. Using this feature, Webmasters can readily detect common "problem areas" such as congested links and popular Web site exit points. In addition, by looking at individual navigation paths on a per-visitor basis, Webmasters can identify popular navigation paths taken by visitors to the site.

In accordance with yet another aspect of the invention, the Web site analysis program includes software routines and associated user interface controls for automatically scanning and mapping dynamically-generated Web pages, such as Web pages generated "on-the-fly" in response to user-specified database queries. This feature generally involves the two-step process of capturing and recording a dataset manually entered by the user into an embedded form of a Web page (such as a page of a previously-mapped Web site), and then automatically resubmitting the dataset (within the form) when the Web site is later re-scanned. As will be appreciated, this feature of the invention can also be applied to conventional Internet search engines.

To effectuate the capture of one or more datasets in the preferred implementation, the user initiates a capture session from the user interface; this causes a standard Web browser to be launched and temporarily configured to use the Web site analysis program as an HTTP-level proxy to communicate with Web sites. Thereafter, until the capture session is terminated by the user, any pages retrieved with the browser, and any forms (datasets) submitted from the browser, are automatically recorded by the Web site analysis program into the site map. When the site map is subsequently updated (using an "automatic update" option of the user interface), the scanning routines automatically re-enter the captured datasets into the corresponding forms and recreate the form submissions. The dynamically-generated Web pages returned in response to these automatic form submissions are then added to the updated site map as respective nodes. A related aspect of the invention involves the associated method of locally capturing the output of the Web browser to generate a sequence that can subsequently be used to automatically evaluate a Web site.

BRIEF DESCRIPTION OF THE DRAWINGS

The various features of the invention will now be described in greater detail with reference to the drawings of

a preferred software package referred to as the Astra™ SiteManager™ Web site analysis tool ("Astra"), its screen displays, and various related components. In these drawings, reference numbers are re-used, where appropriate, to indicate a correspondence between referenced items.

FIG. 1 is a screen display which illustrates an example Web site map generated by Astra, and which illustrates the menu, tool and filter bars of the Astra graphical user interface.

FIGS. 2 and 3 are screen displays which illustrate respective zoomed-in views of the site map of FIG. 1.

FIG. 4 is a screen display which illustrates a split-screen display mode, wherein a graphical representation of a Web site is displayed in an upper window and a textual representation of the Web site is displayed in a lower window.

FIG. 5 is a screen display which illustrates a navigational aid of the Astra graphical user interface.

FIG. 6 is a screen display illustrating a feature which allows a user to selectively view the outbound links of URI in a hierarchical display format.

FIG. 7 is a block diagram which illustrates the general architecture of Astra, which is shown in the context of a client computer communicating with a Web site.

FIG. 8 illustrates the object model used by Astra.

FIG. 9 illustrates a multi-threaded process used by Astra for scanning and mapping Web sites.

FIG. 10 illustrates the general decision process used by Astra to scan a URL.

FIG. 11 is a block diagram which illustrates a method used by Astra to scan dynamically-generated Web pages.

FIG. 12 is a flow diagram which further illustrates the method for scanning dynamically-generated Web pages.

FIGS. 13-15 are a sequence of screen displays which further illustrate the operation of Astra's dynamic page scanning feature.

FIG. 16 is a screen display which illustrates the site map of FIG. 1 following the application of a filter which filters out all URLs (and associated links) having a status other than "OK." FIG. 17 illustrates the general program sequence followed by Astra to generate filtered maps of the type shown in FIG. 16.

FIG. 18 illustrates the filtered map of FIG. 16 redisplayed in Astra's Visual Web Display format.

FIG. 19 is a screen display which illustrates an activity monitoring feature of Astra.

FIG. 20 illustrates a decision process used by Astra to generate link activity data (of the type illustrated in FIG. 19) from a server access log file.

FIG. 21 is a screen display which illustrates a map comparison tool of Astra.

FIG. 22 is a screen display which illustrates a link repair feature of Astra.

FIGS. 23 and 24 are partial screen displays which illustrate layout features in accordance with another embodiment of the invention.

The screen displays included in the figures were generated from screen captures taken during the execution of the Astra code. In order to comply with patent office standards, the original screen captures have been modified to reduce shading and to replace certain color-coded regions with appropriate cross hatching. All copyrights in these screen displays are hereby reserved.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The description of the preferred embodiments is arranged within the following sections:

- I. Glossary of Terms and Acronyms
- II. Overview
- III. Map Layout and Display Methodology
- IV. Astra Graphical User Interface
- V. Astra Software Architecture
- VI. Scanning Process
- VII. Scanning and Mapping of Dynamically-Generated Pages
- VIII. Display of Filtered Maps
- IX. Tracking and Display of Visitor Activity
- X. Map Comparison Tool
- XI. Link Repair Plug-in
- XII. Conclusion

I. Glossary of Terms and Acronyms
The following definitions and explanations provide background information pertaining to the technical field of the present invention, and are intended to facilitate an understanding of both the invention and the preferred embodiments thereof. Additional definitions are provided throughout the detailed description.

Internet. The Internet is a collection of interconnected public and private computer networks that are linked together by a set of standard protocols (such as TCP/IP, HTTP, FTP and Gopher) to form a global, distributed network.

Document. Generally, a collection of data that can be viewed using an application program, and that appears or is treated as a self-contained entity. Documents typically include control codes that specify how the document content is displayed by the application program. An "HTML document" is a special type of document which includes HTML (HyperText Markup Language) codes to permit the document to be viewed using a Web browser program. An HTML document that is accessible on a World Wide Web site is commonly referred to as a "Web document" or "Web page." Web documents commonly include embedded components, such as GIF (Graphics Interchange Format) files, which are represented within the HTML coding as links to other URLs. (See "HTML" and "URL" below.)

Hyperlink. A navigational link from one document to another, or from one portion (or component) of a document to another. Typically, a hyperlink is displayed as a highlighted word or phrase that can be clicked on using the mouse to jump to the associated document or document portion.

Hypertext System. A computer-based informational system in which documents (and possibly other types of data entities) are linked together via hyperlinks to form a user-navigable "web." Although the term "text" appears within "hypertext," the documents and hyperlinks of a hypertext system may (and typically do) include other forms of media. For example, a hyperlink to a sound file may be represented within a document by graphic image of an audio speaker.

World Wide Web. A distributed, global hypertext system, based on an set of standard protocols and conventions (such as HTTP and HTML, discussed below), which uses the Internet as a transport mechanism. A software program which allows users to request and view World Wide Web ("Web") documents is commonly referred to as a "Web browser," and a program which responds to such requests by returning ("serving") Web documents is commonly referred to as a "Web server."

Web Site. As used herein, "web site" refers generally to a database or other collection of inter-linked hypertextual

documents ("web documents") and associated data entities, which is accessible via a computer network, and which forms part of a larger, distributed informational system. Depending upon its context, the term may also refer to the associated hardware and/or software server components used to provide access to such documents. When used herein with initial capitalization (i.e., "Web site"), the term refers more specifically to a web site of the World Wide Web. (In general, a Web site corresponds to a particular Internet domain name, such as "merc-intcom," and includes the content of or associated with a particular organization.) Other types of web sites may include, for example, a hypertextual database of a corporate "intranet" (i.e., an internal network which uses standard Internet protocols), or a site of a hypertext system that uses document retrieval protocols other than those of the World Wide Web.

Content Object. As used herein, a data entity (document, document component, etc.) that can be selectively retrieved from a web site. In the context of the World Wide Web, common types of content objects include HTML documents, GIF files, sound files, video files, Java applets and aglets, and downloadable applications, and each object has a unique identifier (referred to as the "URL") which specifies the location of the object (See "URL" below.)

URL (Uniform Resource Locator). A unique address which fully specifies the location of a content object on the Internet. The general format of a URL is protocol://machine-address/path/filename. (As will be apparent from the context in which it is used, the term "URL" is also used herein to refer to the corresponding content object itself.)

Graph/Tree. In the context of database systems, the term "graph" (or "graph structure") refers generally to a data structure that can be represented as a collection of interconnected nodes. As described below, a Web site can conveniently be represented as a graph in which each node of the graph corresponds to a content object of the Web site, and in which each interconnection between two nodes represents a link within the Web site. A "tree" is a specific type of graph structure in which exactly one path exists from a main or "root" node to each additional node of the structure. The terms "parent" and "child" are commonly used to refer to the interrelationships of nodes within a tree structure (or other hierarchical graph structure), and the term "leaf" or "leaf node" is used to refer to nodes that have no children. For additional information on graph and tree data structures, see Alfred V. Aho et al, *Data Structures and Algorithms*, Addison-Wesley, 1982.

TCP/IP (Transfer Control Protocol/Internet Protocol). A standard Internet protocol which specifies how computers exchange data over the Internet. TCP/IP is the lowest level data transfer protocol of the standard Internet protocols.

HTML (HyperText Markup Language). A standard coding convention and set of codes for attaching presentation and linking attributes to informational content within documents. During a document authoring stage, the HTML codes (referred to as "tags") are embedded within the informational content of the document. When the Web document (or "HTML document") is subsequently transmitted by a Web server to a Web browser, the codes are interpreted by the browser and used to parse and display the document. In addition to specifying how the Web browser is to display the document, HTML tags can be used create hyperlinks to other Web documents. For more information on HTML, see Ian S. Graham, *The HTML Source Book*, John Wiley and Sons, Inc., 1995 (ISBN 0471-11894-4).

HTTP (Hypertext Transfer Protocol). The standard World Wide Web client-server protocol used for the exchange of

information (such as HTML documents, and client requests for such documents) between a Web browser and a Web server. HTTP includes several different types of messages which can be sent from the client to the server to request different types of server actions. For example, a "GET" message, which has the format GET <URL>, causes the server to return the content object located at the specified URL.

Webcrawling. Generally, the process of accessing and processing web site content (typically using an automated searching/parsing program) and generating a condensed representation of such content Webcrawling routines are commonly used by commercial Internet search engines (such as Infoseek™ and Alta Vista™) to generate large indexes of the terms that appear within the various Web pages of the World Wide Web.

API (Application Program Interface). A software interface that allows application programs (or other types of programs) to share data or otherwise communicate with one another. A typical API comprises a library of API functions or "methods" which can be called in order to initiate specific types of operations.

CGI (Common Gateway Interface). A standard interface which specifies how a Web server (or possibly another information server) launches and interacts with external programs (such as a database search engine) in response to requests from clients. With CGI, the Web server can serve information which is stored in a format that is not readable by the client, and present such information in the form of a client-readable Web page. A CGI program (called a "CGI script") may be invoked, for example, when a Web user fills out an on-screen form which specifies a database query. For more information on CGI, see Ian S. Graham, *The HTML Source Book*, John Wiley and Sons, Inc., 1995 (ISBN 0471-11894-4), pp. 231-278.

OLE (Object Linking and Embedding). An object technology, implemented by Windows-based applications, which allows objects to be linked to one another and embedded within one another. OLE Automation, which is a feature of OLE 2, enables a program's functionality to be exposed as OLE objects that can be used to build other applications. For additional information on OLE and OLE Automation, see *OLE 2 Programmer's Reference Manual, Volume One*, Microsoft Corporation, 1996 (ISBN 1-55615-628-6).

II. Overview

The present invention provides a variety of software-related features for facilitating the mapping, analysis and management of Web sites. In the preferred embodiment, these features are embodied within a software package which runs on a client computer under either the Windows® NT or the Windows® 95 operating system. The software package is referred to herein as "Astra."

Given the address of a Web site's home page, Astra automatically scans the Web site and creates a graphical site map showing all of the URLs of the site and the links between these URLs. In accordance with one aspect of the invention, the layout and display method used by Astra for generating the site map provides a highly intuitive, graphical representation which allows the user to visualize the layout of the site. Using this mapping feature, in combination with Astra's powerful set of integrated tools for navigating, filtering and manipulating the Web site map, users can intuitively perform such actions as isolate and repair broken links, focus in on Web pages (and other content objects) of a particular content type and/or status, and highlight modi-

fications made to a Web site since a prior mapping. In addition, users can utilize a Dynamic Scan™ feature of Astra to automatically append dynamically-generated Web pages (such as pages generated using CGI scripts) to their maps. Further, using Astra's activity monitoring features, users can monitor visitor activity levels on individual links and URLs, and study visitor behavior patterns during Web site visits.

In accordance with another aspect of the invention, Astra has a highly extensible architecture which facilitates the addition of new tools to the Astra framework. As part of this architecture, a "core" Astra component (which includes the basic Web site scanning and mapping functionality) has an API for supporting the addition of plug-in components. This API includes functions for allowing the plug-in components to manipulate the display of the site map, and to display their own respective data in conjunction with the Astra site map. Through this API, new applications can be added which extend the functionality of the package while taking advantage of the Astra mapping scheme.

Throughout this description, preliminary names of product features and software components are used with initial capitalization. These names are used herein for ease of description only, and are not intended to limit the scope of the invention.

FIGS. 1-3 illustrate Astra's primary layout methodology, referred to herein as "Visual Web Display™," for displaying graphical representations ("maps") of Web sites. These figures will also be used to describe some of the graphical user interface (GUI) features of Astra.

FIG. 1 illustrates a site map 30 of a demonstration Web site which was derived from the actual Web site of Mercury Interactive, Inc. (i.e., the URLs accessible under the "merc-int.com" Internet domain name). (For purposes of this detailed description, it may be assumed that "Web site" refers to the content associated with a particular Internet domain name.) The Web site is depicted by Astra as a S collection of nodes, with pairs of nodes interconnected by lines. Each node of the map represents a respective content object of the Web site and corresponds to a respective URL. (The term "URL" is used herein to refer interchangeably to both the address of the content object and to the object itself; where a distinction between the two is helpful to an understanding of the invention, the term "URL" is followed by an explanatory parenthetical.) Examples of URLs (content objects) which may exist within a typical Web site include HTML documents (also referred to herein as "Web pages"), image files (e.g., GIF and PCX files), mail messages, Java applets and aglets, audio files, video files, and applications.

As generally illustrated by FIGS. 3 and 4, different icons are used to represent the different URL types when the nodes are viewed in a sufficiently zoomed-in mode. (Generic icons of the type best illustrated by FIG. 18 are used to display nodes that fall below a predetermined size threshold.) As described below, special icons and visual representations are also used to indicate status information with respect to the URLs. For example, special icons are used to depict, respectively, inaccessible URLs, URLs which are missing, URLs for which access was denied by the server, and URLs which have been detected but have not been scanned. (The term "scan" refers generally to the process of sending informational requests to server components of a computer network, and in the context of the preferred embodiment, refers to the process of sending requests to Web server components to obtain Web site content associated with specific URLs.)

The lines which interconnect the nodes (URI icons) in FIGS. 1-3 (and the subsequent figures with screen displays) represent links between URLs. As is well understood in the art, the functions performed by these links vary according to URL type. For example, a link from one HTML document to another HTML document normally represents a hyperlink which allows the user to jump from one document to the other while navigating the Web site with a browser. In FIG. 1, an example of a hyperlink which links the home page URL (shown at the center of the map) to another HTML page (displayed to the right of the home page) is denoted by reference number 32. (As generally illustrated in FIG. 1 and the other figures which illustrate screen displays, regular HTML documents are displayed by Astra as a shaded document having text thereon.) A link between an HTML document and a GIF file, such as link 36 in FIG. 3, normally represents a graphic which is embedded within the Web page.

Maps of the type illustrated in FIG. 1 are generated by Astra using an HTTP-level scanning process (described below) which involves the reading and parsing the Web site's HTML pages to identify the architecture (i.e., the arrangement of URLs and links) of the Web site, and to obtain various status information (described below) about the Web site's URLs. The basic scanning process used for this purpose is generally similar to the scanning process used by conventional Web crawlers. As part of Astra's Dynamic Scan feature, Astra additionally implements a special dynamic page scanning process which permits dynamically-generated Web pages to be scanned and included in the Web site map. As described below, this process involves capturing the output of a Web browser when the user submits an HTML-embedded form (such as when the user submits a database query), and then reusing the captured dataset during the scanning process to recreate the form submission and append the results to the map.

Table 1 lists the predefined icons that are used by Astra to graphically represent different URL types within site maps. As illustrated, the URL icons generally fall into two categories: object-type ("URL type") icons and status icons. The object-type icons are used to indicate the content or service type of URLs that have been successfully scanned. The status icons are used to indicate the scanning status (not found, access denied, etc.) of URLs for which either (i) scanning has not been performed, or (ii) scanning was unsuccessful. Various examples of these two types of icons are included in the figures.

TABLE 1

URL Type	Scanning Status
HTML	Not found
HTML with Form	Not Scanned
Image	Inaccessible
Sound	Access Denied
Application	
Text	
Unknown	
Video	
Gopher	
FTP	
Dynamic Page	

Once the map has been generated, the user can interactively navigate the map using various navigation tools of the Astra GUI, such as the zoom-in and zoom-out buttons 34, 36 (FIG. 1) and the scrolling controls 40, 42 (FIGS. 2 and 3). To zoom-in on a particular region of the map 30, the user can

click on the zoom-in button 34 and then use the mouse to draw a box around the map region of interest; Astra will then re-size the highlighted region to generally fit the display screen. As will be recognized by those skilled in the art, the ability to zoom in and out between high level, perspective views which reveal the overall architecture of the site, and magnified (zoomed-in) sub-views which reveal URL-specific information about the Web site, greatly facilitates the task of navigating and monitoring Web site content.

As generally illustrated by FIG. 3, the annotations (page titles, filenames, etc.) of the URLs begin to appear (below the associated icons) as the user continues to zoom in. As further illustrated by FIG. 3, the URL (address) of a node is displayed when the mouse cursor is positioned over the corresponding icon.

While navigating the map, the user can retrieve a URL (content object) from the server by double-clicking on the corresponding URI icon; this causes Astra to launch the client computer's default Web browser (if not already running), which in-turn retrieves the URL from the Web server. For example, the user can double-click on the URL icon for an HTML document (using the left mouse button) to retrieve and view the corresponding Web page. When the user clicks on a URL icon using the right mouse button, a menu appears which allows the user to perform a variety of actions with respect to the URL, including viewing the URL's properties, and launching an HTML editor to retrieve and edit the URL. With reference to FIG. 3, for example, the user can click on node 44 (using the right mouse button), and can then launch an HTML editor to edit the HTML document and delete the reference to missing URL 45. (As illustrated by FIG. 3, missing URLs are represented within Astra maps by a question mark icon.)

One important feature of Astra, referred to herein as "Automatic Update," allows the user to update an existing Web site map to reflect any changes that have been made to the map since a prior mapping of the site. To initiate this feature, the user selects a "start Automatic Update" button 37 (FIG. 1), or selects the corresponding menu item, while viewing a site map. This initiates a re-scanning process in which Astra scans the URLs of the Web site and updates the map data structure to reflect the current architecture of the site. As part of this process, Astra implements a caching protocol which eliminates the need to download URLs and URL headers that have not been modified since the most recent mapping. (This protocol is described below under the heading "SCANNING PROCESS.") This typically allows the map to be updated in a much shorter period of time than is required to perform the original mapping. This feature is particularly useful for Webmasters of complex Web sites that have rapidly changing content.

III. Map Layout and Display Methodology (FIGS. 1-3, 23 and 24)

An important aspect of the invention is the methodology used by Astra for presenting the user with a graphical, navigable representation of the Web site. This feature of Astra, which is referred to as Visual Web Display (abbreviated as "VWD" herein), allows the user to view and navigate complex Web structures while visualizing the inter-relationships between the data entities of such structures. The method used by Astra to generate VWD site maps is referred to herein as the "Solar Layout method," and is described at the end of this section.

One aspect of the VWD format is the manner in which children nodes ("children") are displayed relative to their respective parent nodes ("parents"). (In the context of the

preferred embodiment, the term "node" refers generally to a URL icon as displayed within the site map.) As illustrated by the collection of nodes shown in FIG. 3, the parent 44 is displayed in the center of the cluster, and the seven children 48 are positioned around the parent 44 over an angular range of 360 degrees. One benefit of this layout pattern is that it allows collections of related nodes to be grouped together on the screen in relatively close proximity to one another, making it easy for the user identify the parent-child relationships of the nodes. This is in contrast to the expandable folder type representations used by Webmapper™, the Windows® 95 Explorer, and other WindowsO applications, in which it is common for a child to be separated from its parent folder by a long list of other children.

In this FIG. 3 example, all of the children 48 are leaf nodes (i.e., nodes which do not themselves have children). As a result, all of the children 48 are positioned approximately equidistant from the parent 44, and are spaced apart from one another by substantially equal angular increments. Similar graphical representations to that of FIG. 3 are illustrated in FIG. 1 by node clusters 52, 54 and 56. As illustrated by these three clusters in FIG. 1, both (i) the size of parent icon and (ii) the distance from the parent to its children are proportional to the number of immediate children of the parent. Thus, for example, cluster 56 has a larger diameter (and a larger parent icon) than clusters 52 and 54. This has the desirable effect of emphasizing the pages of the Web site that have the largest numbers of outgoing links. (As used herein, the term "outgoing links" includes links to GIF files and other embedded components of document.)

As best illustrated by cluster 64 in FIG. 2, of which node 65 is the primary parent or "root" node, children which have two or more of their own children (i.e., grandchildren of the root) are positioned at a greater distance from the root node 65 than the leaf nodes of the cluster, with this distance being generally proportional to the size of the sub-cluster of which the child is the parent. For example, node 66 (which has 3 children) is positioned farther from the cluster's root node 65 than leaf nodes 70; and the parent of cluster 60 is positioned farther from the root node 65 than node 66. As illustrated in FIG. 1, this layout principal is advantageously applied to all of the nodes of the Web site that have children. The recursive method (referred to as "Solar Layout") used by Astra to implement these layout and display principles is described below.

Another aspect of the layout method is that the largest "satellite" cluster of a parent node is centered generally opposite from (along the same line as) the incoming link to the parent node. This is illustrated, for example, by cluster 54 in FIG. 1 and by cluster 60 in FIG. 2, both of which are positioned along the same line as their respective parents. This aspect of the layout arrangement tends to facilitate visualization by the user of the overall architecture of the site.

As will be apparent from an observation of FIG. 1, the graphical map produced by the application of the above layout and display principles has a layout which resembles the general arrangement of a solar system, with the home page positioned as the sun, the children of the home page being in orbit around the sun, the grandchildren of the home page being in orbit around their immediate respective parents, and so on. One benefit of this mapping arrangement is that it is well suited for displaying the entire site map of a complex Web site on a single display screen (as illustrated in FIG. 1). Another benefit is that it provides an intuitive structure for navigating the URLs of a complex Web site. While this mapping methodology is particularly useful for

the mapping of Web sites, the methodology can also be applied, with the realization of similar benefits, to the mapping of other types of databases. For example, the VWD methodology can be used to facilitate the viewing and navigation of a conventional PC file system.

Another benefit of this site map layout and display methodology is that the resulting display structure is well suited for the overlaying of information on the map. Astra takes full advantage of this benefit by providing a set of API functions which allow other applications (Astra plug-ins) to manipulate and add their respective display data to the site map. An example of an Astra plug-in which utilizes this feature is the Action Tracker™ tool, which superimposes user activity data onto the site map based on analyses of server access log files. The Astra plug-in API and the Action Tracker plug-in are described in detail below.

As illustrated in FIG. 1, all of the nodes of the site map (with the exception of the home page node) are displayed as having a single incoming link, even though some of the URLs of the depicted Web site actually have multiple incoming links. Stated differently, the Web site is depicted in the site map 30 as though the URLs are arranged within a tree data structure (with the home page as the main root), even though a tree data structure is not actually used. This simplification to the Web site architecture is made by extracting a span tree from the actual Web site architecture prior to the application of a recursive layout algorithm, and then displaying only those links which are part of the spanning tree. (In applications in which the database being mapped is already arranged within a tree directory structure, this step can be omitted.) As a result, each URL of the Web site is displayed exactly once in the site map. Thus, for example, even though a particular GIF file may be embedded within many different pages of the Web site, the GIF file will appear only once within the map. This simplification to the Web site architecture for mapping purposes makes it practical and feasible to graphically map, navigate and analyze complex Web sites in the manner described above.

Because the Visual Web Display format does not show all of the links of the Web site, Astra supports two additional display formats which enable the user to display, respectively, all of the incoming links and all of the outgoing links of a selected node. To display all of the outgoing links of a given node, the user selects the node with the mouse and then selects the "display outgoing links" button 72 (FIG. 1) from the tool bar 46. Astra then displays a hierarchical view (in the general form of a tree) of the selected node and its outgoing links, as illustrated by FIG. 6. Similarly, to display the incoming links of a node, the user selects the node and then clicks on the "display incoming links" button 71. (A screen display illustrating the incoming links format is shown in FIG. 22.) To restore the Visual Web Display view, the user clicks on the VWD button 73.

The Solar Layout method (used to generate VWD-format site maps) generally consists of three steps, the second two of which are performed recursively on a node-by-node basis. These three steps are outlined below, together with associated pseudocode representations. In addition, a source code listing of the method (in C++) is included in the microfiche appendix as Appendix A.

Step 1—Select Span Tree

In this step, a span tree is extracted from the graph data structure which represents the arrangement of nodes and links of the Web site. (The graph data structure is implemented as a "Site Graph" OLE object, as described below.)

Any standard span tree algorithm can be used for this purpose. In the preferred embodiment, a shortest-path span tree algorithm known as “Dijkstra’s algorithm” is used, as implemented within the commercially-available LEDA (Library of Efficient Data types and Algorithms) software package. As applied within Astra, this algorithm finds the shortest paths from a main root node (corresponding to the Web site’s home page or some other user-specified starting point) to all other nodes of the graph structure. The result of this step is a tree data structure which includes all of the URLs of the graph data structure with the home page represented as the main root of the tree. For examples of other span tree algorithms which can be used, see Alfred V. Aho et al, *Data Structures and Algorithms*, Addison-Wesley, 1982.)

Step 2—Solar Plan

This is a recursive step which is applied on a node-by-node basis in order to determine (i) the display size of each node, (ii) the angular spacings for positioning the children nodes around their respective parents, and (iii) the distances for spacing the children from their respective parents. For each parent node, the respective sizes of the parent’s satellites are initially determined. (A “satellite” is any child of the parent plus the child’s descendants, if any.) The satellite sizes are then used to allocate (a) angular spacings for positioning the satellites around the parent, and (b) the radial distances between the satellites and the parent. This process is repeated for each parent node (starting with the lower level parent nodes and working up toward the home page) until all nodes of the graph have been processed. The following is a pseudocode representation of this process:

```

Node::SolarPlan()
{
  IF node has no children
    return basic graphical dimension for a single node
  ELSE
    For each linked node as selected in the span tree, call SolarPlan()
    recursively;
    Based on the sum of the sizes of the satellites, allocate angle for
    positioning satellites around parent, and set satellite distances from
    parent;
    Calculate size of present cluster (parent plus satellites).
}

```

A modified Solar Plan process which incorporates two additional layout features is described below and illustrated by FIGS. 23 and 24,

Step 3—Solar Place

This step recursively positions the nodes on the display screen, and is implemented after Step 2 has been applied to all of the nodes of the graph. The sequence starts by positioning the home page at the center, and then uses the angle and distance settings calculated in Step 2 to position the children of the home page around the home page. This process is repeated recursively for each parent node until all of the nodes have been positioned on the screen.

```

Node::SolarPlace(x, y, entry_angle)
{
  Move this node to location (x,y)
  For each satellite:

```

-continued

```

  calculate final angle as the sum of the entry_angle and the angle
  allocated in Step 2; Calculate satellite center (x and y coordinates)
  based on new angle and distance from current node; Call SolarPlace
  using the above-calculated angle and location.
}

```

In the above pseudocode representation, the “x” and “y” parameters specify the screen position for the placement of a node (icon), and the “entry_angle” parameter specifies the angle of the line (link) between the node and its respective parent. In the preferred embodiment, the method is implemented such that the largest satellite of a parent node is positioned using the same entry angle as the parent node, so that the satellite center, parent node, and parent of the parent node all fall generally along the same line. (The determination of the largest satellite is performed in Step 2.) As indicated above, this aspect of the layout method is illustrated in FIG. 1 by cluster (satellite) 54, which is positioned along the same line as both its immediate parent icon and the home page icon.

A modified Solar Plan process will now be described with reference to the screen displays of FIGS. 23 and 24, and to the corresponding pseudocode representation below. This modified process incorporates two additional layout features which relate to the positioning of the satellites around a parent. These layout features are implemented within the source code listing (Appendix A), and are represented generally by the highlighted text of the following pseudocode sequence:

```

Node::SolarPlan()
{
  IF node has no children
    return basic graphical dimension for a single node
  ELSE
    For each linked node as selected in the span tree, call SolarPlan()
    recursively;
    Based on the sum of the sizes of the satellites + minimal weight of the
    incoming link, allocate angle for positioning satellites around parent,
    and set satellite distances from parent;
    Sort satellite list as follows: smallest child first, and in jumps of two
    next child up to the biggest, and then back to second biggest and in
    jumps of two down to smallest (e.g. 1, 3, 5 . . . biggest second
    biggest, . . . 6, 4, 2);
    Calculate size of present cluster (parent plus satellites).
}

```

The first of the two layout features is illustrated by FIG. 23, which is a partial screen display (together with associated annotations) of a parent-child cluster comprising a parent 79 and seven children or satellites 75. This layout feature involves allocating an angular interval (e.g., 20 degrees) to the incoming link 81 to the parent 79, and then angularly spacing the satellites 75 (which in this example are all leaf nodes) over the remaining angular range. In the preferred embodiment, this is accomplished by assigning a minimal weight (corresponding to the angular interval) to the incoming link 81, and then treating this link 81 as one of the outgoing links 83 when assigning angular positions to the satellites 75. As a result of this step, the satellites 75 are positioned around the parent 79 over an angular range of less than 360 degrees—in contrast to the clusters of FIGS. 1–5, in which the satellites are positioned over the full 360 range. (In this FIG. 23 example, because all of the satellites 75 are leaf nodes, the satellites 75 are positioned equidistant from the parent 79 with equal angular spacings.) One benefit of

this added step is that it allows the user to more easily distinguish the incoming link **81** to a parent **79** from the outgoing links **83** from the parent. With reference to the angular notations of FIG. 23, the minimal weight is preferably selected such that the angle θ , between the incoming link **81** and each of the two adjacent parent-child links **83** is greater than or equal to the minimum angle θ_2 between adjacent parent-child links **83** for the given cluster. This layout feature is also illustrated by FIG. 24.

The second of the two additional layout features involves ordering the satellites around the parent based on the respective sizes of the satellites. This feature comes into play when a parent node has multiple satellites that differ in size from one another. The layout arrangement which is produced by this feature is generally illustrated by FIG. 24, which shows a cluster having a parent node (labeled "CNN SHOWBIZ") and 49 satellites. As illustrated by this screen image, the satellites are ordered such that the smallest satellites **85** are angularly positioned closest to the incoming link **89** to the parent, and such that the largest satellites **91A-E** are positioned generally opposite from the incoming link **89**. This is preferably accomplished by sorting the satellites using the sorting algorithm of the above pseudocode sequence (which produces a sorted satellite list in which the satellites progress upward from smallest to largest, and then progress downward from second largest to second smallest), and then positioning the satellites around the parent (starting at the incoming link **89**) in the order which results from the sorting process. In this example, the largest satellite **91A** is positioned opposite the incoming link **89**; the second and third largest satellites **91B** and **91C** are positioned adjacent to the largest satellite **91A**; the fourth and fifth largest satellites **91D** and **91E** are positioned adjacent to the second and third largest satellites **91B** and **91C** (respectively); and so on. As is apparent from FIG. 24, this layout feature tends to produce a highly symmetrical layout.

Other aspects of the Solar Layout method will be apparent from an observation of the screen displays and from the source code listing of Appendix A.

IV. Astra Graphical User Interface (FIGS. 1 and 4-6)

As illustrated in FIG. 1, the Astra menu bar includes seven menu headings: FILE, VIEW, SCAN, MAP, URL, TOOLS and HELP. From the FILE menu the user can perform various file-related operations, such as save a map file to disk or open a previously generated map file. From the VIEW menu the user can select various display options of the Astra GUI. From the SCAN menu the user can control various scanning-related activities, such as initiate or pause the automatic updating of a map, or initiate a dynamic page scan session. From the MAP menu, the user can manipulate the display of the map, by, for example, collapsing (hiding) all leaf nodes, or selecting the Visual Web Display mode. From the URL menu, the user can perform operations with respect to user-selected URIs, such as display the URI's content with a browser, invoke an editor to modify the URI's content, and display the incoming or outgoing links to/from the URL.

From the TOOLS menu the user can invoke various analysis and management related tools. For example, the user can invoke a map comparison tool which generates a graphical comparison between two maps. This tool is particularly useful for allowing the user to readily identify any changes that have been made to a Web site's content since a previous mapping. The user can also invoke the Action Tracker tool, which superimposes link activity data on the Web site map to allow the user to readily ascertain the links and URLs that have the most hits. (The Action Tracker tool

is described in detail below under the heading "TRACKING AND DISPLAY OF VISITOR ACTIVITY.") The user can also invoke a Link Doctor tool which facilitates the repairing of broken links. These and other tools of Astra are described in the subsequent sections.

With further reference to FIG. 1, the Astra GUI includes a tool bar **46** and a filter bar **47**, both of which can be selectively displayed as needed. The tool bar **46** includes buttons for initiating commonly-performed operations. From left to right in FIG. 1, these functions are as follows: (a) start generation of new map, (b) open map file, (c) save map to disk, (d) print, (e) size map to fit within window, (f) zoom in, (g) zoom out, (h) display incoming links of selected node; (i) display outgoing links of selected node, (j) display map in Visual Web Display format, (k) initiate Automatic Update, (l) pause Automatic Update, (m) resume Automatic Update, (n) initiate Dynamic Scan, and (o) stop Dynamic Scan. (The function performed by each button is indicated textually when the mouse cursor is positioned over the respective button.)

The filter bar **47** includes a variety of different filter buttons for filtering the content of site maps. When the user clicks on a filter button, Astra automatically hides all links and pages of a particular type or status, as illustrated in FIG. 16 and discussed below. The filter buttons are generally divided into three groups: content/service filters **49**, status filters **50**, and location filters **51**. From left to right in FIG. 1, the content/service filters **49** filter out URLs of the following content or service types: (a) HTML, (b) HTML forms, (c) images, (d) audio, (e) CGI, (f) Java, (g) other applications, (h) plain text, (i) unknown, (j) redirect, (k) video, (l) Gopher, (m) FTP, and (n) all other Internet services. The status filters **50** filter out URLs of the following statuses (from left to right): (a) not found, (b) inaccessible (e.g., no response from server), (c) access denied, (d) not scanned, and (e) OK. The left-hand and right-hand location filters **51** filter out local URLs and external URLs, respectively, based on the domain names of the URLs. Multiple filters can be applied concurrently.

FIG. 4 illustrates a split-screen mode which allows the user to view a graphical representation of the Web site in an upper window **76** while viewing a corresponding textual representation (referred to as "List View") in a lower window **78**. To expose the List View window **78**, the user drags and drops the separation bar **80** to the desired position on the screen. Each line of text displayed in the List View window **78** represents one node of the site map, and includes various information about the node. For each node, this information includes: the URL (i.e., address), an annotation, the scanning status (OK, not found, inaccessible, etc.), the associated communications protocol (HTTP, mailto, FTP, etc.), the content type, the file size (known only if the entire file has been retrieved), the numbers of inbound links and outbound links, and the date and time of last modification. (The outbound link and last modification information can be exposed in the FIG. 4 screen display by dragging the horizontal scrolling control **77** to the right.)

As described below, this information about the nodes is obtained by Astra during the scanning process, and is stored in the same data structure **114** (FIG. 9) that is used to build the map. As additionally described below, whenever the user initiates an Automatic Update, Astra uses the date/time of last modification information stored locally in association with each previously-mapped HTML document to determine whether the document needs to be retrieved and parsed. (The parsing process is used to identify links to other URLs, and to identify other HTML elements relevant to the

mapping process.) As indicated above, this provides the significant advantage of allowing the Web site to be re-mapped without having to repeat the entire scanning/parsing process.

With further reference to FIG. 4, whenever the user selects a node in the upper window 76, the corresponding line in the List View window 78 is automatically highlighted. (As illustrated by node 84 in FIG. 4, Astra graphically represents the selection of a node by outlining the node's icon in black.) Likewise, whenever the user selects a line in the List View window 78, the corresponding node is automatically highlighted in the upper window 76. This feature allows the user to rapidly and efficiently associate each textual line with its graphical counterpart, and vice versa. In addition, by clicking on the headers 82 of the separation bar 80, the user can view the listed URLs in a sorted order. For example, if the user clicks on the "in links" header, Astra will automatically sort the list of URLs according to the number of incoming links, and then display the sorted listing in the List View window 78.

FIG. 5 illustrates a Pan Window feature of Astra. This feature facilitates navigation of the site map while in a zoomed-in mode by presenting the user with a perspective view of the navigational position within the map. To display the Pan Window 86, the user selects the "Pan Window" menu option from the VIEW menu while viewing a map. Within the Pan Window, the user is presented with a display of the entire map 30, with a dashed box 87 indicating the portion of the map that corresponds to the zoomed-in screen display. As the user navigates the site map (using the scrolling controls 40, 42 and/or other navigational controls), the dashed box automatically moves along the map to track the zoomed-in screen display. The user can also scroll through the map by simply dragging the dashed box 87 with the mouse. In the preferred embodiment, the Pan Window feature is implemented in-part using a commercially-available from Stingray™ Corporation called SEC++, which is designed to facilitate the zoomed-in viewing of a general purpose graphic image.

FIG. 6 illustrates the general display format used by Astra for displaying the outgoing links of a selected node 88. To display a node's outgoing links, the user selects the node with the mouse and then clicks on the "show outgoing links" button 72 on the tool bar. As illustrated, Astra then displays all outgoing links from the node (including any links that do not appear in the VWD site map), and displays additional levels of outgoing links (if any) which emanate from the children of the selected node. The display format used for this purpose is in the general format of a tree, with the selected node displayed as the root of the tree. An analogous display format (illustrated in FIG. 22) is used for displaying the incoming links to a node.

V. Astra Software Architecture (FIGS. 7 and 8)

FIG. 7 pictorially illustrates the general architecture of Astra, as installed on a client computer 92. As illustrated, the architecture generally consists of a core Astra component 94 which communicates with a variety of different Astra plug-in applications 96 via a plug-in API 98. The Astra core 94 includes the basic functionality for the scanning and mapping of Web sites, and includes the above-described GUI features for facilitating navigation of Web site maps. Through the plug-in API 98, the Astra core 94 provides an extensible framework for allowing new applications to be written which extend the basic functionality of the Astra core. As described below, the architecture is structured such that the plug-in applications can make extensive use of Astra site maps to display plug-in specific information

The Astra plug-ins 96 and API 98 are based on OLE Automation technology, which provides facilities for allowing the plug-in components to publish information to other objects via the operating system registry (not shown). (The "registry" is a database used under the Windows® 95 and Windows® NT operating systems to store configuration information about a computer, including information about Windows-based applications installed on the computer.) At start-up, the Astra core 94 reads the registry to identify the Astra plug-ins that are currently installed on the client computer 92, and then uses this information to launch the installed plug-ins.

In a preferred implementation, the architecture includes five Astra plug-ins: Link Doctor, Action Tracker, Test World, Load Wizard and Search Meter. The functions performed by these plug-ins are summarized by Table 2. Other applications which will normally be installed on the client computer in conjunction with Astra include a standard Web browser (FIGS. 11 and 12), and one or more editors (not shown) for editing URL content.

TABLE 2

PLUG-IN	FUNCTION PERFORMED
Link Doctor	Fixes broken links automatically
Action Tracker	Retrieves and evaluates server log files to generate Web site activity data (such as activity levels on individual links), and superimposes such data on site map in a user-adjustable manner.
Test World	Generates and drives tests automatically
Load Wizard	Utilizes site map to automatically generate test scripts for the load testing of Web sites with Mercury Interactive's LoadRunner™ and SiteTest™ software packages.
Search Meter	Displays search engine results visually

The Astra API allows external client applications, such as the plug-in applications 96 shown in FIG. 7, to communicate with the Astra core 94 in order to form a variety of tasks. Via this API, client applications can perform the following types of operations:

1. Superimpose graphical information on the site map;
2. Access information gathered by the Astra scanning engine in order to generate Web site statistics;
3. Attach custom attributes to the site map, and to individual nodes and links of the site map;
4. Access some or all of a Web page's contents (HTML) during the Web site scanning process;
5. Embed the Astra GUI within the client application;
6. Add menu items to the Astra menu; and
7. Obtain access to network functionality.

The specific objects and methods associated with the API are discussed below with reference to FIG. 8. In addition, a complete listing of the API is included in the microfiche as Appendix B.

During the Web site scanning process, the Astra core 94 communicates over the Internet 110 (or an intranet) with the one or more Web server applications 112 ("Web servers") which make up the subject Web site 113. The Web servers 112 may, for example, run on a single computer, run on multiple computers located at a single geographic location (which may, but need not, be the location of the client computer 92), or run on multiple computers that are geographically distributed. In addition, the Web servers 112 of the Web site 113 may be virtually distributed across multiple Internet domains.

As is conventional with Internet applications, the Astra core 94 uses the TCP/IP layer 108 of the computer's

operating system to communicate with the Web site 113. Any one or more of the Astra plug-ins 96 may also use the TCP/IP layer 108 to communicate with the Web site 113. In the preferred embodiment, for example, the Action Tracker plug-in communicates with the Web sites (via the Astra plug-in API) to retrieve server access log files for performing Web site activity analyses.

FIG. 8 illustrates the object model used by the Astra API. As illustrated, the model includes six classes of objects, all of which are implemented as OLE Automation objects. By name, the six object classes are Astra, Site Graph, Edges, Edge, Nodes, and Node. The Astra object 94 is an application object, and corresponds generally to the Astra core 94 shown in FIG. 7. The Astra object 94 accesses and manipulates data stored by a Site Graph object 114. Each Site Graph object corresponds generally to a map of a Web site, and includes information about the URLs and links (including links not displayed in the Visual Web Display view) of the Web site. The site-specific data stored by the Site Graph object 114 is contained within and managed by the Edges, Edge, Nodes and Nodes objects, which are subclasses of the Graph object.

Each Node object 115 represents a respective node (URL) of the site map, and each Edge object 116 represents a respective link between two URLs (nodes) of the map. Associated with each Node object and each Edge object is a set of attributes (not shown), including display attributes which specify how the respective object is to be represented graphically within the site map. For example, each Node object and each Edge object include respective attributes for specifying the color, visibility, size, screen position, and an annotation for the display of the object. These attributes can be manipulated via API calls to the methods supported by these objects 115, 116. For example, the Astra plug-ins (FIG. 7) can manipulate the visibility attributes of the Edge objects to selectively hide the corresponding links on the screen. (This feature is illustrated below in the description of the Action Tracker plug-in.) In addition, the Astra API includes methods for allowing the plug-ins to define and attach custom attributes to the Edge and Node objects.

The Nodes and Edges objects 118, 119 are container objects which represent collections of Node objects 115 and Edge objects 116, respectively. Any criterion can be used by the applications for grouping together Node objects and Edge objects. As depicted in FIG. 8, a single Graph object 114 may include multiple Nodes objects 118 and multiple Edges objects 119.

The methods of the Astra plug-in API generally fall into five functional categories. These categories, and the objects to which the associated methods apply, are listed below. Additional information on these methods is provided in the API listing in Appendix B.

ASTRA GUI METHODS. These methods control various aspects of the Astra GUI, such as adding, deleting, enabling and disabling Astra menu items.

Supporting objects: Astra, Site Graph.

GROUPING AND ACCESS METHODS. These methods permit groupings of nodes and links to be formed, and permit the nodes and links within these groups to be accessed. Supporting objects: Site Graph, Nodes, Edges.

NODE/EDGE APPEARANCE METHODS. These methods provide control over display attributes (visibility, color, etc.) of links and nodes of the map.

Supporting Objects: Node, Edge.

ATTRIBUTE ATTACHMENT METHODS. These methods permit the attachment of custom information to specific objects, and provide access to such information. Supporting

objects: Site Graph, Node, Edge. Example use: Number of "hits" displayed by Action Tracker.

SCAN-TIME CONTENT ACCESS METHODS. These methods provide access by applications to Web page content retrieved during the scanning process. Supporting Objects: Site Graph, Node. Example use: At scan time, textual content of each page is passed to a spell checker application to perform a site-wide spell check.

As will be appreciated from the foregoing, the Astra architecture provides a highly extensible mapping framework which can be extended in functionality by the addition of new plug-ins applications. Additional aspects of the architecture are specified in the API description of Appendix B.

VI. Scanning Process (FIGS. 9 and 10)

As will be apparent, the terms "node" and "link" are used in portions of the remaining description to refer to their corresponding object representations—the Node object and the Edge object.

The multi-threaded scanning process used by the Astra core 94 for scanning and mapping a Web site will now be described with reference to FIGS. 9 and 10. As depicted in FIG. 9, Astra uses two types of threads to scan and map the Web site: a main thread 122 and multiple lower-level scanning threads 122. The use of multiple scanning threads provides the significant benefit of allowing multiple server requests to be pending simultaneously, which in-turn reduces the time required to complete the scanning process. A task manager process (not shown) handles issues related to the management of the threads, including the synchronization of the scanning threads 120 to the main thread 120, and the allocation of scanning threads 122 to operating system threads.

The main thread 120 is responsible for launching the scanning threads 122 on a URL-by-URL basis, and uses the URL-specific information returned by the scanning threads 122 to populate the Site Graph object 114 ("Site Graph") with the nodes, links, and associated information about the Web site 113. In addition, as pictorially illustrated by the graph and map symbols in box 114, the main thread 120 periodically applies the Solar Layout method to the nodes and links of the Site Graph 114 to generate a map data structure which represents the Visual Web Display map of the Web site. (As described below, this map data structure is generated by manipulating the display attributes of the Node objects and Edge objects, and does not actually involve the generation of a separate data structure.)

Upon initiation of the scanning process by the user, the main thread 120 obtains the URL (address) of the home page (or the URL of some other starting location) of the Web site to be scanned. If the scanning process is initiated by selecting the "Automatic Update" option, the main thread 120 obtains this URL from the previously-generated Site Graph 114. Otherwise, the user is prompted to manually enter the URL of the home page.

Once the home page URL has been obtained, the main thread 120 launches a scanning thread 122 to scan the HTML home page. As the HTML document is returned, the scanning thread 122 parses the HTML to identify links to other URLs, and to identify other predetermined HTML elements (such as embedded forms) used by Astra. (As described below with reference to FIG. 10, if an Automatic Update is being performed, the scanning thread downloads the home page only if the page has been modified since the last scanning of the URL; if no download of the page is required, this outgoing link information is extracted from the previously-generated Site Graph 114.) In addition, the scan-

ning thread 122 extracts certain information from the header of the HTML document, including the date/time of last modification, and the other information displayed in the List View window 78 of FIG. 4. The link and header information extracted by the scanning thread 122 is represented in FIG. 9 by one of the boxes 130 labeled "URL data."

Upon completion, the scanning thread 122 notifies the main thread 120 that it has finished scanning the home page. The main thread then reads the URL data extracted by the scanning thread 122 and stores this data in the Site Graph 114 in association with a Node object which represents the home page URL. In addition, for each internal link (i.e., link to a URL within the same Internet domain) identified by the scanning thread 122, the main thread 120 creates (or updates) a corresponding Edge object and a corresponding Node object within the Site Graph 114, and launches a new scanning thread 122 to read the identified URL. (Edge and Node objects are also created for links to external URLs, but these external URLs are not scanned in the default mode.) These newly-launched scanning threads then proceed to scan their respective URLs in the same manner as described above (with the exception that no downloading and parsing is performed when the subject URL is a non-HTML file). Thus, scanning threads 122 are launched on a URL-by-URL basis until either all of the URLs of the site have been scanned or the user halts the scanning process. Following the completion of the scanning process, the Site Graph 114 fully represents the site map of the Web site, and contains the various URL-specific information displayed in the Astra List View window 78 (FIG. 4). When the user saves a site map via the Astra GUI, the Site Graph 114 is written to disk.

In a default mode, links to external URLs detected during the scanning process are displayed in the site map using the "not scanned" icon (192 in FIG. 13), indicating that these URLs have not been verified. If the user selects a "verify external links" scanning option prior to initiating the scanning process, Astra will automatically scan these external URLs and update the map accordingly.

As part of the HTML parsing process, the scanning threads 122 detect any forms that are embedded within the HTML documents. (As described below, these forms are commonly used to allow the user to initiate back-end database queries which result in the dynamic generation of Web pages.) When a form is detected during an Automatic Update operation, the main thread 120 checks the Site Graph 114 to determine whether one or more datasets (captured by Astra as part of the Dynamic Scan feature) have been stored in association with the HTML document. For each dataset detected, Astra performs a dynamic page scan operation which involves the submission of the dataset to the URL specified within the form. This feature is further described below under the heading SCANNING AND MAPPING OF DYNAMICALLY-GENERATED PAGES.

Once the entire Web site has 113 been scanned, the Site Graph 114 represents the architecture of the Web site, including all of the detected URLs and links of the site. (If the user pauses the scanning process prior to completion, the Site Graph and VWD map represent a scanned subset of the Web site.) As described above, this data structure 114 is in the general form of a list of Node objects (one per URL) and Edge objects (one per link), with associated information attached as attributes of these objects. For each URL of the site, the information stored within the Site Graph typically includes the following: the URL type, the scanning status (OK, not found, inaccessible, unread, or access denied), the data and time of last modification, the URLs (addresses) of all incoming and outgoing links, the file size (if the URL was actually retrieved), an annotation, and the associated protocol.

Periodically during the scanning process, the main thread 120 executes a Visual Web Display routine which applies the Solar Layout method to the URLs and links of the Site Graph 114. (The term "routine," as used herein, refers to a functionally-distinguishable portion of the executable code of a larger program or software package, but is not intended to imply the modularity or capability of such code portion.) As indicated above, this method selects the links to be displayed within the site map (by selecting a span tree from the graph structure), and determines the layout and size for the display of the nodes (URLs) and non-hidden links of the map. The execution of this display routine results in modifications to the display attributes of the nodes (Node objects) and links (Edge objects) of the Site Graph 114 in accordance with the above-described layout and display principles. For example, for each link which is not present in the span tree, the visibility attribute of the link is set to "hidden." (As described below, link and node attributes are also modified in response to various user actions during the viewing of the map, such as the application of filters to the site map.)

In the preferred embodiment, the Visual Web Display routine is executed each time a predetermined threshold of new URLs have been scanned. Each time the routine is executed, the screen is automatically updated (in Visual Web Display format) to show the additional URLs that have been identified since the last execution of the routine. This allows the user to view the step-by-step generation of the site map during the scanning process. The user can selectively pause and restart the scanning process using respective controls on the Astra toolbar 46.

FIG. 10 illustrates the general decision process followed by a scanning thread 122 when a URL is scanned. This process implements the above-mentioned caching scheme for reducing unnecessary downloads of URLs and URL headers during Automatic Update operations. With reference to decision block 140, it is initially determined whether the URL has previously been scanned. If it has not been scanned, the thread either requests the file from the server (if the URL is an HTML file), or else requests the URL's header from the server, as illustrated by blocks 142-146. (URL headers are retrieved using the HEAD method of the HTTP protocol.) In either case, the scanning thread waits for the server to respond, and generates an appropriate status code (such as a code indicating that the URL was not found or was inaccessible) if a timeout occurs or if the server returns an error code, as indicated by block 150.

If, on the other hand, the URL has previously been mapped (block 140), the date/time of last modification stored in the Site Graph 114 (FIG. 9) is used to determine whether or not a retrieval of the URL is necessary. This is accomplished using standard argument fields of the HTTP "GET" method which enable the client to specify a "date/time of last modification" condition for the return of the file. With reference to blocks 158 and 160, the GET request is for the entire URL if the file is an HTML file (block 158), and is for the URL header if the file is a non-HTML file (block 160). Referring again to block 150, the thread then waits for the server response, and returns an appropriate status code if an error occurs.

As indicated by block 164, if an HTML file is returned as the result of the server request, the scanning thread parses the HTML and identifies any links within the file to other URLs. As indicated above, the main thread 120 launches additional scanning threads 122 to scan these URLs if any links are detected, with the exception that external links are not scanned unless a "verify external links" option has been selected by the user.

As indicated by the foregoing, the scanning process of the present invention provides a high degree of bandwidth efficiency by avoiding unnecessary retrievals of URLs and URL headers that have not been modified since the previous mapping, and by using multiple threads to scan the Web site.

VII. Scanning and Mapping of Dynamically-Generated Pages (FIGS. 11–15)

A feature of the invention which permits the scanning and mapping of dynamically-generated Web pages will now be described. By way of background, a dynamically-generated Web page (“dynamic page”) is a page that is generated “on-the-fly” by a Web site in response to some user input, such as a database query. Under existing Web technology, the user manually types-in the information (referred to herein as the “dataset”) into an embedded form of an HTML document while viewing the document with a Web browser, and then selects a “submit” type button to submit the dataset to a Web site that has back-end database access or real-time data generation capabilities. (Technologies which provide such Web server extension capabilities include CGI, Microsoft’s ISAPI, and Netscape’s NSAPI.) A Web server extension module (such as a CGI script) then processes the dataset (by, for example, performing a database search, or generating real-time data) to generate the data to be returned to the user, and the data is returned to the browser in the form of a standard Web page.

One deficiency in existing Web site mapping programs is that they do not support the automatic retrieval of dynamic pages. As a result, these mapping programs are not well suited for tracking changes to back-end databases, and do not provide an efficient mechanism for testing the functionality of back-end database search components. The present invention overcomes these deficiencies by providing a mechanism for capturing datasets entered by the user into a standard Web browser, and for automatically re-submitting such datasets during the updating of site maps. The feature of Astra which provides these capabilities is referred to as Dynamic Scan™.

FIG. 11 illustrates the general flow of information between components during a Dynamic Scan capture session, which can be initiated by the user from the Astra tool bar. Depicted in the drawing is a client computer 92 communicating with a Web site 113 over the Internet 110 via respective TCP/IP layers 108, 178. The Web site 113 includes a Web server application 112 which interoperates with CGI scripts (shown as layer 180) to generate Web pages on-the-fly. Running on the client computer 92 in conjunction with the Astra application 94 is a standard Web browser 170 (such as Netscape Navigator or Microsoft’s Internet Explorer), which is automatically launched by Astra when the user activates the capture session. As illustrated, the Web browser 170 is configured to use the Astra application 94 as an HTTP-level proxy. Thus, all HTTP-level messages (client requests) generated by the Web browser 170 are initially passed to Astra 94, which in-turn makes the client requests on behalf of the Web browser. Server responses (HTML pages, etc.) to such requests are returned to Astra by the client computer’s TCP/IP layer 108, and are then forwarded to the browser to maintain the impression of normal browsing.

During the Dynamic Scan capture session, the user types-in data into one or more fields 174 of an HTML document 172 while viewing the document with the browser 170. The HTML document 172 may, for example, be an internal URL which is part of a Web site map, or may be an external URL which has been linked to the site map for mapping purposes. When the user submits the form, Astra extracts the

manually-entered dataset, and stores this dataset (in association with the HTML document 172) for subsequent use. When Astra subsequently re-scans the HTML document 172 (during an Automatic Update of the associated site map), Astra automatically retrieves the dataset, and submits the dataset to the Web site 113 to recreate the form submission. Thus, for example, once the user has typed-in and submitted a database query in connection with a URL of a site map, Astra will automatically perform the database query (and map the results, as described below) the next time an Automatic Update of the map is performed.

With further reference to FIG. 11, when the Web site 113 returns the dynamic page during the capture session (or during a subsequent Automatic Update session), Astra automatically adds a corresponding node to the site map, with this node being displayed as being linked to the form page. (Screen displays taken during a sample capture session are shown in FIGS. 13–15 and are described below.) In addition, Astra parses the dynamic page, and adds respective nodes to the map for each outgoing link of the dynamic page. (In the default setting, these outgoing links are not scanned.) Astra also parses any static Web pages that are retrieved with the browser during the Dynamic Scan capture session, and updates the site map (by appending appropriate URL icons) to reflect the static pages.

FIG. 12 illustrates the general flow of information during a Dynamic Scan capture session, and will be used to describe the process in greater detail. Labeled arrows in FIG. 12 represent the flow of information between software and database components of the client and server computers. As will be apparent, certain operations (such as updates to the map structure 128) need not be performed in the order shown.

Prior to initiating the Dynamic Scan session, the user specifies a page 172 which includes an embedded form. (This step is not shown in FIG. 12). This can be done by browsing the site map with the Astra GUI to locate the node of a form page 172 (depicted by Astra using a special icon), and then selecting the node with the mouse. The user then initiates a Dynamic Scan session, which causes the following dialog to appear on the screen: YOU ARE ABOUT TO ENTER DYNAMIC SCAN MODE. IN THIS MODE YOU WORK WITH A BROWSER AS USUAL, BUT ALL YOUR ACTIONS (INCLUDING FORM SUBMISSIONS) ARE RECORDED IN THE SITE MAP. TO EXIT FROM THIS MODE, PRESS THE “STOP DYNAMIC SCAN” BUTTON ON THE MAIN TOOLBAR OR CHOOSE THE “STOP DYNAMIC SCAN” OPTION IN THE SCAN MENU.

When the user clicks on the “OK” button, Astra modifies the configuration of the Web browser 170 within the registry 182 of the client computer to set Astra 94 as a proxy of the browser, as illustrated by arrow A of FIG. 12. (As will be recognized by those skilled in the art, the specific modification which needs to be made to the registry 182 depends upon the default browser installed on the client computer.) Astra then launches the browser 170, and passes the URL (address) of the selected form page to the browser for display. Once the browser has been launched, Astra modifies the registry 182 (arrow B) to restore the original browser configuration. This ensures that the browser will not attempt to use Astra as a proxy on subsequent browser launches, but does disable the browser’s use of Astra as a proxy during the Dynamic Scan session.

As depicted in FIG. 12, the browser 170 retrieves and displays the form page 172, enabling the user to complete the form. In response to the submission by the user of the

form, the browser 170 passes an HTTP-level (GET or POST) message to Astra 94, as indicated by arrow C. This message includes the dataset entered by the user, and specifies the URL (address) of the CGI script or other Web server extension component 180 to which the form is addressed. Upon receiving this HTTP message, Astra displays the dialog "YOU ARE ABOUT TO ADD A DATASET TO THE CURRENT URL IN THE SITE MAP," and presents the user with an "OK" button and a "CANCEL" button.

Assuming the user selects the OK button, Astra extracts the dataset entered by the user and then forwards the HTTP-level message to its destination, as illustrated by arrow E. In addition, as depicted by arrow D, Astra stores this dataset in the Site Graph 114 in association with the form page 172. As described above, this dataset will automatically be retrieved and re-submitted each time the form page 172 is re-scanned as part of an Automatic Update operation. With reference to arrows F and G, when the Web server 112 returns the dynamic page 184, Astra 94 parses the page and updates the Site Graph 114 to reflect the page and any outgoing links of the dynamic page. (In this regard, Astra handles the dynamic page in the same manner as for other HTML documents retrieved during the normal scanning process.) In addition, as depicted by arrow H, Astra forwards the dynamic page 184 to the Web browser 170 (which in-turn displays the page) to maintain an impression of normal Web browsing.

Following the above sequence, the user can select the "stop dynamic scan" button or menu option to end the capture session and close the browser 170. Alternatively, the user can continue the browsing session and make additional updates to the site map. For example, the user can select the "back" button 186 (FIG. 14) of the browser to go back to the form page and submit a new dataset, in which case Astra will record the dataset and resulting page in the same manner as described above.

Although the system of the preferred embodiment utilizes conventional proxy technology to redirect and monitor the output of the Web browser 170, it will be recognized that other technologies and redirection methods can be used for this purpose. For example, the output of the Web browser could be monitored using conventional Internet firewall technologies.

FIGS. 13-15 are a sequence of screen displays taken during a Dynamic Scan capture session in which a simple database query was entered into a search page of the Infoseek™ search engine. FIG. 13, which is the first display screen of the sequence, illustrates a simple map 190 generated by opening a new map and then specifying http://www.infoseek.com/ as the URL. Displayed at the center of the map is the form page icon for the Infoseek™ search page. The 20 children 192 of the form page icon correspond to external links (i.e., links to URLs outside the infoseek.com domain), and are therefore displayed using the "not scanned" icon. (As described above, if the "verify external links" option of Astra is selected, Astra will verify the presence of such external URLs and update the map accordingly.)

FIG. 14 illustrates a subsequent screen display generated by starting a Dynamic Scan session with the Infoseek™ page selected, and then typing in the word "school" into the query field 194 of the page. (Intermediate displays generated by Astra during the Dynamic Scan session are omitted.) As illustrated in the figure, the Web browser comes up within a window 196, allowing the user to access the Astra controls and view the site map 190 during the Dynamic Scan session.

FIG. 15 illustrates the updated map 190' generated by Astra as a result of the FIG. 14 database query. The node

(icon) 200 labeled "titles" in the map represents the dynamic page returned by the Infoseek™ Web site, and is depicted by Astra as being linked to the Infoseek™ form page. A special "dynamic page" icon 200 is used to represent this newly-added node, so that the user can readily distinguish the node from nodes representing statically-generated pages. The children 204 of the dynamic page node 200 represent outgoing links from the dynamic page, and are detected by Astra by parsing the HTML of the dynamic page. In the present example, at least some of the children 204 represent search results returned by the Infoseek™ search engine and listed in the dynamic page.

As generally illustrated by FIG. 15, in which the children 204 of the dynamic page 200 are represented with Astra's "not scanned," Astra does not automatically scan the children of the dynamically-generated Web page during the Dynamic Scan session. To effectively scan a child page 204, the user can retrieve the page with the browser during the Dynamic Scan session, which will cause Astra to parse the child page and update the map accordingly.

Following the sequence illustrated by FIGS. 13-15, the user can, for example, save the map 190' to disk, which will cause the corresponding Site Graph 114 to be written to disk. If the user subsequently retrieves the map 190' and initiates an Automatic Update operation, Astra will automatically submit the query "school" to the Infoseek™ search engine, and update the map 190' to reflect the search results returned. (Children 204 which do not come up in this later search will not be displayed in the updated map.) By comparing this updated map to the original map 190' (either manually or using Astra's map comparison tool), the user can readily identify any new search result URLs that were returned by the search engine.

While the above-described Dynamic Scan feature is particularly useful in Web site mapping applications, it will be recognized that the feature can also be used in other types of applications. For example, the feature can be used to permit the scanning of dynamically-generated pages by general purpose Web crawlers. In addition, although the feature is implemented in the preferred embodiment such that the user can use a standard, stand-alone Web browser, it will be readily apparent that the feature can be implemented using a special "built-in" Web browser that is integrated with the scanning and mapping code.

VIII. Display of Filtered Maps (FIGS. 16-18)

The content, status and location filters of Astra provide a simple mechanism for allowing the user to focus-in on URLs which exhibit particular characteristics, while making use of the intuitive layout and display methods used by Astra for the display of site maps. To apply a filter, the user simply selects the corresponding filter button on the filter toolbar 47 while viewing a site map. (The specific filters that are available within Astra are listed above under the heading ASTRA GRAPHICAL USER INTERFACE.) Astra then automatically generates and displays a filtered version of the map. In addition to navigating the filtered map using Astra's navigation controls, the user can select the Visual Web Display button 73 (FIG. 16) to view the filtered map in Astra's VWD format. Combinations of the filters can be applied to the site map concurrently.

FIG. 16 illustrates the general display format used by Astra when a filter is initially applied to a site map. This example was generated by selecting the "hide OK URLs" button 220 on the filter toolbar 47 while viewing a site map similar to the map 30 of FIG. 1. As illustrated by the screen display, the selection of the filter causes Astra to generate a filtered map 30' which is in the form of skeletal view of the original map, with only the links and URLs of interest remaining.

As generally illustrated by FIG. 16, the filtered map 30' consists primarily of the following components of the original map 30: (i) the URLs which satisfy (pass through) the filter, (ii) the links to the URLs which satisfy the filter, and (iii) all "intermediate" nodes and links (if any) needed to maintain connectivity between the root (home page) URL and the URLs which satisfy the filter. (This display methodology is used for all of the filters of the filter toolbar 47, and is also used when multiple filters are applied.) In this example, the filtered map 30' thus consists of the home page URL, all URLs which have a scanning status other than "OK," and the links and nodes needed to maintain connectivity to the non-OK URLs. To allow the user to readily distinguish between the two types of URLs, Astra displays the URLs which satisfy the filter in a prominent color (such as red) when the filtered map is viewed in a zoomed-out mode. The general process used by Astra to generate the skeletal view of the filtered map is illustrated by FIG. 17.

While viewing the filtered map, the user can perform any of a number of actions, such as zoom in and out to reveal additional URL information, launch editor programs to edit the displayed URLs, and apply additional filters to the map. In addition, the user can select the Visual Web Display button 73 to display the filtered map in Astra's VWD format. To restore the hidden nodes and links to the map, the user clicks on the selected filter button to remove the filter.

FIG. 18 illustrates the filtered map of FIG. 16 following selection by the user of the VWD button 73. As generally illustrated by these two figures, the selection of the VWD button 73 causes Astra to apply the Solar Layout method to the nodes and links of the filtered map. In addition, to provide the user with a contextual setting for viewing the remaining URLs, Astra restores the visibility of selected nodes and links in the immediate vicinity of the URLs that satisfy the filter. As generally illustrated by node icons 226, 228 and 230 in FIG. 18, an icon color coding scheme is used to allow the user to distinguish the URL icons which satisfy the filter from those which do not, and to allow the user to distinguish URLs which have not been scanned.

IX. Tracking and Display of Visitor Activity (FIGS. 19 and 20)

An important feature of Astra is its ability to track user (visitor) activity and behavior patterns with respect to a Web site and to graphically display this information (using color coding, annotations, etc.) on the site map. In the preferred embodiment, this feature is implemented in-part by the Action Tracker plug-in, which gathers user activity data by retrieving and analyzing server log files commonly maintained by Web servers. Using this feature, Webmasters can view site maps which graphically display such information as: the most frequently-accessed URLs, the most heavily traveled links and paths, and the most popular site entry and exit points. As will be appreciated by those skilled in the art, the ability to view such information in the context of a site map greatly simplifies the task of evaluating and maintaining Web site effectiveness.

By way of background, standard Web servers commonly maintain server access log files ("log files") which include information about accesses to the Web site by users. These files are typically maintained in one of two standard formats: the HTTP Server Access Log File format, or the HTTP Server Referrer Log File format (Both of these formats are commonly used by Web servers available from Microsoft, Netscape, and NSCA, and both formats are supported by Astra.) Each entry (line) in a log file represents a successful access to the associated Web site, and contains various information about the access event. This information nor-

mally includes: the path to the accessed URI, an identifier of the user (typically in the form of an IP address), and the date and time of the access. Each log file stored on a physical server typically represents some window of time, such as one month.

In accordance with the invention, Astra uses the information contained within a log file in combination with the associated site graph to determine probable paths taken by visitors to the Web site. (The term "visitor" is used herein to distinguish the user of the Web site from the user of Astra, but is not intended to imply that the Web site user must be located remotely from the Web site.) This generally involves using access date/time stamps to determine the chronological sequence of URLs followed by each visitor (on a visitor-by-visitor basis), and comparing this information against link information stored in the site map (i.e., the Site Graph object 114) to determine the probable navigation path taken between the accessed URLs. (This method is described in more detail below.) By determining the navigation path followed by a visitor, Astra also determines the site entry and exits points taken by the visitor and all of the links traversed by the visitor. By performing this method for each visitor represented in the log file and appropriately combining the information of all of the visitors, Astra generates statistical data (such as the number of "hits" or the number of exit events) with respect to each link and node of the Web site, and attaches this information to the corresponding Node and Edge objects 115, 116 (FIG. 8) of the Site Graph 114.

To activate the Action Tracker feature, the user selects the Action Tracker option from the TOOLS menu while viewing a site map. The user is then presented with the option of either retrieving the server log file or loading a previously-saved Astra Activity File. Astra Activity Files are compressed versions of the log files generated by Astra and stored locally on the client machine, and can be generated and saved via controls within the Action Tracker controls. Astra also provides an option which allows the user to append a log file to an existing Astra Activity file, so that multiple log files can be conveniently combined for analysis purposes. Once the Activity File or server log file has been loaded, an Action Tracker dialog box (FIG. 19) opens which provides controls for allowing the user to selectively display different types of activity data on the map.

FIG. 19 illustrates the general display format used by the Action Tracker plug-in to display activity levels on the links of a site. As illustrated by the screen display, the links between URLs are displayed using a color-coding scheme which allows the user to associate different link colors (and URL icon colors) with different relative levels of user activity. As generally illustrated by the color legend, three distinct colors are used to represent three (respective) adjacent ranges of user activity.

In the illustrated display mode (uncolored links hidden, uncolored URLs not hidden), all of the URLs of the site map are displayed, but the only links that are displayed are those which satisfy a user-adjustable minimum activity threshold. Each visible link is displayed as a one-way arrow (indicating the link direction), and includes a numerical annotation indicating the total number of hits revealed by the log or activity file. The number of hits per URL can be viewed in List View mode in a corresponding column. As can be seen from an observation of the screen display, the displayed links include links which do not appear in the Visual Web Display view of the map.

With further reference to FIG. 19, a slide control 240 allows the user to adjust the "hits" thresholds corresponding

to each of the three colors. By clicking and dragging the slide control, the user can vary the number of displayed links in a controllable manner to reveal different levels of user (visitor) activity. This feature is particularly useful for identifying congested links, which can be remedied by the addition of appropriate data redundancies.

FIG. 20 illustrates the general process used by the Action Tracker plug-in to detect the link activity data (number of hits per link) from the log file. The displayed flow chart assumes that the log file has already been retrieved, and that the attribute "hits" has been defined for each link (Edge object) of the Site Graph and set to zero. As illustrated by the flow chart, the general decision process is applied line-by-line to the log file (each line representing an access to a URL) until all of the lines have been processed. With reference to blocks 250 and 252, each time a new line of the log file is ready, it is initially determined whether or not the log file reflects a previous access by the user to the Web site. This determination is made by searching for other entries within the log file which have the same user identifier (e.g., IP address) and an earlier date/time stamp.

Blocks 254 and 256 illustrate the steps that are performed if the user (visitor) previously visited the site. Initially, the Site Graph is accessed to determine whether a link exists from the most-recently accessed URL to the current URL, as indicated by decision block 254. If such a link exists, it is assumed that the visitor used this link to get to the current URL, and the usage level ("hits" attribute) of the identified link is incremented by one. If no such link is identified between the most-recently accessed URI and the current URL, an assumption is made that the user back-tracked along the navigation path (by using the "BACK" button of the browser) before jumping to the current URL. Thus, decision step 254 is repeated for each prior access by the user to the site, in reverse chronological order, until either a link to the current URL is identified or all of the prior accesses are evaluated. If a link is detected during this process, the "hits" attribute of the link is incremented.

As indicated by block 258, the above process continues on a line-by-line basis until all of the lines of the log file have been processed. Following the execution of this routine, the "hits" attribute of each link represents an approximation (based on the above assumptions) of the number of times the link was traversed during the time frame represented by the log file.

As will be apparent, the general methodology illustrated by the FIG. 20 flow chart can be used to detect a variety of different types of activity information, which can be superimposed on the site map (by modifying node and link display attributes) in the same general manner as described above. The following are examples of some of the types of activity data that can be displayed, together with descriptions of several features of the invention which relate to the display of the activity data:

Exit Points. Exit points are deduced from the log file on a visitor-by-visitor basis by looking for the last URL accessed by each visitor, and by looking for large time gaps between consecutive accesses to the site. An "exits" attribute is defined for each node to keep track of the total number of exit events from each node. The color-coding scheme described above is then used to allow the user to controllably display different thresholds of exit events.

Usage Zones. When viewing a large site map in its entirety (as in FIG. 1), it tends to be difficult to identify individual URL icons within the map. This in-turn makes it difficult to view the color-coding scheme used by the Action Tracker plug-in to display URL usage levels. The Usage

Zones™ feature alleviates this problem by enlarging the size of the colored URL icons (i.e., the icons of nodes which fall within the predetermined activity level thresholds) to a predetermined minimum size. (This is accomplished by increasing the "display size" attributes of these icons.) If these colored nodes are close together on the map, the enlarged icons merge to form a colored zone on the map. This facilitates the visual identification of high-activity zones of the site.

Complete Path Display. With this feature, the complete path of each visitor is displayed on the map on a visitor-by-visitor basis, with the visitor identifier and the URL access time tags displayed in the List View window 78 (FIG. 4). This feature permits fine-grain inspection of the site usage data, which is useful, for example, for analyzing security attacks and studying visitor behavior patterns.

Log Filters. Because server access log files tend to be large, it is desirable to be able to filter the log file and to display only certain types of information. This feature allows the user to specify custom filters to be applied to the log file for purposes of limiting the scope of the usage analysis. Using this feature, the user can, for example, specify specific time and date ranges to monitor, or limit the usage analysis to specific IP addresses or domains. In addition, the user can specify a minimum visit duration which must be satisfied before the Action Tracker will count an access as a visit.

X. Map Comparison Tool (FIG. 21)

FIG. 21 illustrates a screen display generated using Astra's Change Viewer™ map comparison tool. As illustrated by the screen display, the comparison tool generates a comparison map 268 which uses a color-coding scheme to highlight differences between two site maps, allowing the user to visualize the changes that have been made to a Web site since a prior mapping of the site. Using the check boxes within the Change Viewer dialog box 270, the user can selectively display the following: new URLs and links, modified URLs, deleted URLs and links, and unmodified URLs and links. As illustrated, each node and link of the comparison map is displayed in one of four distinct colors to indicate its respective comparison status: new, modified, deleted, or unmodified.

To compare two maps, the user selects the "Compare Maps" option from the TOOLS menu while viewing the current map, and then specifies the filename of the prior map. Astra then performs a node-by-node and link-by-link comparison of the two map structures (Site Graphs) to identify the changes. This involves comparing the "URL" attributes of the associated Node and Edge objects to identify URLs and links that have been added and deleted, and comparing the "date/time of last modification" attributes of like Node objects (i.e., Node objects with the same "URL" attribute) to identify URLs that have been modified. During this process, a comparison map data structure is generated which reflects the comparison of the two maps, using color attributes to indicate the comparison outcomes (new, modified, deleted or unmodified) of the resulting nodes and links. Once the comparison map data structure has been generated, Astra applies the Solar Layout method to the structure and displays the comparison map 268 in Astra's VWD format. (The user can also view the comparison map in Astra's "incoming links" and "outgoing links" display modes.) The user can then adjust the "show" settings in the dialog box 270, which causes Astra to traverse the comparison map data structure and adjust the visibility attributes according to the current settings.

XI. Link Repair Plug-in (FIG. 22) FIG. 22 illustrates the operation of Astra's Link Doctor plug-in. To access this

feature, the user selects the "Link Doctor" option from the TOOLS menu while viewing a site map. The Link Doctor dialog box **284** then appears with a listing (in the "broken links" pane **286**) of all of the broken links (i.e., URLs of missing content objects) detected within the site map. (Astra detects the missing links by searching the Site Graph for Node objects having a status of "not found.") When the user selects a URL from the broken links pane (as illustrated in the screen display), Astra automatically lists all of the URLs which reference the missing content object in the "appearing in" pane **288**. This allows the user to rapidly identify all of the URLs (content objects) that are directly affected by the broken link.

In addition to listing all of the referencing URLs in the "appearing in" pane **288**, Astra generates a graphical display (in Astra's "incoming links" display mode) which shows the selected (missing) URL **290** and all of the URLs **292** which have links to the missing URL. In this example, the missing URL is a GIF file which is embedded within eight different HTML files **292**. From the display shown in FIG. 22, the user can select one of the referencing nodes **292** (by either clicking on its icon or its listing in the "appearing in" pane), and then select the "Edit" button **296** to edit the HTML document and eliminate the reference to the missing file.

XII. Conclusion

While certain preferred embodiments of the invention have been described, these embodiments have been presented by way of example only, and are not intended to limit the scope of the present invention. For example, although the present invention has been described with reference to the standard protocols, services and components of the World Wide Web, it should be recognized that the invention is not so limited, and that the various aspects of the invention can be readily applied to other types of web sites, including intranet sites and network sites that use proprietary client-server protocols. In addition, while the Web site mapping and map navigation features of the invention have been described in the context of a preferred Web site management and analysis program, these features can also be included within browser programs to facilitate the visualization and navigation of Web sites by visitors. Accordingly, the breadth and scope of the present invention should be defined only in accordance with the following claims and their equivalents.

In the claims which follow, reference characters used to designate claim steps are provided for convenience of description only, and are not intended to imply any particular order for performing the steps.

What is claimed is:

1. A computer program for facilitating the visualization of a web site, the computer program comprising, on a computer readable medium:

executable scanning code which scans the web site and generates a representation of the web site within a memory of a computer, the web site representation including representations of content objects and links of the web site;

executable hierarchy identification code which reduces the web site representation generated by the scanning code to a hierarchical tree representation in which each content object corresponds to a respective node of the tree; and

executable mapping code which generates and displays a graphical map of the web site on a display screen of the computer based on the tree representation, the mapping code configured to graphically represent the content objects of the web site as respective content object icons, the mapping code comprising a layout routine

which positions and interconnects the content object icons within the map based upon parent-child relationships of the content objects within the tree, the layout routine positioning the icons such that icons of parent content objects ("parents") are spatially grouped together with icons of the respective immediate children content objects ("children") of said parents to form a hierarchical arrangement of parent-child icon clusters in which the children generally surround their respective immediate parents over multiple parent-child levels.

2. The computer program according to claim 1, wherein the hierarchy identification code is configured to apply a shortest path span tree algorithm to the web site representation.

3. The computer program according to claim 1, wherein the layout routine implements a layout method which positions the icons of a plurality of immediate children of a common parent at angular increments around the icon of the parent the layout routine is configured to recursively apply the layout method at least once for each parent content object that has multiple children.

4. The computer program according to claim 3, wherein the layout routine automatically positions the respective icons of the plurality of immediate children away from the icon of the parent by substantially equal radial distances when all of the immediate children correspond to leaf nodes of the tree, said radial distances being directly proportional to a total number of the plurality of immediate children.

5. The computer program according to claim 1, wherein the layout routine is configured to select the display sizes of the content object icons within the map in direct proportion to respective numbers of outgoing links, so that web pages with relatively large numbers of outgoing links are represented by relatively large icons throughout the map.

6. The computer program according to claim 1, wherein the mapping code is configured to graphically represent each link of the tree as a respective line which connects an icon of a parent to an icon of an immediate child of the parent within the map.

7. The computer program according to claim 6, wherein the mapping code is configured to position an icon of a home page content object of the web site generally at the center of the map such that the hierarchical arrangement of icon clusters extends outward from the icon of the home page content object.

8. The computer program according to claim 1, further comprising a navigation interface which allows a user of the computer to interactively navigate the graphical map, the navigation interface including magnification controls for allowing the user to select a portion of the map and view the selected portion in a magnified, zoomed-in display mode.

9. The computer program according to claim 8, wherein the navigation interface presents the user with a contextual view of a navigation position within the map by displaying, within a window of the display screen, a zoomed-out view of the map together with a graphical indication of the portion being viewed in the zoomed-in mode.

10. The computer program according to claim 8, wherein the navigation interface includes executable code for automatically exposing object-specific information within the map in response to magnification of the map via the magnification controls.

11. The computer program according to claim 10, wherein the code for automatically exposing object-specific information comprises code for automatically displaying a textual annotation of a content object when an icon of the

content object is magnified beyond a predetermined magnification level.

12. The computer program according to claim 1, wherein the scanning code includes code for automatically scanning a dynamically-generated web page using a data set captured during a prior web site browsing session, the dynamically-generated web page being a page which is generated dynamically in response to data submitted by a user.

13. The computer program according to claim 1, further comprising map comparison code which compares a current map of the web site to a previously-generated map of the web site, and generates a comparison map which graphically represents differences between the current and previously-generated maps, to thereby allow a user to view changes that have been made to the web site.

14. The computer program according to claim 1, further comprising an application program interface ("API"), the API including methods which allow application programs to superimpose site data onto the graphical map.

15. The computer program according to claim 14, further comprising an application program which makes calls to the API to superimpose web site usage data onto the map.

16. The computer program according to claim 1, wherein the layout routine positions icons of children around icons of their respective parents over a range of substantially 360 degrees over multiple parent-child levels.

17. The computer program according to claim 1, further comprising filtering code which provides a user option to apply a content filter for hiding icons of content objects of a predetermined type.

18. The computer program according to claim 1, wherein the mapping code provides a user option to re-apply the layout routine to a filtered map generated by the filtering code.

19. The computer program according to claim 1, wherein the layout routine positions a largest satellite cluster of a first parent using the same entry angle as used to position the first parent, so that the largest satellite, the first parent, and a parent of the first parent fall generally along the same line.

20. A method of facilitating the visualization of a web site by a user, the method comprising the computer-implemented steps of:

- (a) scanning the web site to generate a representation of the web site;
- (b) generating a graphical site map of the web site on a display screen of a computer using the representation generated in step (a) to enable the user to view a structural representation of the web site, the site map comprising a plurality of interconnected icons which represent content objects of the web site; and
- (c) providing a navigation interface which allows the user to interactively navigate the site map, the navigation interface including a zoom control which allows the user to interactively zoom in and out on user-selected portions of the map.

21. The method according to claim 20, wherein step (b) comprises displaying textual annotations within the site map in association with the icons.

22. The method according to claim 20, wherein step (b) comprises automatically adjusting sizes of at least the icons such that the entire site map fits on the display screen.

23. The method according to claim 22, wherein the navigation interface concurrently displays first and second views of the site map when the user views the site map in a zoomed-in mode, the first view being a magnified view of a user-selected portion of the site map, the second view being a perspective view of the site map together with a graphical indication of the user-selected portion.

24. The method according to claim 20, wherein step (b) comprises displaying the icons within the map such that groups of sibling icons generally surround their respective parent icons.

25. A method of facilitating the visualization of a web site, the method comprising the computer-implemented steps of: identifying a hierarchy of content objects of the web site; and

generating a graphical map of the hierarchy on a display screen, the graphical map comprising interconnected content object icons that represent respective content objects, the step of generating comprising:

- (a) identifying a parent content object ("parent") of the hierarchy that has multiple immediate children content objects ("children");
- (b) identifying the multiple immediate children of the parent identified in step (a);
- (c) positioning respective icons of the immediate children identified in step (b) around an icon of the parent identified in step (a) such that the icons of the children generally surround the icon of the parent within the map; and
- (d) repeating steps (a), (b) and (c) for each of a plurality of additional parents of the hierarchy that have multiple immediate children.

26. The method according to claim 25, wherein the step of identifying a hierarchy comprises generating a tree representation of the content objects and links of the web site.

27. The method according to claim 25, wherein step (c) is performed such that the icons of the immediate children are spaced at substantially equal angular intervals around the icon of the parent when none of the immediate children has a child.

28. The method according to claim 25, wherein step (c) is performed such that all of the icons of the immediate children are radially spaced away from the icon of the parent by substantially the same distance when none of the immediate children has a child.

29. The method according to claim 28, wherein the distance is directly proportional to the number of immediate children.

30. The method according to claim 25, wherein the step of generating the graphical map further comprises selecting icon display sizes such that the icon of each parent that has no grandchildren is directly proportional in size to the number of immediate children of that parent.

31. The method according to claim 25, wherein the step of generating a graphical map further comprises connecting, within the map, the respective icon of each child of the hierarchy to the icon of the child's immediate parent, to graphically represent the links of the hierarchy.

32. The method according to claim 31, wherein the step of identifying the hierarchy comprises:

generating a graph data structure which represents at least some of the content objects and links of the web site; and

applying a span tree algorithm to the graph data structure to generate a hierarchical tree representation of the web site, the step of applying causing a plurality of the links of the graph data structure to be omitted from the graphical map.

33. The method according to claim 32, wherein the step of applying the span tree algorithm comprises assigning a home page content object of the web site as a root node of the tree.

34. The method according to claim 25, wherein steps (a)-(d) are performed such that an icon of a home page

35

content object is positioned generally at the center of the map, and such that respective icons of a plurality of immediate children of the home page content object are positioned around the home page icon over a range of substantially 360 degrees.

35. The method according to claim 25, wherein step (c) comprises calculating angular spacings for positioning the icons of the children identified in step (b) around the icon of the parent identified in step (a).

36. The method according to claim 25, further comprising the steps of:

presenting a user with a content filter which enables the user to selectively hide icons of content objects of a predetermined type within the map; and

automatically hiding icons of content objects of the predetermined type in response to user actuation of the content filter, to generate a filtered map.

37. The method according to claim 36, wherein the step of automatically hiding is performed without substantially changing a general layout of content object icons within the map, so that the filtered map is presented to the user as a skeletal view of the original map.

38. The method according to claim 37, further comprising the steps of:

presenting the user with a layout control which enables the user to selectively apply a layout method to the filtered the map; and

reformatting the filtered map on the display screen in response to user actuation of the layout control to generate a non-skeletal view of the filtered map, the step of reformatting comprising reapplying steps (a)–(d) to the content objects of the filtered map.

39. The method according to claim 38, wherein the step of reformatting comprises modifying respective sizes of at least some of the content object icons of the filtered map.

40. The method according to claim 25, further comprising the steps of:

presenting a user with a variable zoom control which enables the user to magnify a user-selected portion of the graphical map; and

enlarging the user-selected portion of the map on the display screen in response to user actuation of the zoom control.

41. The method according to claim 40, wherein the step of enlarging comprises displaying, within the map, content object information that is not displayed when the map is viewed in a zoomed-out display mode.

42. The method according to claim 40, further comprising the step of presenting the user with a contextual view of a navigation position within the map by displaying, within a window of the display screen, a zoomed-out view of the map together with a graphical indication of the user-selected portion.

43. The method according to claim 25, further comprising positioning a largest satellite cluster of a parent using the same entry angle as used to position the parent.

44. A method of generating a graphical map of a tree data structure on a display screen, the tree structure comprising a plurality of objects, the method comprising the computer-implemented steps of:

(a) representing the objects as respective icons within the map;

(b) identifying, within the tree structure, a parent object (“parent”) that has multiple immediate children objects (“children”);

(c) identifying the multiple immediate children of the parent identified in step (b);

36

(d) positioning the respective icons of the immediate children identified in step (c) around the icon of the parent identified in step (b) on the display screen such that the icons of the children surround the icon of the parent within the map;

(e) displaying a respective parent-child connection within the map between each child icon and the parent icon positioned in step (d); and

(f) repeating steps (b), (c), (d) and (e) for each of a plurality of additional parents of the tree structure that have multiple immediate children.

45. The method according to claim 44, wherein step (d) comprises calculating angular spacings for positioning the icons of the children identified in step (c) around the icon of the parent identified in step (b).

46. The method according to claim 44, wherein steps (b)–(f) are performed such that icons of children objects are positioned around icons of their respective parents over a range of substantially 360 degrees.

47. The method according to claim 44, wherein step (d) is performed such that the icons of the immediate children are spaced at substantially equal angular intervals around the icon of the parent when none of the immediate children has a child.

48. The method according to claim 44, wherein step (d) is performed such that all of the icons of the immediate children are radially spaced away from the icon of the parent by substantially the same distance when none of the immediate children has a child.

49. The method according to claim 48, wherein the distance is directly proportional to the number of immediate children.

50. The method according to claim 44, further comprising selecting icon display sizes such that the icon of each parent object that has no grandchildren objects is directly proportional in size to the number of immediate children of that parent object.

51. The method according to claim 44, wherein the tree data structure represents an arrangement of content objects of a web site.

52. The method according to claim 44, wherein the tree data structure represents a locally-stored arrangement of files and file directories.

53. The method according to claim 44, further comprising the steps of:

presenting a user with a variable zoom control which enables the user to selectively zoom-in on portions of the graphical map; and

enlarging a portion of the map on the display screen in response to user actuation of the zoom control.

54. The method according to claim 53, further comprising the step of presenting the user with a contextual view of a navigation position within the map by displaying, within a window of the display screen, a zoomed-out view of the map together with a graphical indication of the portion.

55. The method according to claim 44, further comprising positioning a largest satellite cluster of a parent using the same entry angle as used to position the parent.

56. A method of representing a hierarchical node-link structure on a display screen, comprising:

(a) identifying a plurality of parent nodes of the structure that have multiple children nodes, including parent nodes at multiple different levels of the hierarchical structure;

(b) positioning a parent node identified in step (a) on the display screen, and positioning the children nodes of

37

the parent node around the parent node over an angular range which exceeds 180 degrees; and

- (c) repeating step (b) recursively for multiple additional parent nodes identified in step (a), including parent nodes at multiple different levels of the hierarchical structure;

wherein the method produces a map which comprises a hierarchical arrangement of parent-child node clusters.

57. The method of claim 56, further comprising positioning a largest satellite node cluster of a first parent node on the display screen using the same entry angle as used to position the first parent node, so that the largest satellite cluster, the first parent node, and a parent of the first parent fall generally along the same line.

58. A method of graphically representing a web document on a display screen, the web document comprising a main document portion which comprises a plurality of links to a plurality of respective document components, the method comprising the computer-implemented steps of:

representing the main document portion as a first icon on the display screen;

representing the plurality of document components as a plurality of respective additional icons that are positioned on the display screen generally around the first icons the step of representing comprising spacing each of the additional icons away from the first icon by substantially the same distance, and calculating said distance based at least in-part on a total number of said document components; and

representing each of the plurality of links as a respective interconnection between the first icon and a respective one of the additional icons.

59. The method according to claim 58, wherein the step of representing the plurality of document components comprises positioning the additional icons circularly around the first icon on the display screen.

60. The method according to claim 59, wherein the step of positioning comprises spacing the additional icons around the first icon at substantially equal angular intervals.

61. The method according to claim 58, wherein the step of representing the plurality of document components com-

38

prises positioning the additional icons circularly around the first icon on the display screen over an angular range of less than 360 degrees.

62. The method according to claim 61, further comprising:

representing a link to the main document portion as at least a line which extends from the first icon on the display screen; and

selecting the angular range such that the angular range does not encompass the line.

63. The method according to claim 62, wherein the step of representing the plurality of document components comprises spacing the additional icons apart from one another at substantially equal angular intervals within the angular range.

64. The method according to claim 58, further comprising modifying icon display sizes so that the first icon is larger than any of the plurality of additional icons.

65. The method according to claim 58, further comprising displaying, in close proximity to each additional icon, a respective textual annotation which identifies at least a content type of the respective document component.

66. The method according to claim 58, further comprising the computer-implemented steps of:

parsing the main document component to identify the plurality of links and the plurality of document components;

attempting to access each of the plurality of document components identified during the step of parsing, and representing a document component for which the attempt to access is unsuccessful with a special icon that represents a failed access attempt.

67. The method according to claim 58, wherein the main document component further comprises a hyperlink to a second web document, and the method further comprises displaying the hyperlink as an interconnection between the first icon and an icon that represents the second web document.

* * * * *

APPENDIX 10



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/689,927	10/12/2000	Abha Lessing	16517-3	7458

7590 07/08/2004
 Clifford W Browning
 Woodard Emhardt Naughton Moriarty & McNett
 Bank One Center Tower
 111 Monument Circle Suite 3700
 Indianapolis, IN 47304-5137

RECEIVED
 JUL 13 2004

Woodard, Emhardt, Moriarty,
 McNett & Henry LLP

EXAMINER

CAMPBELL, JOSHUA D

ART UNIT	PAPER NUMBER
2178	5

DATE MAILED: 07/08/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

8.5.04
 10.5.04
ENTERED
 1.2.05

Office Action Summary

Application No.

09/689,927

Applicant(s)

LESSING ET AL.

Examiner

Joshua D Campbell

Art Unit

2178

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 12 October 2000.
- 2a) This action is FINAL. 2b) This action is non-final.
- 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-97 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) Claim(s) _____ is/are allowed.
- 6) Claim(s) 1-97 is/are rejected.
- 7) Claim(s) _____ is/are objected to.
- 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
- 10) The drawing(s) filed on 12 October 2000 is/are: a) accepted or b) objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) All b) Some * c) None of:
- Certified copies of the priority documents have been received.
 - Certified copies of the priority documents have been received in Application No. _____.
 - Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) Notice of References Cited (PTO-892)
- 2) Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date 3 and 4.
- 4) Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____.
- 5) Notice of Informal Patent Application (PTO-152)
- 6) Other: _____.

DETAILED ACTION

1. This action is responsive to communications: Application filed on 10/12/2000 and IDSs filed 10/12/2000 and 05/13/2002.
2. Claims 1-97 are pending in this case. Claims 1, 11, 12, 22, 23, 33, 34, 54, 74, and 94 are independent claims.

Drawings

3. The drawings were received on 10/12/2000. These drawings are accepted.

Claim Rejections - 35 USC § 102

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

5. Claims 1-7, 12-18, and 23-29 are rejected under 35 U.S.C. 102(b) as being anticipated by Dolan et al. (hereinafter Dolan, US Patent Number 5,963,208, issued on October 5, 1999).

Regarding independent claim 1, Dolan discloses a method in which a predefined portion is displayed in a first display region (Figure 1, 104 of Dolan). Dolan also discloses a method in which a point is displayed in a multidimensional space for said predefined portion (Figure 1, 102 of Dolan).

Art Unit: 2178

Regarding dependent claims 2 and 3, Dolan discloses a method in which a second point is displayed in the display region and information regarding the second point may be shown in the second display region (Figure 1, 102 of Dolan).

Regarding dependent claim 4-7, Dolan discloses a method in which any number of points and any number of axis derived from the first point may be displayed in two display regions, making it possible to navigate among the points and having the ability to return to the first (anchor) point (Figure 1, 102 and 104 of Dolan).

Regarding independent claim 12 and dependent claims 13-18, the claims incorporate substantially similar subject matter as claims 1-7. Thus, the claims are rejected along the same rationale as claims 1-7.

Regarding independent claim 23 and dependent claims 24-29, the claims incorporate substantially similar subject matter as claims 1-7. Thus, the claims are rejected along the same rationale as claims 1-7.

6. Claims 34-35, 37-38, 43-44, 47, 54-55, 57-58, 63-64, 67, 74-75, 77-78, 83-84, and 87 are rejected under 35 U.S.C. 102(b) as being anticipated by Fay (US Patent Number 5,892,513, issued on April 6, 1999).

Regarding independent claim 34, Fay discloses a method in which predefined portions are stored in terminal nodes (Figure 1 and 2 of Fay). Fay also discloses a method in which one or more higher-level nodes are used to organize the terminal nodes, in which all nodes include the identity of a parent node, a position indicator, and an identifier (Figure 3 and 4 of Fay).

Art Unit: 2178

Regarding dependent claim 35, Fay discloses a method in which the highest-level node has a null parent identity (column 3, line 46-column 4, line 23 of Fay).

Regarding dependent claim 37, Fay discloses a method in which a position indicator indicates a position of a higher-level node relative to a sibling node (Figure 3 and 4, "link clusters" and "parent nodes" of Fay).

Regarding dependent claim 38, Fay discloses a method in which a modified portion can be stored at a node (column 2, line 60-column 3, 46 of Fay).

Regarding dependent claim 43, Fay discloses a method in which all nodes include the identity of a parent node, a position indicator, and an identifier (Figure 3 and 4 of Fay).

Regarding dependent claim 44, Fay discloses a method in which a position indicator indicates a position of a terminal node relative to a sibling node (Figure 3 and 4, "link clusters" and "child nodes" of Fay).

Regarding dependent claim 47, Fay discloses a method in which a modified and predefined portions can be stored at the terminal nodes (column 2, line 60-column 3, 46 of Fay).

Regarding independent claim 54 and dependent claims 55, 57-58, 63-64, and 67, the claims incorporate substantially similar subject matter as claims 34-35, 37-38, 43-44, and 47. Thus, the claims are rejected along the same rationale as claims 34-35, 37-38, 43-44, and 47.

Regarding independent claim 74 and dependent claims 75, 77-78, 83-84, and 87, the claims incorporate substantially similar subject matter as claims 34-35, 37-

Art Unit: 2178

38, 43-44, and 47. Thus, the claims are rejected along the same rationale as claims 34-35, 37-38, 43-44, and 47.

Claim Rejections - 35 USC § 103

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

9. Claims 9, 20, and 31 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dolan et al. (hereinafter Dolan, US Patent Number 5,963,208, issued on October 5, 1999).

Regarding dependent claims 9, 20, and 31, Dolan discloses a method for providing easy navigation in a document based on document portions. Dolan does not

Art Unit: 2178

disclose that the document in question is a provision of legislation. However, as shown by the inventor on Page 1, lines 8-22 of the specification a provision of legislation is a document that is hard to navigate that contains many sections (Acts). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have used the method of Dolan with a provision of legislation because it would have provided an easier method of document navigation.

10. Claims 8, 10-11, 19, 21-22, 30, and 32-33 are rejected under 35 U.S.C. 103(a) as being unpatentable over Dolan et al. (hereinafter Dolan, US Patent Number 5,963,208, issued on October 5, 1999) as applied to claim 2 above, and further in view of Weinberg et al. (hereinafter Weinberg, US Patent Number 6,144,962, filed on April 11, 1997).

Regarding dependent claim 8, Dolan does not disclose a method in which time based versions of documents are shown on the axis. However, Weinberg discloses a method in which time based versions of documents and document portions are shown as part of a document map on an axis (Figure 1 and column 23, line 39-column 24, line 25 of Weinberg). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the methods of Dolan and Weinberg because it would have allowed for a user to view up-to-date information at all times.

Regarding dependent claim 10, Dolan does not disclose a method in which an axis represents search criteria and results corresponding to a portion. However, Weinberg discloses a method in which the portions on the axis can exist as the results

Art Unit: 2178

of searches or queries (column 17, line 54-column 18, line 54 of Weinberg). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the methods of Dolan and Weinberg because it would have allowed users easier navigation of Internet search related documents.

Regarding dependent claim 11, Dolan discloses a method in which at least one based point, at least a first axis depending from base point, and a further point and an axis derived from the said base point are all displayed (Figure 1, 102 of Dolan). Dolan also discloses a method in which the user may navigate the multidimensional space and return to the said base point when necessary (Figure 1, 102 of Dolan). Dolan does not disclose a method in which a view comprises at least two anchor sets or the ability to adjust the view point to become the new base point. However, Weinberg discloses a method in which multiple anchor sets may be viewed, and that at any time the based point may be changed to a different view point (Figures 1-3 of Weinberg). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the methods of Dolan and Weinberg because it would have allowed navigation of multiple document structures from one view space.

Regarding dependent claims 19 and 21-22, the claims incorporate substantially similar subject matter as claims 8 and 10-11. Thus, the claims are rejected along the same rationale as claims 8 and 10-11.

Regarding dependent claims 30 and 32-33, the claims incorporate substantially similar subject matter as claims 8 and 10-11. Thus, the claims are rejected along the same rationale as claims 8 and 10-11.

Art Unit: 2178

11. Claims 36, 45, 50-53, 56, 65, 70-73, 76, 85, and 90-97 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fay (US Patent Number 5,892,513, issued on April 6, 1999) as applied to claim 34 above, and further in view of McIntosh (US Patent Number 6,185,576, filed on October 21, 1998).

Regarding dependent claim 36, Fay does not disclose a method in which a portion is associated with a time during which it is valid. However, McIntosh discloses a method in which documents are associated with a scope, which defines when they are valid (column 31, line 21-column 46, line 45 of McIntosh). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the methods of Fay and McIntosh because it would have allowed easier management of time sensitive material.

Regarding dependent claim 46, Fay discloses a method in which a node is identified by its identifier (Figure 3 and 4). Fay does not disclose a method in which the terminal node is identified based on a combination of identifier and scope. McIntosh discloses a method in which documents are associated with a scope, which defines when they are valid, thus blocking access when the validity has expired, which provides navigation based both on ID and scope (column 31, line 21-column 46, line 45 of McIntosh). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the methods of Fay and McIntosh because it would have provided preventative measures blocking users from viewing expired materials.

Art Unit: 2178

Regarding dependent claim 50, Fay discloses a method for providing easy navigation in a document based on document portions. Fay does not disclose that the document in question is commentary. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have used the method of Fay with a commentary document because it was well known at the time the invention was made that documents could contain commentary and the method of Fay is directed to all general document.

Regarding dependent claim 51-53 and 95, Fay does not disclose a method in which a portion contains a scope that includes a start date, end date, and an update date that is in between the start and end date. However, McIntosh discloses a method in which a document has a scope, which includes a start date, an expiration (end) date, and an update date that exists in between the start and expiration date (column 31, line 21-column 46, line 45 of McIntosh). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the methods of Fay and McIntosh because it would have allowed easier management of time sensitive material.

Regarding dependent claims 56, 65, 70-73, and 96, the claims incorporate substantially similar subject matter as claims 36, 45, 50-53, and 95. Thus, the claims are rejected along the same rationale as claims 36, 45, 50-53, and 95.

Regarding dependent claims 76, 85, 90-93, and 97, the claims incorporate substantially similar subject matter as claims 36, 45, 50-53, and 95. Thus, the claims are rejected along the same rationale as claims 36, 45, 50-53, and 95.

Art Unit: 2178

Regarding independent claim 94, Fay discloses a method in which predefined portions are stored in terminal nodes (Figure 1 and 2 of Fay). Fay also discloses a method in which one or more higher-level nodes are used to organize the terminal nodes, in which all nodes include the identity of a parent node, a position indicator, and an identifier (Figure 3 and 4 of Fay). Fay discloses a method for providing easy navigation in a document based on document portions. Fay does not disclose that the document in question is commentary. However, it would have been obvious to one of ordinary skill in the art at the time the invention was made to have used the method of Fay with a commentary document because it was well known at the time the invention was made that documents could contain commentary and the method of Fay is directed to all general document.

Also, Fay does not disclose a method in which a portion contains a scope that includes a start date, end date, and an update date that is in between the start and end date. However, McIntosh discloses a method in which a document has a scope, which includes a start date, an expiration (end) date, and an update date that exists in between the start and expiration date (column 31, line 21-column 46, line 45 of McIntosh). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the methods of Fay and McIntosh because it would have allowed easier management of time sensitive material.

12. Claims 39-40, 59-60, and 79-80 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fay (US Patent Number 5,892,513, issued on April 6, 1999) as

Art Unit: 2178

applied to claim 34 above, and further in view of Chau et al. (hereinafter Chau, US Patent Application Publication Number 2002/0133484, filing date of December 2, 1999).

Regarding dependent claims 39 and 40, Fay does not disclose a method in which the portions correspond to flat file records in a database or that XML data is divided into portions and placed into those flat files. However, Chau discloses a method in which XML is broken into portions and stored into flat file records in a database, and when that file is requested the portions are retrieved from the flat file database record (page 4, paragraph 0074-page 5, paragraph 0082 of Chau). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the method of Fay with the method of Chau because it would have provided a uniform way to store document portions.

Regarding dependent claims 59 and 60, the claims incorporate substantially similar subject matter as claims 39 and 40. Thus, the claims are rejected along the same rationale as claims 39 and 40.

Regarding dependent claims 79 and 80, the claims incorporate substantially similar subject matter as claims 39 and 40. Thus, the claims are rejected along the same rationale as claims 39 and 40.

13. Claims 41-42, 48-49, 61-62, 68-69, 81-82, and 88-89 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fay (US Patent Number 5,892,513, issued on April 6, 1999) as applied to claim 34 above, and further in view of Chau et al. (hereinafter Chau, US Patent Application Publication Number 2002/0133484, filing date

Art Unit: 2178

of December 2, 1999) as applied to claims 40 and 43 above, and further in view of Dolan et al. (hereinafter Dolan, US Patent Number 5,963,208, issued on October 5, 1999).

Regarding dependent claims 41 and 42, neither Fay nor Chau disclose a method in which a terminal node contains a label of the publication or that label data is associated with a higher-level node. However, Dolan discloses a method in which a terminal node includes a label of the publication that is associated with a higher-level node (Figure 7 of Dolan). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the methods of Fay and Dolan because it would have allowed the terminal nodes to exist more independently.

Regarding dependent claim 48 and 49, neither Fay nor Chau disclose a method in which a terminal node contains a label of the publication or that label data is associated with a higher-level node. However, Dolan discloses a method in which a terminal node includes a label of the publication that is associated with a higher-level node (Figure 7 of Dolan). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the methods of Fay and Dolan because it would have allowed the terminal nodes to exist more independently.

Regarding dependent claims 61-62 and 68-69, the claims incorporate substantially similar subject matter as claims 41-42 and 48-49. Thus, the claims are rejected along the same rationale as claims 41-42 and 48-49.

Art Unit: 2178

Regarding dependent claims 81-82 and 88-89, the claims incorporate substantially similar subject matter as claims 41-42 and 48-49. Thus, the claims are rejected along the same rationale as claims 41-42 and 48-49.

14. Claims 46, 66, and 86 are rejected under 35 U.S.C. 103(a) as being unpatentable over Fay (US Patent Number 5,892,513, issued on April 6, 1999) as applied to claim 34 above, and further in view of McIntosh (US Patent Number 6,185,576, filed on October 21, 1998) as applied to claim 36 above, and further in view of Weinberg et al. (hereinafter Weinberg, US Patent Number 6,144,962, filed on April 11, 1997).

Regarding dependent claims 46, neither Fay nor McIntosh disclose a method in which a higher-level nodes scope is dependent upon on or more descendant nodes and their scopes. However, Weinberg discloses a method in which the scope (validity based on need for update) of all nodes in an anchor sets can be dependent on the need to update a single terminal node (Figure 1 and column 23, line 39-column 24, line 25 of Weinberg). It would have been obvious to one of ordinary skill in the art at the time the invention was made to have combined the methods of Fay and Weinberg because it would have insured the user was viewing the most up-to-date document structure.

Regarding dependent claims 66, the claims incorporate substantially similar subject matter as claims 46. Thus, the claims are rejected along the same rationale as claims 45-46.

Art Unit: 2178

Regarding dependent claims 86, the claims incorporate substantially similar subject matter as claims 46. Thus, the claims are rejected along the same rationale as claims 46

Conclusion

15. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure:

US Patent Number 5,935,210

US Patent Number 6,189,019

US Patent Number 6,584,480

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Joshua D Campbell whose telephone number is (703)305-5764. The examiner can normally be reached on M-F (8:00 AM - 4:30 PM).

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather Herndon can be reached on (703)308-5186. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Art Unit: 2178

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

JDC
June 25, 2004



**STEPHEN S. HONG
PRIMARY EXAMINER**

Notice of References Cited	Application/Control No. 09/689,927	Applicant(s)/Patent Under Reexamination LESSING ET AL.	
	Examiner Joshua D Campbell	Art Unit 2178	Page 1 of 1

U.S. PATENT DOCUMENTS

*	Document Number Country Code-Number-Kind Code	Date MM-YYYY	Name	Classification
A	US-5,963,208 A	10-1999	Dolan et al.	345/760
B	US-5,892,513 A	04-1999	Fay, Robert G.	345/854
C	US-6,144,962 A	11-2000	Weinberg et al.	707/10
D	US-6,185,576 B1	02-2001	McIntosh, Lowrie	707/200
E	US-2002/0133484 A1	09-2002	Chau et al.	707/3
F	US-5,935,210 A	08-1999	Stark, Randall J.	709/224
G	US-6,189,019 B1	02-2001	Blumer et al.	715/513
H	US-6,584,480 B1	06-2003	Ferrel et al.	715/513
I	US-			
J	US-			
K	US-			
L	US-			
M	US-			

FOREIGN PATENT DOCUMENTS

*	Document Number Country Code-Number-Kind Code	Date MM-YYYY	Country	Name	Classification
N					
O					
P					
Q					
R					
S					
T					

NON-PATENT DOCUMENTS

*	Include as applicable: Author, Title Date, Publisher, Edition or Volume, Pertinent Pages)
U	
V	
W	
X	

*A copy of this reference is not being furnished with this Office action. (See MPEP § 707.05(a).)
Dates in MM-YYYY format are publication dates. Classifications may be US or foreign.

APPENDIX 11



IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Handwritten initials: AF / 41

In re patent application of:)	Before the Examiner
)	Joshua D. Campbell
Abha Lessing et al.)	
)	Group Art Unit 2178
Serial No. 09/689,927)	

Filed October 12, 2000)	
)	
MALTWEB MULTI-AXIS VIEWING)	
INTERFACE AND HIGHER LEVEL)	
SCOPING)	November 8, 2004

RECEIVED
NOV 18 2004
 Technology Center 2100

AMENDMENT AFTER OFFICE ACTION MAILED JULY 8, 2004

Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

In response to the Office Action mailed July 8, 2004, the period for response to which having been extended to November 8, 2004 by the accompanying Petition For Extension Of Time, please enter the following amendments and consider the following remarks.

I hereby certify this correspondence is being deposited with the United States Postal Service as first class mail in an envelope addressed to the Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on **November 8, 2004**.

Clifford W. Browning
 Name of Registered Representative

Clifford W. Browning
 Signature

November 8, 2004
 Date