

(19) **DANMARK**

(10) **DK/EP 1210675 T3**



Patent- og
Varemærkestyrelsen

(12) **Oversættelse af
europæisk patentskrift**

-
- (51) Int.Cl.: **G 06 F 17/30 (2006.01)**
- (45) Oversættelsen bekendtgjort den: **2016-01-11**
- (80) Dato for Den Europæiske Patentmyndigheds bekendtgørelse om meddelelse af patentet: **2015-09-30**
- (86) Europæisk ansøgning nr.: **00937569.2**
- (86) Europæisk indleveringsdag: **2000-05-17**
- (87) Den europæiske ansøgnings publiceringsdag: **2002-06-05**
- (86) International ansøgning nr.: **US2000013471**
- (87) Internationalt publikationsnr.: **WO2000072196**
- (30) Prioritet: **1999-05-20 US 315316**
- (84) Designerede stater: **AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE**
- (73) Patenthaver: **Evresearch Ltd., 1611 Tennyson Court, Columbus, OH 43235, USA**
- (72) Opfinder: **MAYNARD, George, 4639 Cleveland Road, Wooster, OH 44695, USA**
- (74) Fuldmægtig i Danmark: **Holme Patent A/S, Valbygårdsvej 33, 2500 Valby, Danmark**
- (54) Benævnelse: **System til informationshåndtering, -hentning og -fremvisning samt tilhørende fremgangsmåde**
- (56) Fremdragne publikationer:
US-A- 5 265 065
US-A- 5 642 502
US-A- 5 784 608
US-A- 5 864 863
US-A- 5 963 940
US-A- 5 999 925
LIDDY E.D.: 'DR-LINK system: phase I summary' PROCEEDINGS OR THE TIPSTER TEXT PROGRAM PHASE I WORKSHOP April 1994, pages 93 - 112, XP002929158

DESCRIPTION

BACKGROUND

[0001] The present invention is a computerized system and method for searching through and retrieving informations from an informational resource; and more particularly, the present invention is an information management, retrieval and display system for searching through an informational resource and for displaying the results of the search in an collapsible/expandable format based upon a user-selected display criteria or hierarchy.

[0002] An inherent drawback in many conventional search engines or search tools, such as Infoseek™, AltaVista™, Hotbot™, is that the results of the search are typically organized according to the number of hits that the search word or phrase made in each document (Web page) being searched. This type of search result display requires the end user to go through the hits one by one in order to finally access the document he/she was looking for. Another drawback with such conventional search engines is that the results of the search do not take into account that a word may have several different meanings, and may be used in many different contexts. For example, if an end user were looking for information on a cartoon mouse, because the search query would contain the word "mouse," the list of hits will include documents for the electronic cursor-control devices, documents providing biological information on mice, documents providing pet information on mice, etc. Therefore, the end user may have to go through an enormous number of these hits before finally (if ever) reaching a hit related to the cartoon mouse.

[0003] Thus, there is a need for a search engine or search tool that allows the end user to that arranges the search results in a manner that allows the end user to effectively and quickly obtain items of interest.

[0004] U.S. Patent No. 5,784,608 to Hooper et al. is generally directed to a method and apparatus for automatically creating hypertext documents from information using profiles and topics. The method and apparatus disclosed by US5784608 can only process informational resources (i.e. a list of search results) that are structured with mark-up such as hypertext mark-up language (HTML). There remains a need for a tool that can process both structured and/or unstructured informational resources without needing to add or modify mark-up in the informational resources.

SUMMARY

[0005] Embodiments of the present invention are recited by the appended claims.

[0006] The present invention is an information management, retrieval and display system for searching through an informational resource, such as a document (e.g., a treaty), a number of individual documents (e.g., Web pages resident on the Internet), or a stream of information (e.g., DNA code, source code, satellite data transmissions, etc.) and for displaying the results of the search in an collapsible/expandable format based upon a user-selected display criteria or hierarchy. Such a display hierarchy will allow the end-user to effectively and quickly obtain items of interest from the search results. The type or format of the informational resource is not critical.

[0007] The invention includes four primary modules, a break module, an indexing module, a search module and an un-break module. The break module is an expert system operating upon a set of expert rules that define its operation. The break module parses through the informational resource to break up the informational resource into finite elements (such as paragraphs, sections, sub-sections, segments etc.). The break module also creates categorical tags for each of these finite elements, where the categorical tags assigned to each of the finite elements are based upon an analysis (defined by the set of expert system rules) of the contents of each of the finite elements. The categorical tag can include a standard classification such as, for example, "Dewey Decimal-type" number. The categorical tag can also include an organizational attribute (such as pertaining to the type or location of the finite element with respect to the rest of the rest of the informational resource), a date-stamp, a categorical word, etc. Preferably, the categorical tags are inserted into the finite element.

[0008] The index module parses through the finite elements identified/created/processed by the break module and creates a searchable database having a database record for each of the finite elements identified by the break module. The searchable database is a type of reverse index, where each record includes an address or location of the corresponding finite element (and, in turn, the categorical tag included therewith), and strings (such as words, phrases, etc.) contained in the finite element and their frequency (i.e., their weight) within the finite element.

[0009] In applications where the users of the invention do not have control of the information being searched (i.e., Web pages on

the Internet), each database record may also include the categorical tag, since the categorical tag will not be able to be inserted by the break module into the finite elements themselves. Furthermore, with the Web search application, it may not be necessary to utilize the break and un-break modules at all since each Web-page or link might be considered a finite-element for the purposes of the present invention.

[0010] Once the reverse index is created, a search of the reverse index may be performed. Key strings (such as key words, phrases or symbol segments) may be supplied by an end user as a search query, and a display hierarchy or criteria may also be selected or defined by the user. The selected display criteria will instruct the search module how to manipulate the data of the search results. Specifically, the selected display criteria will define if the search results are to be displayed in an order or structure based entirely upon the information contained within the categorical tags (research-centric), if the search results are to be displayed in an order depending entirely upon the frequency of the key strings present in the finite elements (conventional), or if the search results are to be displayed in an order or structure based upon a combination of the two (document-centric).

[0011] The search module accesses the search query and searches through the reverse index for database records matching the specific search term or query. The search results are then displayed in an collapsible/expandable (tree) structure by applying the information in the categorical tags for each of the finite elements satisfying the search criteria to the selected display hierarchy. For example, if the selected hierarchy is a document-centric hierarchy, a first level of the display hierarchy may be, for example, the year in which the finite element was created; a second level of the display hierarchy may be, for example, the order in which the finite elements appear in the document; and a third level of the display hierarchy may be, for example, based upon the frequency in which the search words appear in each of the finite elements. The operation of the search module, as with the break and index modules, is based upon a set of expert rules. Therefore, if the search results are not satisfactory, the expert rules in the break, index and/or search modules are modified and the procedure is performed again.

[0012] Once one of the finite elements in the search result display are selected by the end-user, the un-break module allows the end user to view a contiguous portion of the informational resource that the selected finite element belongs to. The un-break module will assemble selected finite element with other related finite element to reconstruct the contiguous portion of the informational resource. The un-break module refers to the categorical tag of the selected finite element for information related to the location of the finite element with respect to the entire informational resource, and will then build a portion of the informational resource from all of the finite elements belong to that portion. For example, if the selected finite element is a paragraph of a document, the un-break module may be configured to rebuild the chapter of the document to which the paragraph belongs. As with the other modules of the present invention, the operation of the un-break module is controlled by a set of expert rules that may be modified if the results are unsatisfactory.

[0013] It is envisioned that the rule sets will be created and refined by an expert on the document or information being searched. For example, if the system of the present invention is to be associated with Volume 37 of the Code of Federal Regulations, an individual (or group of individuals) with intimate knowledge of the Volume would be best suited to generate and fine-tune the rule sets. The fine-tuning of the rule sets would involve the individual continuously performing example searches on the Volume using the rule sets, and continuously modifying the rule sets to obtain the search results with the desired content and format. Once the rule sets have been fine-tuned, the search module of the present invention can be packaged along with the Volume and sold or distributed as a searchable Volume. Likewise, the search module could operate on a Web-site so that users can access the Web-site and perform searches on the Volume. Since the rule sets have already been defined and fine-tuned by the "experts," the users would have a fully operable search engine that performs searches and displays results in accordance with an expert's intimate knowledge with the Volume.

[0014] As mentioned above, it is also envisioned that an example of the invention is designed to search through a number of individual Web pages resident on the Internet and to display the results of the search in an collapsible/expandable format based upon a user-selected display criteria or hierarchy. In such an example a break module in the form described above may not be necessary because each Web page may already be considered to be a "finite element" and the search engine will not be able to modify the Web pages. Accordingly, in such an application, the index module will parse through each of the Web pages (finite elements) to create a searchable database having a record for each of the Web pages. Each record in the searchable database will include the Web address of the Web page, non-common words contained in the Web page along with their frequency (weight), and a categorical tag, as described above, which includes data based upon an analysis of the contents of the Web page. The index module will also review each of the Web pages to determine if the creator of the Web page had embedded a categorical tag into the Web page itself; and if such an embedded categorical tag is found, the index module may simply insert this pre-defined categorical tag into the database record rather than creating one itself. Therefore, as the present invention becomes more prevalently used on the Internet, Web page creators may desire to create their own categorical tags for their Web pages rather than having the search engine create one for them. With this feature, the Web page designer may be able to influence the search results, perhaps to achieve a more accurate depiction of the Web site. Of course, such a feature may also be used by the Web designers in a

deceptive manner, where the categorical tag will cause the Web page to be listed in search results when the searcher is looking for an entirely different type of information. Recognizing this potential problem, the index module will include an option where it will compare the actual contents of the Web page against the embedded categorical tags, and will create a new categorical tag to be inserted into the database record if there is a significant difference. Likewise, the index module can be configured to filter out Web sites having undesirable or unsavory content as indicated by the embedded categorical tags or as determined by a review of the content of the Web page itself.

[0015] In another example of the invention, the dynamic expert rule sets may be configured to accept and index all manner of static and dynamic information (such as news-feeds, data transmissions, etc.) on a global scale where an end-user will be able to efficiently and quickly obtain any sort of information he/she wishes from a hierarchal search result display based upon a categorical organization scheme such as the Dewey-Decimal system.

[0016] Thus, in one aspect of the present invention, a method for retrieving information from an informational resource comprises the steps of: (a) dividing the informational resource into a plurality of finite elements; (b) assigning a categorical tag to each of the plurality of finite elements, where the categorical tag includes data pertaining to a content of the finite element; (c) generating a searchable database record for each of the plurality of finite elements, where each searchable database record includes at least one string contained within the finite element, where the string can be a word, a phrase, a symbol, a group of symbols, a data segment or the like; (d) supplying a search string; (e) searching the searchable database for searchable database records containing the search string; (f) arranging the results of the searching step in a hierarchal structure according, at least in part, to the data in the categorical tags assigned to the finite elements found in the searching step; and (g) displaying the results of the searching step in the hierarchal structure.

[0017] The informational resource may be a single document, a plurality of documents or a stream of data, and the step of identifying the finite elements may include the steps of identifying sections or sub-sections within the document(s) or data stream or by simply identifying the documents themselves. The step of dividing the informational resource into a plurality of finite elements is preferably performed by an expert system according to a rule set; and the step of assigning a categorical tag to each of the plurality of finite elements is also preferably performed by an expert system according to another rule set. If unsatisfactory results are obtained in step (g) above, one or both of the rule sets may be modified by the end user and the steps (a) through (g) may be performed again.

[0018] Each database record preferably includes an address or pointer to the corresponding finite element and further preferably includes all of the non-common strings (e.g., words or phrases) contained within the corresponding finite element along with the frequency that such strings appear.

[0019] In another aspect of the present invention, a method for retrieving information from an informational resource includes the steps of: defining a first rule set for dividing the informational resource into a plurality of finite elements; utilizing the first rule set, dividing the informational resource into a plurality of finite elements; defining a second rule set for creating a categorical tag for one of the plurality of finite elements; utilizing the second rule set to create a categorical tag for each of the plurality of finite elements; generating a searchable database including a searchable database record for each of the finite elements; searching the searchable database for relevant database records; associating the relevant database records found in the search with corresponding relevant finite elements; selecting a hierarchy for displaying identifying phrases pertaining to the relevant finite elements; ordering the relevant finite elements in the hierarchy according, at least in part, to the categorical tag for each of the finite elements; and displaying the identifying phrases pertaining to the relevant finite elements according to the results of the ordering step.

[0020] In another aspect of the present invention, a data storage device (such as a CD ROM) is provided, which comprises: an informational resource divided into a plurality of finite elements, where each of the finite elements includes a categorical tag and a database record assigned thereto, where the categorical tag includes data pertaining to a content of the finite element and the database record includes at least one string contained within the finite element; and also comprises software instructions programmed to retrieve and display at least a portion of the informational resource. The software instructions are configured to perform the steps of: supplying a search string, searching through the database records for relevant database records containing the search string, arranging the results of the searching step in a hierarchal structure according to the information in the categorical tags assigned to the finite elements corresponding to the relevant database records, and displaying identifying phrases for the finite elements corresponding to the relevant database records in the hierarchal structure.

BRIEF DESCRIPTION OF THE DRAWINGS

[0021]

Fig. 1 is a schematic flow-diagram representation of the operation of a first example of the present invention;

Figs. 2A and 2B are flow-chart representations of the operation of the example illustrated in Fig. 1;

Fig. 3 is a flow-chart representation of an operation of a second example of the invention, resident on a data storage device such as a CD ROM;

Fig. 4 is a schematic flow-diagram representation of the operation of a third example of the present invention; and

Figs. 5A and 5B are flow-chart representations of the operation of the example illustrated in Fig. 4.

DETAILED DESCRIPTION

[0022] The present invention is an information management, retrieval and display system for searching through an informational resource, such as a document (e.g., a treaty), a number of individual documents (e.g., Web pages resident on the Internet), or a stream of information (e.g., DNA code, source code, satellite data transmissions, etc.) and for displaying the results of the search in an collapsible/expandable format based upon a user-selected display criteria or hierarchy. Such a display hierarchy will allow the end-user to effectively and quickly obtain items of interest from the search results.

[0023] As shown in Fig. 1, in a first example of the invention, the information management, retrieval and display system includes four primary modules, a break module 10, an indexing module 12, a search module 14 and an un-break module 16. Each of these processing modules are preferably expert engines operating upon a set of expert rules that define the operation of the individual module. As will be described in further detail below, the expert rules for these modules are preferably generated by a person or persons having intimate knowledge of the document or documents 18 being searched; and the fine tuning of the expert rules is a iterative process where the expert will modify or change the rules of one or more of the above modules if a search through the document or documents proves to be unsatisfactory.

[0024] The break module 10 parses through an informational resource, such as a group of documents 18 to break up the group of documents into "finite elements" 20a-20z. Each finite element is a user-defined "basket" of information from documents that is to be individually indexed and searched. The finite element is usually not a single word, phrase or symbol, but is a section or portion of an informational resource that can be identified and isolated by the break module. A simple example of a finite element would be the individual paragraphs of a document. Other examples of finite elements would include sub-chapters of a document, individual pages of a document, and other types of identifiable sections of a document. In some instances, the finite element can be the entire document itself. The break module is also responsible for analyzing the contents of each finite element 20a-20z and creating a categorical tag 22a-22z for each finite element, which is to be inserted into the finite element. The categorical tags 22a-22z may include a standard classification based upon the content analysis such as, for example, a "Dewey Decimal" type number, or some other categorical reference number. The categorical tag may also include an organizational attribute such as pertaining to the type of finite element or the location of the finite element within the document, a date stamp, a categorical word or phrase summarizing the contents of the finite element, etc. As will be discussed in detail below, the contents of each categorical tag provides information to the search module 12 so as to assist the search module in creating the hierarchical display of the search results.

[0025] The index module parses through each of the finite elements created by the break module and creates a searchable database 23 including a database record 24a-24z for each of the finite elements created by the break module. The searchable database 23 is a type of reverse index, where each record 24a-24z includes an address or location of the corresponding finite element and all words contained within the finite element (preferably excluding common words such as "and," "in," "the," ...) along with their frequency of appearance within the finite element (i.e., their weight).

[0026] At some point during the process, a user, which may be an end user or may be the expert developing the rule sets, will enter a search query 26 and an optional hierarchical selection 28. The search query may be any conventional search query as available to those of ordinary skill in the art and may include search words or phrases and/or operators tying the words together. The hierarchy selection informs the search module the type of display format that the user wishes to see the results displayed within. Specifically, the hierarchy selection will inform the search module whether or not the search results are to be displayed in an order or structure based entirely upon the information contained within the categorical tags (research-centric), if the search results are to be displayed in an order depending entirely upon the frequency of the key words or phrases present within the finite elements (conventional), or if the search results are to be displayed in an order or structure based upon a combination of the two

(document-centric).

[0027] The search module will utilize the search query to search through the database records 24a-24z so as to find the database records 30 matching the words or phrases in the search query. The search module will then, depending upon the selected hierarchy 28, display the search results 32 in an order or collapsible/expandable tree structure based upon information from the categorical tags 22 included in the finite elements 20 that are associated with the records 30 matching the search query. For example, a first level of the display hierarchy might be ordered according to the chapters of a document that the finite elements are contained within. Information regarding the chapters that the finite elements are contained within will be resident within the categorical tags associated with the finite elements. A second level of the display results may order the finite elements for each chapter based upon the weight or frequency that the search words or phrases appear within each finite element. Therefore, on the search results screen the end user will select which chapter he or she would like to view a relevant finite element from and the display will then expand to show the finite elements from that chapter matching the search query. These finite elements contained within this chapter will be ordered depending upon the weight of the search query or words. From there, the user will make a selection 34 indicating to the un-break module 16 which of the finite elements the user wishes to view.

[0028] It will be appreciated by those of ordinary skill in the art that the different combinations of ordering schemes and tiers for any given hierarchy is virtually limitless. Other examples of ordering schemes and tiers can be based upon the topic of the finite element, the author or provider of the finite element, the time/date of the finite element, the position of the finite element with respect to the information resource, etc. It is also within the scope of the invention that the hierarchy only include one tier or level of ordering.

[0029] While it is preferred that the search module displays the search results in an collapsible/expandable tree structure, it is also within the scope of the invention that the display results be displayed in alternate hierarchal or tiered structures. An example of an alternate hierarchal/tiered structure is the use of a cascaded or tiled display to present the various levels of the hierarchy. Of course, if there is only one tier or level of ordering, the display structure would not need to be collapsible/expandable.

[0030] The search module may also be configured to recognize that a string in the search query may have other permeations, which may be used by the search engine to provide matches with the database records. For example, if the search query includes a word in a first language, it is within the scope of the invention for the search module to provide the word in other languages when looking for matches with the database records. Likewise, it is within the scope of the invention for the search module to provide other known forms or tenses of the word; and it is also within the scope of the invention for the search module to provide other search words having a similar or the same meaning.

[0031] The un-break module 16 accesses the categorical tag of the selected finite element 34 to determine the other finite elements 36 of the documents 18 that are to be grouped together so as to form the single contiguous display 38. For example, if the selected finite element 34 is a paragraph of the document, the un-break module 16 will refer to the categorical tags of the remaining finite elements to determine the other finite elements 36 that appear on the same page as the selected finite element so as to display the entire page 38 rather than the single paragraph. Likewise, the un-break module can group related finite elements together in a contiguous chapter, section, or other contiguous identifiable portion of the document or documents. Simply put, the un-break module is used for displaying the selected finite element in context with the remaining portions of the informational resource.

[0032] While it is preferred that the un-break module is utilized to reconstruct contiguous portions of the informational resource, it is within the scope of the invention to configure the expert rule sets of the un-break module to construct new informational resources using the selected finite elements and other finite elements from the original informational resource. For example, the un-break module may be configured to compile all of the finite elements matching the search query into a new informational resource, using the categorical tags for these finite elements to dictate the order in which the finite elements will be compiled. In another example, the un-break module may be configured to review the categorical tag of the selected finite element to determine other finite elements that are related to the selected finite element based on the date that the finite elements were created, or the author/owner of the finite element, or the content of the finite element; and the un-break module will then construct a new informational resource compiling all of the related finite elements.

[0033] Figs. 2A and 2B provide a flow chart representation of an operation of the information management, retrieval and display system for the example described above. As shown in functional block 40, a first step is to access the informational resource being examined. As illustrated in functional block 42, the next step is to select the appropriate expert rule sets to apply for searching through the informational resource. The particular rule set selected will depend upon the type of information resource that was accessed in step 40. For example, a set of expert rule sets used for searching through and analyzing the Antarctic Treaty will be different than a set of rule sets used for analyzing and searching through volume 37 of the Code of Federal Regulations. As shown

in functional block 44, the next step is to break the information resource into a plurality of finite elements according to a first set of the expert systems rules. As discussed above, this step involves breaking the informational resource into identifiable segments of information such as paragraphs, subsections, pages, chapters, subchapters and the like. An example rule set for breaking the Antarctic Treaty into a plurality of finite elements is provided below in Table 1.

TABLE 1: FINISHED EXAMPLE OF A 'RULE SET' FOR AUTOMATICALLY DIVIDING DOCUMENTS INTO SEGMENTS OR ELEMENTS¹		
DOCUMENT DIVISION LEVELS	SPECIFIC DOCUMENT DIVISIONS	PATTERN MATCHING RULES
Primary Level	Antarctic Treaty, Conventions, Protocol and its Annexes	Recognize by bolded large fonts centered on page
Secondary Level	Recommendations, Measures, etc.	Recognize by Roman numerals
Tertiary Level	Articles within documents from the primary or secondary levels	Recognize by medium fonts centered on page with a colon
Grouped Level	Antarctic Treaty Consultative Meeting	Group documents by their Roman numerals
Appended Level	Year	Append the signature date for documents at the primary, secondary or grouped levels

¹ Based on the public-domain documents in the Antarctic Treaty Handbook which has been published since the 1960's by the United States Department of State in hardcopy form only and which now has been converted into a searchable database.

² Source codes are described using JAVA but could easily be written in PERL or any other programming language. See Appendix A for example source code segments

[0034] As shown in the above table, the example rule set is adapted to divide the Antarctic Treaty into a plurality of levels where a primary level of the Treaty, which involves the Antarctic Treaty, Conventions, Protocol and its Annexes, is recognized by the search engine by identifying bold, large font centered on a page. A secondary level, which are the Recommendations and Measures contained within the Treaty, are recognized by the search engine by identifying Roman numerals. A tertiary level is utilized to divide up the primary and secondary levels into smaller finite elements. This tertiary level of finite elements is recognized by the search engine by identifying medium fonts centered on a page with a colon. The remaining levels of the table should be apparent to those of ordinary skill in the art upon analyzing the table and the associated pattern matching rules.

[0035] Accordingly, the purpose of the above rule set is to create an automatic tool for matching patterns that distinguish hierarchies, segments or elements within any type of informational resource. The rule set is developed in relation to user-defined requirements for the segments or elements that need to be indexed and searched within the informational resource. It will also be apparent to those of ordinary skill in the art that the rule sets will be greatly simplified in informational resources that include already distinguished segments or elements, such as in separate columns or blocks. Preferably, the rule sets are designed by an expert having intimate knowledge of the informational resource, in an iterative manner utilizing feed-back loops as will be described below.

[0036] As shown in functional block 46, a next step is to create a categorical tag for each of the finite elements based upon a positional and/or content analysis of the finite element according to a second set of expert system rules. An example of a rule set for defining categorical tags for finite elements extracted from the Antarctic Treaty is provided below in Table 2.

TABLE 2: EXAMPLE OF CATEGORICAL TAGS THAT WERE AUTOMATICALLY ATTACHED TO FINITE ELEMENTS CREATED WITH THE USER-DEFINED 'RULE SETS'(SEE TABLE 1)¹	
DOCUMENT DIVISION LEVELS	SPECIFIC DOCUMENT DIVISIONS
Primary Level	Antarctic Treaty, Conventions, Protocol and its Annexes
Secondary Level	Recommendations, Measures, etc.
Tertiary Level	Articles within documents from the primary or secondary levels
Grouped Level	Antarctic Treaty Consultative Meeting
Appended Level	Year

¹ Based on the public-domain documents in the Antarctic Treaty Handbook which has been published since the 1960's by the United States Department of State in hardcopy form only and which now has been converted into a searchable database.

TABLE 2: EXAMPLE OF CATEGORICAL TAGS THAT WERE AUTOMATICALLY ATTACHED TO FINITE ELEMENTS CREATED WITH THE USER-DEFINED 'RULE SETS'(SEE TABLE 1)¹

DOCUMENT DIVISION LEVELS

SPECIFIC DOCUMENT DIVISIONS

² Source codes are described using JAVA but could easily be written in PERL or any other programming language. See Appendix A for example source code segments.

[0037] As shown in Table 2, the categorical tag will include notation indicating the finite element's position within each of the various identified levels of the Antarctic Treaty. For example, the categorical tag will include information indicating if on a primary level, the finite element is contained within the Antarctic Treaty, the Conventions, the Protocol or its Annexes. On a secondary level, the categorical tag will indicate whether or not the finite element is included in the Recommendations, Measures, etc. As shown in the bottom of the table, the categorical tag for each of the finite elements will also include a content base notation indicating the year that the particular section or finite element was created. Of course, the type and variations of positional and/or content base notations included in the categorical tags are virtually limitless. For example, the rule set may be configured to analyze the contents of the finite element so as to provide a categorical word or phrase which provides a clue to the user as to the contents of the finite element. Similarly, rather than utilizing a word or phrase, the rule set can analyze the contents or position of the finite element to provide a categorical reference number to the finite element, such as a Dewey Decimal type number.

[0038] As shown in functional step 48, a next step is to insert the categorical tag created above in step 46 into the finite element created in step 44. As shown in functional block 50, a next step is to generate, for each of the finite elements, a searchable database record. Each database record preferably contains the noncommon strings (e.g., words, phrases, symbols) contained within the finite element along with their frequency (i.e., weight). Furthermore, each database record will include an address, location or link to the corresponding finite element. As shown in functional block 52, a next step is to enter a search string such as a word, phrase or symbol(s) and to select a display hierarchy. As shown in functional block 54, a next step is to search through the database records created in functional block 50 for matches between the search string and the noncommon strings of the database records. This searching step will identify the relevant database records having noncommon strings matching the search string. As shown in functional block 56, the relevant database records found in the searching step 54 will be ordered by applying information from each of the categorical tags of the relevant database record's associated finite element to the selected display hierarchy and/or by applying the weight of the matching search strings in the relevant database records to the selected display hierarchy.

[0039] For example, a first level of the display hierarchy for the Antarctic Treaty might be the year that the finite element was created; the second level might be ordered according to the order of the Articles of the Antarctic Treaty; and a third level of the display hierarchy might be ordered according to the weight of the matching strings contained within the database records.

[0040] As shown in functional block 58, a next step would be to display the search results in the collapsible/expandable hierarchy on a display screen. As shown in functional block 60, the user will determine whether the search results were satisfactory, and if not the process will advance to functional block 62 where the user will modify one or more of the rule sets and will return either to functional block 44 or to functional block 52 depending upon which rule sets have been modified.

[0041] If, in functional block 60, the search results are satisfactory, the process will advance to functional block 64 where the user will select one of the finite elements from the search results display. Then in functional block 66, the categorical tag of the selected finite element will be used to identify other finite elements that are to be grouped together with the selected finite element to create a contiguous portion of the informational research to be displayed. Finally, in functional block 68, the contiguous portion of the informational resource will be displayed on the display screen or printed.

[0042] It is envisioned that an expert having intimate knowledge of the informational resource will develop the rule sets based upon his or her knowledge of the informational resource. Thereafter, once the rule sets have been fully developed, the feed-back portion of the above-described flow chart will no longer be necessary.

[0043] Furthermore, once the rule sets have been fully developed, the search module, the un-break module and the fully developed rule sets may be incorporated onto a data storage device (such as a CD ROM, a disk-drive, a floppy-disk and the like) along with an informational resource pre-broken into its plurality of finite elements, where each of the finite elements includes the corresponding categorical tag previously created therefore, along with the pre-created searchable database for the plurality of finite elements. Therefore, such a storage device would essentially provide a searchable document that includes the entire content of the informational resource along with a search engine that has been finely tuned by an expert with intimate knowledge of the informational resource, so that end users of the CD ROM (or other type of storage device) would be able to take advantage of the expert's knowledge and experience in searching through the informational resource contained therewith.

[0044] As shown on Fig. 3, a flow chart representation of an example of the invention resident on a data storage device, such as a CD ROM, is presented. Essentially, this example is equivalent to the example described in Figs. 2A and 2B above, except that the development of the rule sets are not longer required. As shown in functional block 52', a first step would be for the end user to enter a search string and select a display hierarchy. In functional block 54', the next step would be for the search module to search through the database records contained on or downloaded from the CD ROM to match the search string with the non-common strings contained in the searchable database records. As shown in functional block 56' the next step would be for the search module to order the search results by applying information in the categorical tags of the matching finite elements (which are contained in, or are downloaded from the CD ROM) and/or by applying the weight of the matching strings to the selected display hierarchy as discussed above. As shown in functional block 58' the next step is to display the search results in preferably a collapsible/expandable hierarchy. As shown in functional block 60', the end user, upon viewing the search results will determine whether or not the results are satisfactory. If not satisfactory, the process will return to functional block 52' where the end user will input a new search string and/or will select a new display hierarchy. If the display results of step 58' are satisfactory, the process will advance to functional block 64' where the end user will select one of the finite elements from the search results display. Advancing to functional block 66' the un-break module will reconstruct the portion of the information resource that includes the selected finite element by accessing the selected finite element and the other surrounding or related finite elements from the CD ROM to create the contiguous portion of the informational resource that included the finite element.

[0045] In another example of the present invention the information management, retrieval and display system may be specifically configured to search through a number of individual Web pages resident on the Internet and to display the results of the search in an collapsible/expandable format based upon a user selected display criteria or hierarchy. In such an example a break module in the form described above may not be necessary because each Web page may already be considered a "finite element" and the search engines will not be able to modify the Web pages. Therefore, with such an example the search engine will also not be able to insert the categorical into the finite elements. Therefore, in this example the categorical tags may be either stored separately from the finite elements or incorporated directly into the database records. Furthermore, it is envisioned that the Web page creators may desire to create their own categorical tags for their Web pages rather than having the search engine create one for them. With this feature, the Web page designer may be able to influence the search results, perhaps to achieve a more accurate description of his or her Web site. Of course, in such a feature may also be used by the Web designers in a deceptive manner, where the categorical tag will cause the Web page to be listed in search results when the searcher is looking for an entirely different type of information. Recognizing this potential problem, the index module may include an option where it will compare the actual contents of the Web page against the embedded categorical tags inserted by the Web page designer, and may create a new categorical tag to be inserted in the database record for the Web page if there is a significant difference between the two. Likewise, the search engine can be configured to include an optional filter that will filter out Web sites having unsavory contents as indicated by the embedded categorical tags or as determined upon a review of the content of the Web page itself.

[0046] As shown in Fig. 4, in such an example of the invention, the information management, retrieval and display system includes two primary modules, an index module 70 and a search module 72. Each of these processing modules are preferably expert engines operating upon a set of expert rules that define the operation of the individual module. The index module 70 will periodically crawl through the volume of Web pages 74 utilizing a conventional Web crawling or Web searching technology such as a spider technology, which is adapted to examine each Web page (or as many as possible) provided on the Internet. As shown in Fig. 4, several of the Web pages may include a predefined, embedded categorical tag 76 included therewith. As discussed above, such an embedded tag 76 would be inserted in the Web page by the Web page designer so that the search engine of Fig. 4 would utilize this predefined embedded categorical tag rather than creating one on its own. An example of a rule from the expert rule set for defining the categorical tag in this example would be to identify the most prominent word or phrase on the initial screen appearing when the Web site is accessed.

[0047] The index module 70 will also create a searchable database 78 including a database record 80a-80z for each of the Web pages accessed above. This searchable database 78 is a type of reverse index or each record 80a-80z includes a link to a corresponding Web page, all words contained within the Web page (preferably excluding common words) along with their frequency of appearance within the Web page, and a categorical tag created by the index module or a copy of the categorical tag that was included in the particular Web page as described above. It is envisioned that the index module would constantly be re-accessing the Web pages 74 and updating the searchable database 78, since the contents of Web pages are also constantly being updated or changed.

[0048] When a user wishes to conduct a search using the search engine, the user will enter a search query 82 and select an optional hierarchical selection 84. The search query may be any conventional search query as available to those of ordinary skill in the art, it may include a search word or phrases and/or operators tying the words together. The hierarchy selection informs the search module the type of display format that the user wishes to see the results displayed within. Specifically, the hierarchy

selection will inform the search module whether or not the search results are to be displayed in an order or structure based entirely upon the information contained within the categorical tags (research-centric), if the search results are to be displayed in an order depending entirely on the frequency of the key words or phrases present within the finite elements (conventional), or if the search results are to be displayed in an order or structure based upon a combination of the two (document-centric).

[0049] The search module 72 utilizing a search query 82 to search through the database records 80a-80z so as to find the database records 86 matching the words or phrases in the search query. The search module will then, depending upon the selected hierarchy 84, display the search results 88 in an order or in a collapsible/expandable tree structure based upon information from the categorical tags 89 included within the database records 87 matching the search query. From the display 88, the user will make a selection 90 of a link to a Web page that he or she wishes to view and the search module will then display the Web page 92 on the display screen.

[0050] Figs. 5A and 5B provides a flow chart representation of an operation of the example described above in Fig. 4. As illustrated in the function block 94, a first step is to access a Web page on the Internet. In functional block 96, the next step is to determine whether the access Web pages includes an embedded categorical tag. If the Web page includes an embedded categorical tag the process would advance to functional block 98 where the process will determine whether the embedded categorical tag is consistent with the content of the Web page. If the Web page does not include an embedded categorical tag or if the categorical tag is not consistent with the content of the Web page, the process will advance to functional block 100 where a categorical tag will be created for the Web page. If the embedded categorical tag is consistent with the content of the Web page in step 98 or if the categorical tag is created for the Web page in step 100, the process will advance to functional block 102 where a searchable database record will be generated for the Web page. This searchable database record will include the non-common words or phrases contained within the Web page and their frequency (i.e., weight) a link to the Web page and the categorical tag embedded within the Web page or created in step 100 above. The process will then advance to functional block 104 to determine whether a next Web page is to be accessed. If so, the process will return to functional block 94. If the searchable database is complete, the process will advance to functional block 106 where a user will enter a search word or phase in selected display hierarchy.

[0051] Advancing the functional block 108, the search engine will search through database records for matches between the search word or phrase and the non-common word or phrases contained within the database records. Advancing to functional block 110 the search engine will then order the results of the search by applying the information in the categorical tags matching database records to the selected display hierarchy and/or by applying the weight of the search word or phrase in each of the matching database records to the selected display hierarchy. Advancing to functional block 112, the next step would involve displaying the search results on the display screen. In functional block 114, if the search results are satisfactory, the user will select a Web page link on the display screen and the search engine will display the associated Web page selected. If the search results are unsatisfactory, the process will advance to functional block 118 where the user will enter a new search word or phrase and/or select a new display hierarchy and the process will return to functional block 108 so that another search can be performed.

[0052] In the present example the expert rule sets for creating the categorical tags, and the database records may be defined by an expert utilizing an iterative variation of the above process on a limited portion of the Internet (similar to that as described in Figs 2A and 2B above). Once the rule sets have been refined, the rule sets can be applied to the entire Internet. The above described search engine can be operating on a Web site or may be contained in a memory device such as a CD ROM which can be downloaded onto a computer having access to the Internet.

[0053] The following Appendix provides example code segments for dividing the Antarctic Treaty into a plurality of finite elements and for creating categorical tags for the finite elements.

APPENDIX A

[0054]

Table 1:

1) \$line =~ m/((\d)\.(\d\d\d)?\.\d\d\d)?\.\d\d\d?(\d\d\d)?([a-zA-z\-_]+)/
2) \$line =~ m/<P ALIGN="CENTER">([[XVLDCM]+)-([^<]+?)<VP>/
3) \$line =~ m/<P ALIGN="CENTER">(Extract(s)?\s+from\s+Report\s+of\s+([[XVLDCM]+)(th st)\s+ATCM[^\^*]*)/i
4) Simple variable substitutionâââ A Hashtable is generated using the code :

```

#####
# SUB: gen_hashes
#
# Purpose: To generate a tables based on a set pattern from an
# files related to the Antarctic Treaty Project.
#####
sub gen_hashes
{
    my $FileList=@_ [0];
    print "FileList: $FileList<br>\n";
    print "Generating Year-Roman Numeral and Year-Major Document Name Hashes.\n\n";
    print "Processing";
    while ($InputFN = glob("$FileList"))
    {
        open (InFile, "<$InputFN") // die("Cannot open $InputFN\n!");
        print ".";
        while (defined($line = <InFile>))
        {
            chomp($line);
            # Remove unwanted characters from the line
            $line = rem_all($line);
            # print $line;
            # Begin Matching
            # Attempt to grab the last for digits of a line, if they are on the line
            if ($line =~ m/^(w+ w+ w+w+|w+ w+ w+), ?(w+|w+ |w+ |w+ |w+).??(d\d\d)(
w+)?-(d\d\d)(w+)? w+(d\d\d\d)/)
            {
                # Debugging Code
                #print "$1 $2 $3 $4 $5 $6 $7 S8 $9 $10\n";
                $MajorDoc = $1;

                $year = $7;
                if (" $year" ne "")
                {
                    {
                        if (defined($NextLine = <InFile>))
                        {
                            {
                                # Remove unwanted characters from the line
                                $NextLine = rem_all($NextLine);
                                # Match the Roman Numerals at the beginning of the
                                # next line
                                if ($NextLine =~ m/^([IVXLCDM]+)/)
                                {
                                    $RomanNumeral = $1;
                                    SYearTrans{$RomanNumeral} = $year;
                                }
                            }
                        }
                    }
                }
            }
        }
    }
}

```

```

}
}
}
}
}
}
close InFile;
}
print "\n";
}

```

[0055] Then, when the Dewey tag is written out, the ATCM is placed in the appropriate position, whether it be the first, second, or third level using the Year from the currently matched pattern to obtain the correct ATCM from the hashtable (DocTrans table specifically) as generated above.

5) Simple variable substitution à à à A Hashtable is generated using the code :

```

##### # SUB: gen_hashes # #
Purpose: To generate a tables based on a set pattern from an # files related to the Antarctic Treaty Project.
##### sub gen_hashes { my $FileList
= @_ [0]; print "FileList: $FileList<br>\n"; print "Generating Year-Roman Numeral and Year-Major Document Name Hashes.\n\n";
print "Processing", while ($InputFN = glob("$FileList")) { open (InFile, "<$InputFN") // die("Cannot open $InputFNn$!"); print ".";
while (defined($line = <InFile>)) { chomp($line); # Remove unwanted characters from the line $line = rem_all($line); # print $line; #
Begin Matching # Attempt to grab the last for digits of a line, if they are on the line if ($line =~ m/^\w+ \w+ \w+ \w+ \w+ \w+ \w+ ?
(\w+ \w+ \w+ \w+ \w+ \w+)? ?(\d\d\d\d)( \w+)?-(\d\d\d\d)(\w+)? \w+ (\d\d\d\d\d) { # Debugging Code #print "$1 $2 $3 $4 $5 $6 $7 $8 $9
$10\n"; $MajorDoc = $1; $year = $7; if (" $year" ne "") { if (defined($NextLine = <InFile>)) # Remove unwanted characters from the
line SNextLine = rem_all($NextLine); # Match the Roman Numerals at the beginning of the # next line if (SNextLine =~
m/^\{[IVXLCDM]+\}/) { $RomanNumeral = $1; SYearTrans{$RomanNumeral} = $year; $DocTrans{$year} = $MajorDoc; close InFile; }
print "\n"; }

```

[0056] Then, when the Dewey tag is written out, the Year is placed in the appropriate position, whether it be the first, second, or third level using the Roman Numeral from the currently matched pattern to obtain the correct Year from the hashtable (YearTrans table specifically) as generated above.

Table: 2

1) <meta name="Dewey" content="Seventeenth Antarctic Consultative Meeting%====%====%====%====%====%====% Article XII">
2) <meta name="Dewey" content="CONSERVATION OF ANTARCTIC MARINE LIVING RESOURCES%====%====%====%====%====%====%1981 Article VI Relationship to existing conventions relating to the conservation of whales and seals">
3) <meta name="Dewey" content="Resolutions and Measures adopted at the XIXth Antarctic Treaty Consultative Meeting%====%====%Annex C%====%====%Resolutions%====%====%====%====%Resolution 8">
4) <meta name="Dewey" content="Seventeenth Antarctic Consultative Meeting%====%====%====%====%====%====% Article XII">
5) <meta name="Dewey" content="1980%====%====%====%====%====%1980 Article

[0057] While the systems and methods described herein constitute preferred examples of the invention, it is to be understood that the invention is not limited to these precise forms, and that changes may be made therein without departing from the scope of the invention.

REFERENCES CITED IN THE DESCRIPTION

This list of references cited by the applicant is for the reader's convenience only. It does not form part of the European patent document. Even though great care has been taken in compiling the references, errors or omissions cannot be excluded and the EPO disclaims all liability in this regard.

Patent documents cited in the description

- [US5784608A \[0004\] \[0004\]](#)

KRAV

1. Fremgangsmåde til at hente information fra en informativ ressource (18), og som omfatter trinnene,

5 at opdele (44) den informative ressource (18) i et antal endelige elementer (20a-20z),

at kreere et kategoritag (22a-22z) for hvert af antallet af endelige elementer (20a-20z),

10 at generere (50) en søgbar database, som indbefatter en søgbar databaserecord (24a-24z) for hvert af de endelige elementer (20a-20z),

at indlæse (52) et søgespørgsmål (26), som indbefatter ord eller udtryk,

at søge (54) i den søgbare database efter relevante databaserecords (30), der matcher ordene eller udtrykkene i søgespørgsmålet (26), for derved at producere søgeresultater (32),

15 at associere de relevante databaserecords (30) med tilsvarende relevante endelige elementer (34), således at søgeresultaterne (32) indbefatter de relevante databaserecords (30) og de tilsvarende relevante endelige elementer (34),

at vælge (52) et hierarki til fremvisning af søgeresultaterne (32),

20 at sortere (56) de relevante endelige elementer (34) i hierarkiet i overensstemmelse, i det mindste delvist, med kategoritag'et (22a-22z), som er kreeret for hvert af de endelige elementer (34), og

25 at fremvise (58) søgeresultaterne (32) indbefattende de relevante endelige elementer (34) i overensstemmelse med sorteringstrinnets resultater i en sammenfoldelig og ekspanderbar træstruktur under anvendelse af informationerne i de til hvert af de relevante endelige elementer hørende kategoritags,

30 hvori trinnet at opdele (44) den informative ressource (18) i et antal endelige elementer (20a-20z) udføres med et ekspertsystem ifølge et regelsæt, og hvori fremgangsmåden yderligere omfatter trinnene,

at slutbrugeren som respons på fremvisningen modificerer regelsættet ved utilfredsstillende resultater i fremvisningstrinnet; og at trinnene opdele, kreere, generere, indlæse, søge, associere, vælge, sortere og fremvise gentages ved brug af det modificerede regelsæt.

2. Fremgangsmåde til at hente information fra en informativ ressource ifølge krav 1, hvori,
- 5 at kreere kategoritag'et (22a-22z) indbefatter, at kategoritag'et anvises til hvert af antallet endelige elementer (20a-20z),
- kategoritag'et indbefatter data, der vedrører det endelige elements (20a-20z) indhold,
- hver søgbar databaserecord indbefatter mindst en streng, som er indeholdt i det endelige element (20a-20z), og
- 10 hvori søgning i den søgbare database efter relevante databaserecords omfatter at søge i den søgbare database efter relevante søgbare databaserecords, der indeholder søgestrengen, for derved at tilvejebringe søgeresultater, og fremgangsmåden omfatter yderligere,
- at sortere (56) de relevante endelige elementer (20a-20a) omfatter, at
- 15 søgetrinnets resultater arrangeres i en hierarkisk struktur i overensstemmelse med informationerne i kategoritags'ne (22a-22z), der tildeles de endelige elementer (20a-20z), svarende til de søgbare databaserecords, der blev fundet i søgetrinnet.
- 20 3. Fremgangsmåde ifølge krav 2, hvori den informative ressource (18) er et enkelt dokument, og trinnet at opdele (44) den informative ressource (18) i et antal endelige elementer (20a-20z) indbefatter trinnet at identificere et antal sektioner i det enkelte dokument, hvori hvert endeligt element er en enkelt en af antallet af sektioner.
- 25 4. Fremgangsmåde ifølge krav 2, hvori den informative ressource (18) er et antal dokumenter, og trinnet at opdele (44) den informative ressource (18) i et antal endelige elementer (20a-20z) indbefatter trinnet at identificere antallet af dokumenter, hvori hvert endeligt element (20a-20z) er et enkelt et af antallet af dokumenter.
- 30 5. Fremgangsmåde ifølge krav 2, hvori den informative ressource (18) er en datastrøm, og trinnet at opdele (44) den informative ressource (18) i et antal endelige elementer (20a-20z) indbefatter trinnet at identificere segmenter i datastrømmen.

6. Fremgangsmåde ifølge krav 2, hvori trinnet at tildele (48) et kategoritag (22a-22z) til hvert af antallet af endelige elementer (20a-20z) udføres ved hjælp af et ekspertsystem ifølge et regelsæt.
- 5 7. Fremgangsmåde ifølge krav 2, hvori kategoritag'et (22a-22z) indbefatter data, der vedrører en organisatorisk attribut for det endelige element med hensyn til den informative ressource (18).
- 10 8. Fremgangsmåde ifølge krav 2, hvori kategoritag'et (22a-22z) indbefatter data, der vedrører placeringen af det endelige element (20a-20z).
9. Fremgangsmåde ifølge krav 2, hvori kategoritag'et (22a-22z) indbefatter et kategoriudtryk, der vedrører indholdet i det endelige element (20a-20z).
- 15 10. Fremgangsmåde ifølge krav 2, hvori databaserecorden (24a-24z) indbefatter ualmindelige strenge indeholdt i det endelige element (20a-20z).
11. Fremgangsmåde ifølge krav 10, hvori:
den informative ressource (18) er mindst ét dokument, databaserecorden (24a-
20 24z) indbefatter ualmindelige ord indeholdt i det endelige element, og
trinnet at tilføje en søgestreng involverer et trin til at tildele mindst ét søgeord.
12. Fremgangsmåde ifølge krav 2, hvori,
den informative ressource (18) indbefatter et dokument,
25 trinnet at opdele (44) den informative ressource i et antal endelige elementer
(20a-20z) indbefatter trinnet at identificere et antal sektioner i dokumentet, idet
hvert endeligt element er et enkelt et af antallet af sektioner,
kategoritag'et (22a-22z) indbefatter information, som vedrører en lokalisering af
det endelige element (20a-20z) i forhold til andre endelige elementer (20a-20z) i
30 dokumentet, og
trinnet at arrangere søgetrinnets resultater i en hierarkisk struktur i
overensstemmelse med informationen, som kategoritags'ne (22a-22z), der er
tildelt de endelige elementer (20a-20z), som svarer til de søgbare
databaserecords (24a-24z), der blev fundet i søgetrinnet, indbefatter trinnet at

arrangere de endelige elementer (20a-20z) i en rækkefølge, i hvilken de endelige elementer (20a-20z) viser sig i dokumentet.

13. Fremgangsmåde ifølge krav 2, hvori fremvisningstrinnet indbefatter et trin til på en
5 computerskærm at vise navnene på de endelige elementer, der svarer til de søgbare
databaserecords (24a-24z), der blev fundet i søgetrinnet, og fremgangsmåden
omfatter yderligere trinnene,
at vælge et af de viste navne,
tilgå det endelige element, der svarer til det valgte navn, og
10 at rekonstruere en tilstødende del af den informative ressource (18) omkring det
tilgåede endelige element (20a-20z).
14. Fremgangsmåde ifølge krav 13, hvori rekonstruktionstrinnet indbefatter trinnene,
at gennemgå kategoritaget (22a-22z), der er tildelt det tilgåede endelige element
15 (20a-20z), for data, der svarer til lokaliseringen af det tilgåede endelige element
(20a-20z) i forhold til den informative ressource (18),
på basis af de data, der svarer til lokaliseringen af det tilgåede endelige element
(20a-20z), at bestemme den tilstødende del af den informative ressource, som
skal rekonstrueres, og
20 at hente andre endelige elementer (20a-20z), der hører til den tilstødende del af
den informativ ressource (18).
15. Fremgangsmåde ifølge krav 2, hvori fremvisningstrinnet indbefatter et trin til på en
25 computerskærm at vise navnene på de endelige elementer, der svarer til de søgbare
databaserecords (24a-24z), der blev fundet i søgetrinnet, og fremgangsmåden
omfatter yderligere trinnene,
at vælge et af de viste navne,
at tilgå det endelige element, som svarer til det valgte navn; og
at konstruere en ny informativ ressource (18) til fremvisning under anvendelse af
30 det tilgåede endelige element (20a-20z) og andre relaterede endelige elementer
(20a-20z).

16. Fremgangsmåde ifølge krav 2, hvori tildelingstrinnet (48) indbefatter trinnet at indsætte hvert kategoritag (22a-22z) i det respektive ene af antallet af endelige elementer (20a-20z).
- 5 17. Fremgangsmåde ifølge krav 2, hvori genereringstrinnet (50) indbefatter trinnet at indsætte hvert kategoritag (22a-22z) i databaserecorden, der svarer til den respektive ene af de endelige elementer (20a-20z).
- 10 18. Fremgangsmåde ifølge krav 1, hvori informationen er informationen i en webside, og den informative ressource er et antal websider, der findes på Internettet, fremgangsmåden er yderligere til at fremvise websiden, hvert endeligt element (20a-20z) er en webside, og hvert kategoritag indbefatter data, der vedrører et indhold af hver webside; hver databaserecord indbefatter mindst et ord eller udtryk indeholdt i websiden og i kategoritag'et, der er tildelt websiden, og fremgangsmåden omfatter
15 yderligere,
at tilføje et søgeord eller et udtryk, og at søgetrinnet omfatter at afsøge den søgbare database for søgbare databaserecords, der indeholder søgeordet eller udtrykket, hvori sorteringen af de relevante endelige elementer (20a-20z) omfatter, at arrangere resultaterne af søgetrinnet i en hierarkisk struktur i
20 overensstemmelse, i det mindste delvist, med informationen i kategoritags'ne, (22a-22z), som er indbefattet i de søgbare databaserecords (30), som blev fundet i søgetrinnet, og at fremvisningstrinnet omfatter at fremvise søgetrinnets resultater i den hierarkiske struktur.
- 25 19. Fremgangsmåde ifølge krav 1, hvori fremgangsmåden yderligere er til at søge efter og hente en del af en informativ ressource, hvori hvert af de endelige elementer er en definerbar sektion af den informative ressource, hver definerbar sektion indbefatter et stort antal ord, hver søgbar databaserecord (24a-24z) indbefatter mindst ét ord eller udtryk, som er indeholdt i det endelige element; fremgangsmåden
30 omfatter yderligere at tilføje et søgeord eller et udtryk, og fremvisningstrinnet omfatter at fremvise en identifikator for den relevante databaserecord (24a-24z).
20. Fremgangsmåde ifølge krav 19, hvori den informative ressource (18) er en dynamisk strøm af information, og trinnet at opdele den informative ressource (18) i et antal

endelige elementer (20a-20z) indbefatter trinnet at identificere afbrydelser, der forekommer i den dynamiske strøm af informationer.

21. Fremgangsmåde ifølge krav 20, hvori fremvisningstrinnet (58) indbefatter et trin til på
5 en computerskærm at vise navnene på de endelige elementer (20a-20Z), der svarer til de søgbare databaserecords (24a-24z), der blev fundet i søgetrinnet, og fremgangsmåden omfatter yderligere trinnene,

at vælge et af de viste navne,

at tilgå det endelige element, der svarer til det valgte navn; og

10 at rekonstruere en tilstødende del af den informative ressource omkring det tilgåede endelige element.

DRAWINGS

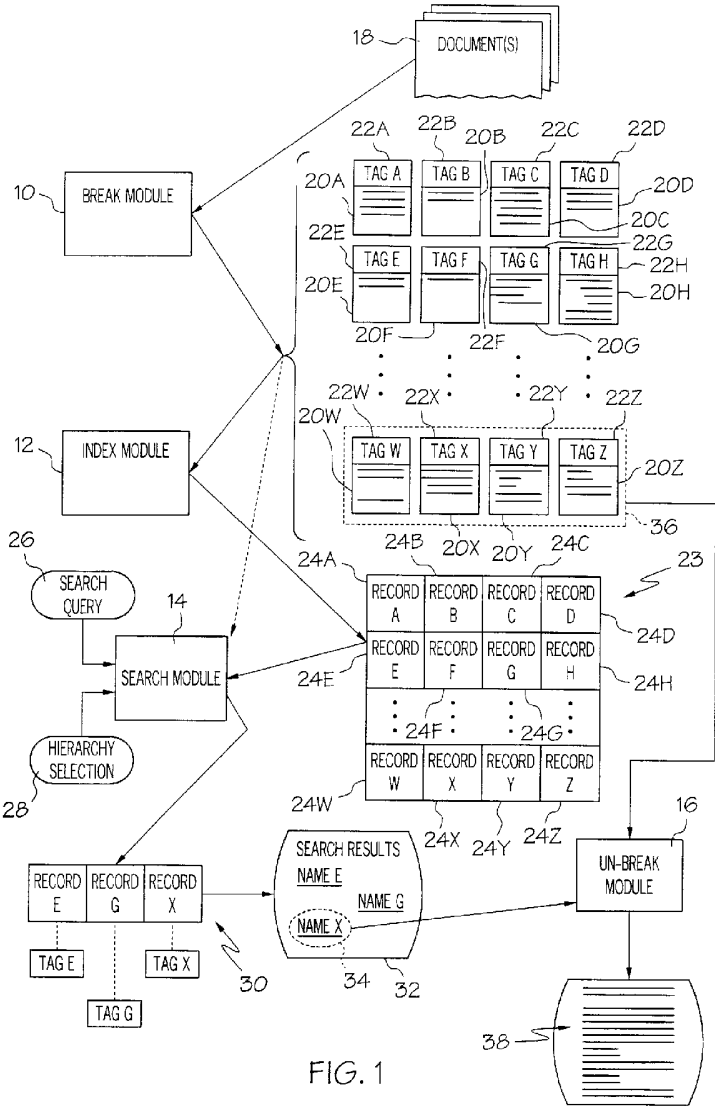


FIG. 1

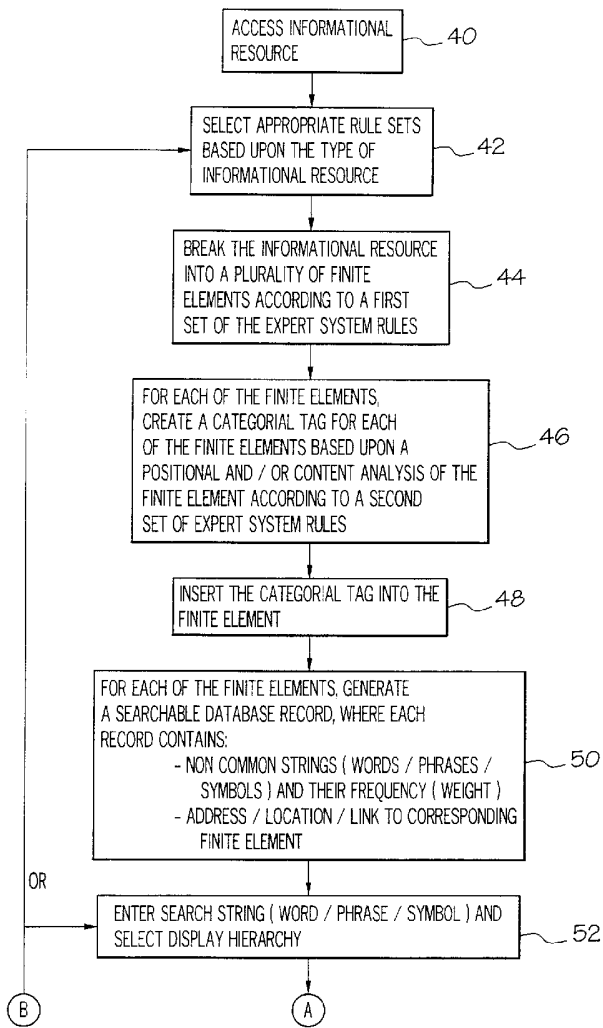


FIG. 2A

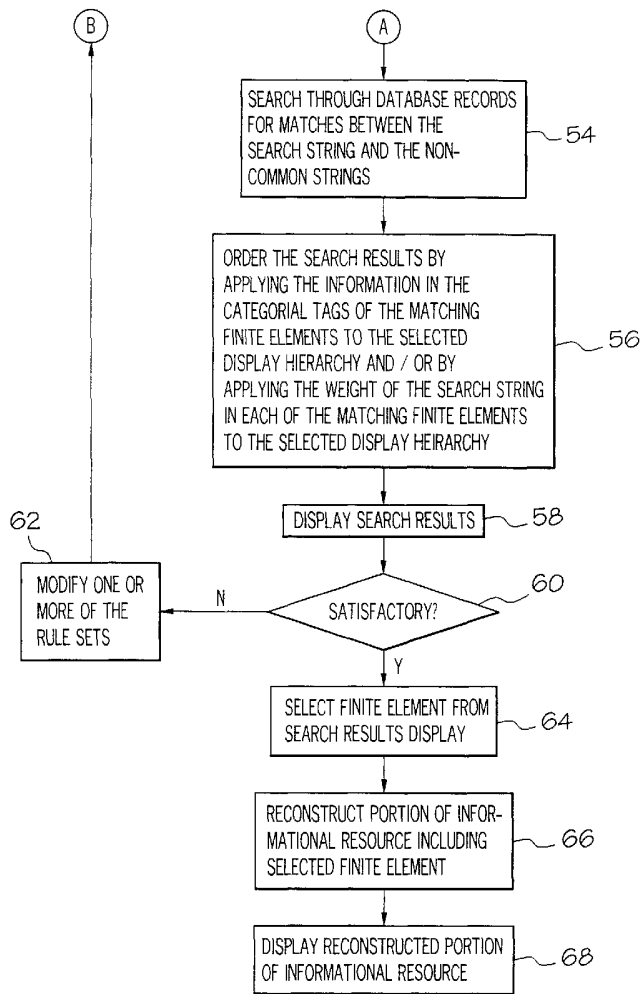


FIG. 2B

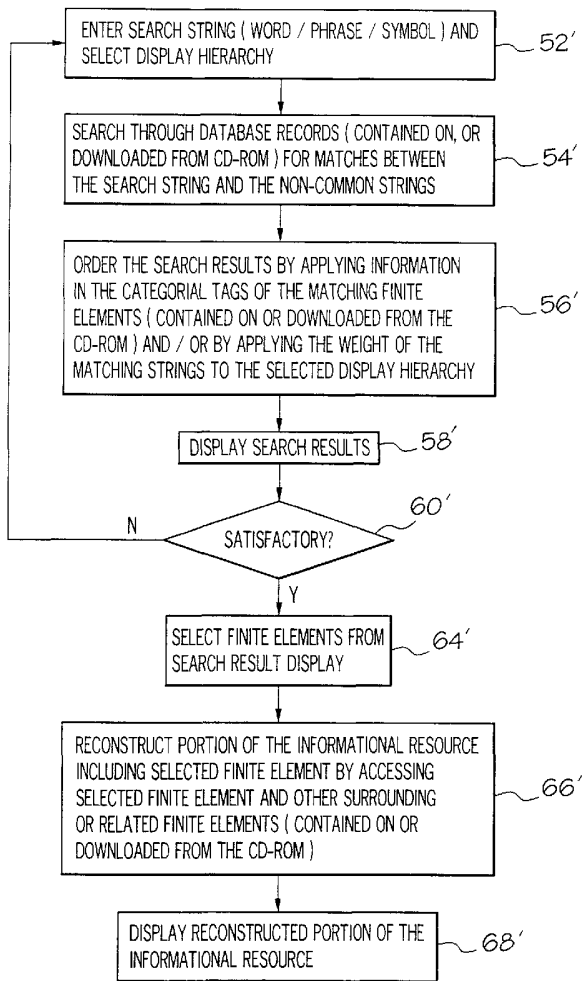


FIG. 3

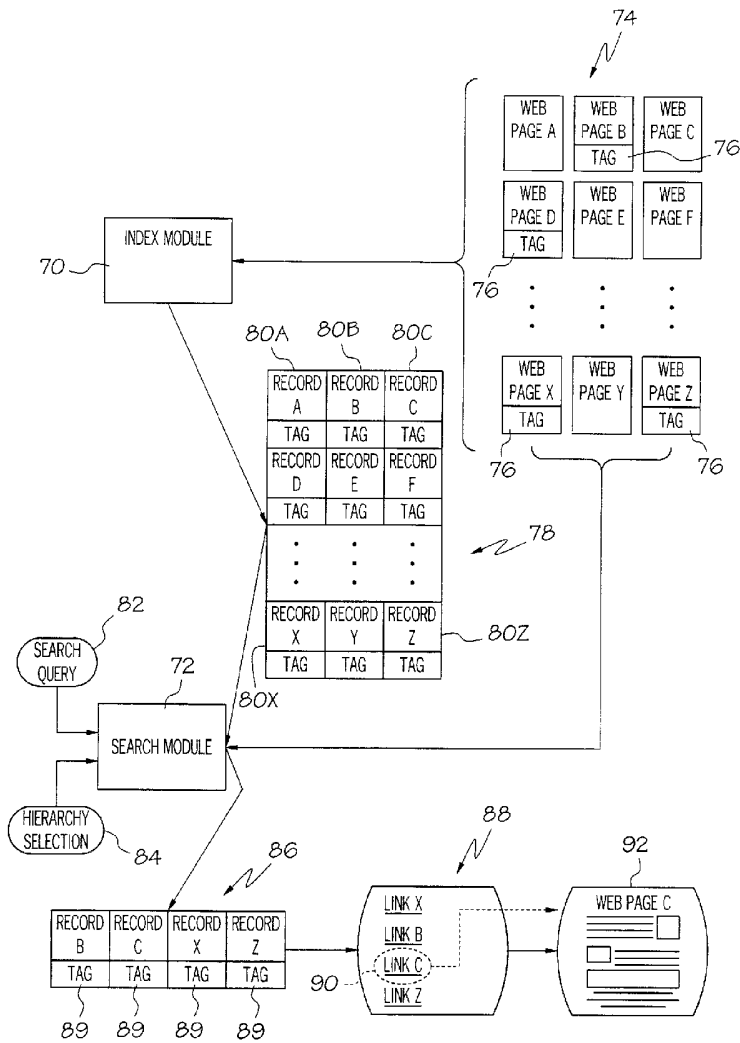


FIG. 4

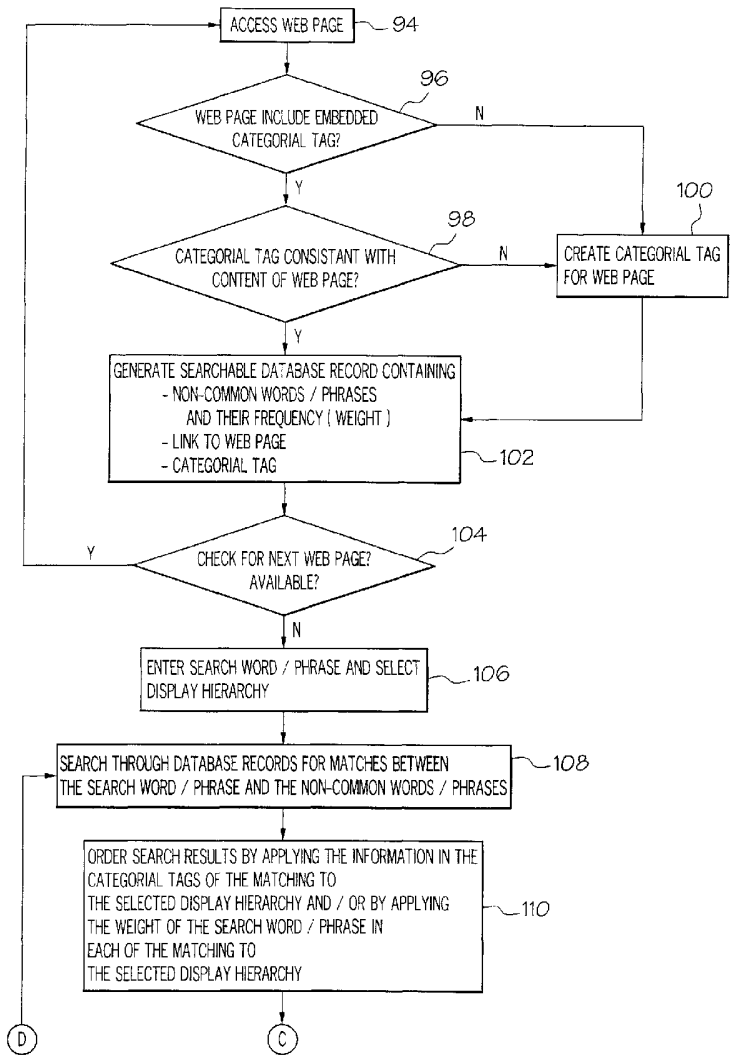


FIG. 5A

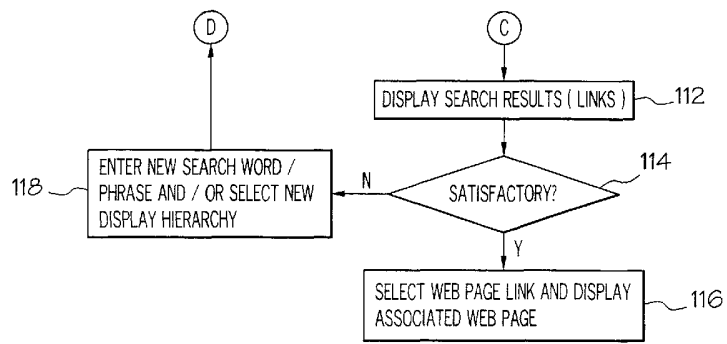


FIG. 5B