

USPTO PATENT FULL-TEXT AND IMAGE DATABASE

[Home](#) [Quick](#) [Advanced](#) [Pat Num](#) [Help](#)
[Bottom](#)
[View Cart](#) [Add to Cart](#)
[Images](#)

(1 of 1)

**United States Patent
Crowe**

**10,552,375
February 4, 2020**

Library indexing system and method

Abstract

A system and method for collectively indexing a collection of documents in paper or electronic form. The collective index is created for all documents, which may come from a variety of sources and be initially presented in a variety of formats. Documents are converted to a standard electronic format, then analyzed and processed to identify, locate, and tag objects of interest. Hierarchical menus are created with multiple categories.

Inventors: Crowe; Joseph Patrick (Nashville, TN)

Applicant:	Name	City	State	Country	Type
-------------------	-------------	-------------	--------------	----------------	-------------

SMITH SECKMAN REID, INC. Nashville TN US

Assignee: SMITH SECKMAN REID, INC. (Nashville, TN)

Family ID: 52584726

Appl. No.: 16/203,889

Filed: November 29, 2018

Prior Publication Data

<u>Document Identifier</u>	<u>Publication Date</u>
US 20190095459 A1	Mar 28, 2019

Related U.S. Patent Documents

<u>Application Number</u>	<u>Filing Date</u>	<u>Patent Number</u>	<u>Issue Date</u>
14478145	Sep 5, 2014	10169349	
61874332	Sep 5, 2013		

Current U.S. Class: 1/1

Current CPC Class: G06F 16/93 (20190101); G06F 16/13 (20190101)

Current International Class: G06F 16/93 (20190101); G06F 16/13 (20190101)

References Cited [\[Referenced By\]](#)

U.S. Patent Documents[6675161](#)

January 2004

Suchter

[6964013](#)

November 2005

Ono

Primary Examiner: Tran; Loc*Attorney, Agent or Firm:* Ramage; Wayne Edward Baker Donelson

Parent Case Text

This application is a continuation of U.S. patent application Ser. No. 14/478,145, filed Sep. 5, 2014, which claims benefit of and priority to U.S. Provisional Application No. 61/874,332, filed Sep. 5, 2013, by Joseph Crowe, and is entitled to that filing date for priority. The specifications, figures and complete disclosures of U.S. Provisional Application No. 61/874,332 and U.S. patent application Ser. No. 14/478,145 are incorporated herein by specific reference for all purposes.

Claims

What is claimed is:

1. A method for creating a collective index for a plurality of documents, comprising the steps of: identifying objects of interest in a plurality of document files in electronic format, wherein objects of interest are not documents in said plurality of document files; creating a single unique object identifier tag for each identified object of interest, wherein the single unique object identifier tag does not identify any documents in said plurality of documents; processing each document file in said plurality of document files, wherein said processing comprises: searching each document file to identify information pertinent to any identified object of interest, where the information has a type; marking, in each document file, information pertinent to an identified object of interest with a unique identifier tag, said unique identifier tag comprising a combination of a form of the unique object identifier tag for the corresponding identified object of interest, and the type of information identified; adding a destination to the document file for that unique identifier tag; and overlaying the marked information within each document file with a button; constructing a plurality of menu files with multiple category levels for said collective index, wherein each identified object of interest corresponds to a button in a menu file; and inserting code text files into the document files and the plurality of menu files, wherein each code text file corresponds to at least one of the identified objects of interest, and comprises pointers to locations within the plurality of document files with information pertinent to the at least one of identified objects of interest; further wherein the code text files inserted into the plurality of menu files cause one or more menus or sub-menus to open a document bearing a same name as a name of a button menu in the one or more menus or sub-menus when a user clicks on a menu button for a particular identified object of interest, wherein the menus or sub-menus present information options available for the particular identified object of interest.
2. The method of claim 1, wherein the document files comes from a variety of sources in different formats.
3. The method of claim 1, wherein the all of the document files and menu files are stored on a single computer-readable storage device.
4. The method of claim 3, wherein the storage device is a computer hard drive.
5. The method of claim 3, wherein the storage device is a USB drive.
6. The method of claim 1, further comprising the step of verifying that each data point for an object of interest is accurately associated with the button and destinations.
7. The method of claim 1, further comprising the step of opening a particular document file to a particular location corresponding to a particular information option selected from said menus or sub-menus.

8. A collective index system for a plurality of documents, comprising: a memory, a computer-readable storage medium with a plurality of document files and menu files stored thereon in a common standard format, wherein code text files have been inserted in said plurality of document files and menu files in accordance with claim 1; wherein each code text file corresponds to at least one identified object of interest in said plurality of document files, and comprises pointers to locations within the plurality of document files with information pertinent to the at least one identified object of interest, wherein objects of interest are not documents in said plurality of document files; further wherein the code text files inserted into the plurality of menu files cause one or more menus or sub-menus to open a document bearing a same name as a name of a button menu in the one or more menus or sub-menus when a user clicks on a menu button for a particular identified object of interest, wherein the menus or sub-menus present information options available for the particular identified object of interest; and further wherein a particular document file is opened to a particular location corresponding to a particular information option upon selection of that particular information option from said menus or sub-menus.
9. The system of claim 8, wherein the storage medium is a USB drive.
10. The system of claim 8, wherein the storage medium is a computer disk drive.
11. The system of claim 8, wherein the storage medium is an external hard disk drive.
12. The system of claim 8, wherein the storage medium is portable.

Description

FIELD OF INVENTION

This invention relates to a system and related methods for indexing a collection of documents in paper or electronic form and creating an intuitive menu structure therefor.

BACKGROUND OF INVENTION

A typical business or commercial enterprise will possess a host of documents in paper or electronic form from a variety of sources. While each may be independently indexed, the documents as a whole are not, thus making it difficult to locate needed reference material quickly and efficiently.

SUMMARY OF INVENTION

In various embodiments, the present invention comprises a computer assisted system and related methods for indexing a collection of documents in paper or electronic form and creating an intuitive menu structure therefor. The collective index is created for all documents, which may come from a variety of sources and be initially presented in a variety of formats.

Initially all documents, papers and data are collected. Electronic form documents are converted to a common standard format (such as PDF), if they are not already in that format. Paper documents are converted to that same common standard format by means known in the art. In one exemplary embodiment, all of the documents in standard format (e.g., PDF) representing the pertinent subject matter to be indexed are placed in a single work or file folder on a storage device (e.g., hard drive) on a computer or server device.

Each document is then analyzed and processed. All buttons and destinations in a document are mapped out. In several embodiments, the processing comprises the creation of a spreadsheet for each type of object of interest. In this embodiment, the left-hand column represents a list or collection of names or titles for each individual object of interest of a certain type (e.g., names of air handling units). The second column contains a unique identifier tag (e.g., an alpha-numeric code) for each named object of interest. Other columns in the spreadsheet represent a combination of the different types of information available for each object of interest and the file locations where that information can be found. While this exemplary embodiment of the

The complete library of PDF documents and menus can then be stored on one or more single computer storage devices (e.g., external disk drive, computer hard drive, portable drive, USB drive, or the like), and provided to end-users. The end-user can initiate the menu file and proceed through the menus to find the information needed, then click on the appropriate link or button to go to that information in the original PDF document, wherever located.

DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a chart of a library indexing process in accordance with an exemplary embodiment of the present invention.

FIG. 2 shows a chart of a cross-indexing and menu construction process from FIG. 1.

FIG. 3 shows an example of an object of interest spreadsheet.

FIG. 4 shows a document file being analyzed.

FIG. 5 shows another example of an object of interest spreadsheet.

FIG. 6 shows an example of a completed object of interest spreadsheet.

FIG. 7 shows an example of a button overlay in a document file.

FIG. 8 shows an example of an object of interest list.

FIG. 9 shows an example of a category hierarchy.

FIGS. 10-16 show examples of object of interest menus.

FIG. 17 shows an example of a top-level category menu.

FIG. 18 shows an example of a second-level category menu.

FIGS. 19-20 show examples of menus with identifier tag overlays.

FIG. 21 shows an example of a folder with text files.

FIG. 22 shows an example of a pop-up window with menu.

FIG. 23 shows an example of a code-verification spreadsheet.

DETAILED DESCRIPTION OF EXEMPLARY EMBODIMENTS

In various embodiments, the present invention comprises a computer assisted system and related methods for indexing a collection of documents in paper or electronic form and creating an intuitive menu structure therefor. With reference to FIG. 1, initially all documents and data are collected 100. Electronic form documents are converted to a common standard format (such as PDF), if they are not already in that format. Paper documents are converted to that same format by means known in the art 102. In one exemplary embodiment, all of the documents in standard format (e.g., PDF) representing the pertinent subject matter to be indexed are placed in a single work or file folder on a storage device (e.g., hard drive) on a computer or server device. Each document is then analyzed and processed (Phase 1, as seen in FIG. 2). All buttons and destinations in a document are mapped out 104.

In several embodiments, the processing comprises the creation of a spreadsheet for each type of object of interest. An example of a spreadsheet for "Air Handling Units" 200 as the object of interest is shown in FIG. 3. In this embodiment, the left-hand column 202 represents a list or collection of names or titles for each individual object of interest of a certain type (e.g., names of air handling units). The second column 204

contains a unique object identifier tag (e.g., an alpha-numeric code) for each named object of interest. Other columns in the spreadsheet represent a combination of the different types of information available for each object of interest and the file locations where that information can be found. While this exemplary embodiment of the invention uses a spreadsheet, it should be noted that any similar form of data file or program type used for storing information can be used.

Each document file is reviewed and searched to find subject matter pertinent to the objects of interest. FIG. 4 illustrates an example of the process where HVAC equipment is the general subject matter, so that any documents depicting the installation, operation, maintenance, repair, controls, or other pertinent information pertaining to HVAC equipment is cross-indexed (as described below). When a piece of information relative to any object of interest is located within a document, the exact view of the information 300 is marked with a unique identifier tag 302. While the unique identifier tag can be in any form, in the embodiment shown, the unique identifier tag is a two-part alpha-numeric string denoting a combination of the object of interest (e.g., Air Handling Unit), and the kind of information pertaining to the object of interest (e.g., drawing plan view of the unit), with an underscore character used to separate the parts. The page view 300 best representing the floor plan view of an air handling unit named "AHU-DT-10-01" thus would receive the unique identifier tag of "FLOOR_DT1001." A "destination" 310 is added to the PDF document file with that unique identifier tag in the list of destinations 320 within the document, thus representing the location 304 and view of that information within the PDF document file.

After a destination 310 is added to the list of destinations 320 in the PDF document, it is marked on the spreadsheet, as seen in FIG. 5, within the row representing the named object of interest and the column representing the PDF file name 210 where the designation resides (e.g., "Mech Drawings.pdf"), the type of information the destination is marking 212 (e.g., "Floor Plan"), or both.

This process is completed for each object of interest in the spreadsheet assigned by type until all PDF documents have been thoroughly searched, analyzed, and provided with destinations. The spreadsheets are simultaneously updated to reflect the destinations created with the PDF documents. FIG. 6 shows an example of a complete spreadsheet, with all objects of interest of a certain type 400 (in this case, air handling units) associated with the relevant PDF document files and the locations within those documents.

If an object of interest was depicted within the PDF documentation, it is overlaid in the document with an invisible button or link 500. The button/link is named 502 after the unique identifier tag of that object of interest, as seen in FIG. 7. As described later, these buttons/links are provided with Javascript (or similar coding) based on the information within the corresponding spreadsheet.

After completion of the processing of all documents (Phase 1), the process proceeds to conceptual menu construction 110 (Phase 2). It should be noted that code construction (Phase 5, described below) can be performed simultaneously with Phase 2.

For Phase 2, all objects of interest cataloged in the spreadsheets described above are now organized by type, using name and unique identifier tag, into a simple columnar list format in multiple type spreadsheets, as seen in FIG. 8. Each type is given a title 600, which in the embodiment shown is placed at the top of the list of objects of interest for that type (e.g., "Air Handling Units"). As described in more detail below, this title is subsequently used for the menu title for a menu representing the type, and for each object of interest belonging to this type, there is a name 202 and unique object identifier tag 204 provided below the title.

Each type is then assigned to a category (i.e., a collection of similar types). In a process similar to that described above for types, each category is given a name, and a category spreadsheet is generated where all types and categories are listed under higher level categories. This process is repeated for all categories in the spreadsheet until one final level category is reached (above all sub-categories and their types). In one exemplary embodiment, the end result is a spreadsheet where each row represents a category or a type of object of interest, and each column indicates the level of the category. FIG. 9 shows an example with one level of sub-categories; however, the process allows for many levels of sub-categories between the highest-level category (e.g., "Home Menu") and the types of objects of interest. Each sub-category belongs to the nearest higher-level category above it (i.e., to the left, in FIG. 9), and each type of object of interest belongs to the sub-category above it.

FIG. 9 shows an example with two category levels, and one type. The top category level 610 is named "Home Menu," and there is one sub-category level 612 with sub-categories "Mechanical," "Electrical," "Fire Protection," and "Medical Gas." Each sub-category has several types 600. Thus, the "Air Handling Units" type belong to the "Mechanical" sub-category, which belongs to the "Home Menu" final category.

The menu category spreadsheet file may be reviewed to ensure that every category or sub-category falls within a higher level category up to the top level, and that each type of object of interest belongs to the lowest level of category. Then the spreadsheets cataloging the objects of interest are reviewed to ensure that all objects of interest belong to a category listed in the menu spreadsheet file.

The process then proceeds to menu face construction 112 (Phase 3), which generates the object of interest menus. In one exemplary embodiment, a Visual Basic macro is executed on each spreadsheet cataloging the objects of interest to examine the quantity of objects of interest, and determine the best layout for the objects to be displayed as buttons in a menu. It then creates a menu for the type of the group, and each object of interest within it is represented by a button. In one embodiment, the type of menu created is a PowerPoint menu, as seen in FIGS. 10-16. The title 620 for each object of interest menu is taken from the title given to the objects of interest list (as seen in FIG. 8), and the object of interest names are applied to individual buttons 622 in the menu. The number of buttons in the menu is automatically determined and placed depending on the total number of objects of interest to be included in the menu.

The menu category spreadsheet file undergoes a similar process to create the Home Menu and the various category and sub-category menus. The process (which can be executed through a Visual Basic macro, or other similar program) examines the quantity of sub-categories for each higher-level category, and determines the best layout for the objects to be displayed as menu buttons. It then creates a menu (such as a PowerPoint type menu) for each category, where the menu title is the category name, and each button represents one of the sub-categories for that category.

FIG. 17 shows an example of a top-level category 610 "Home Menu" with the title "Interactive Maintenance and Engineering Documentation" 630. Several sub-categories are listed and shown as buttons. Once the menus are completed, using the "Mechanical" button 632 from this menu, for example, displays the second-level category menu for "Mechanical Equipment," as seen in FIG. 18, which contains buttons for the objects of interest menus (as described above).

Once the basic menu faces are constructed for all category and type menus (e.g., PowerPoint menus, in the examples shown), the menus are then completed by generating standard format menus 114 (Phase 4). In one exemplary embodiment, where the document files are in PDF format, the completion process generates PDF menus using the basic menu faces. A Visual Basic macro (or other similar program) opens each PowerPoint menu, generates an identical PDF menu, and uses the unique identifier tag information from the button in each PowerPoint file to overlay the PDF with buttons in the same arrangement, with each button 700 tagged by its unique identifier tag 702, as seen in FIG. 19. The buttons correlate with objects of interest, and the button for each object of interest is named after its tag. For example, the object of interest "AHU-DT-10-01" is overlain with a button named "DT1001." The identifier tag can be used to reference the information provided in the spreadsheets cataloging the objects of interest generated in Phase 1, as seen in FIG. 6.

The system similarly generates a PDF menu for each category and sub-category menu face, with the PDF overlain with buttons 700 in the same arrangement, as seen in FIG. 20. Each button is tagged the same as it was named in the PowerPoint file, with a .pdf suffix added 704.

Separate from the menu construction process, the system proceeds with constructing code text files 120 (Phase 5). As noted above, Phase 5 can be performed simultaneously with Phases 2-4.

Phase 5 uses the object of interest spreadsheet file (as seen in FIG. 6) for each type, where all objects of interest have been associated with relevant document files (e.g., PDF document files), and the locations within those files. A macro program (such as a Visual Basic macro program) is executed on each of these type spreadsheet files, and generates a text file for each object of interest based on the information provided for that object (i.e., the information provided in that object's row), wherein the text in the file is in the form of code. In one embodiment, the macro program arranges the information from the spreadsheet row into a Javascript code format which can be placed into PDF files. Each text file is named after the unique identifier

typically is stored in ROM. In contrast, RAM typically contains data or program code or modules that are accessible to or presently being operated on by processor, such as, but not limited to, the operating system, application program, and data.

Client devices also may comprise a variety of other internal or external components, such as a monitor or display, a keyboard, a mouse, a trackball, a pointing device, touch pad, microphone, joystick, satellite dish, scanner, a disk drive, a CD-ROM or DVD drive, or other input or output devices. These and other devices are typically connected to the processor through a user input interface coupled to the system bus, but may be connected by other interface and bus structures, such as a parallel port, serial port, game port or a universal serial bus (USB). A monitor or other type of display device is typically connected to the system bus via a video interface. In addition to the monitor, client devices may also include other peripheral output devices such as speakers and printer, which may be connected through an output peripheral interface.

Client devices may operate on any operating system capable of supporting an application of the type disclosed herein. Client devices also may support a browser or browser-enabled application. Examples of client devices include, but are not limited to, personal computers, laptop computers, personal digital assistants, computer notebooks, hand-held devices, cellular phones, mobile phones, smart phones, pagers, digital tablets, Internet appliances, and other processor-based devices. Users may communicate with each other, and with other systems, networks, and devices, over the network through the respective client devices.

Thus, it should be understood that the embodiments and examples described herein have been chosen and described in order to best illustrate the principles of the invention and its practical applications to thereby enable one of ordinary skill in the art to best utilize the invention in various embodiments and with various modifications as are suited for particular uses contemplated. Even though specific embodiments of this invention have been described, they are not to be taken as exhaustive. There are several variations that will be apparent to those skilled in the art.

* * * * *

