

CONFIDENTIAL

5 Figures 2f and 2g show two different embodiments which utilize an outer
 zone and inner zone for allowing the user to select between the creation of a
 pile where the user can control the appearance of the pile and the creation of a
 pile where the user delegates to the system the task of placing the document
 neatly on top of a new pile. An inner zone 77 is contained within the interior of
 the circle shown over an icon of a document 75 in Figure 2g. The inner zone 77
 is used to indicate to the computer system that the user desires to indicate to the
 computer system that the user desires to control the positioning of documents in
 10 the pile so that the appearance of the graphical representation of the pile is, at
 least to the extent of the orientation of documents in the pile, controlled by the
 user. An outer zone 76 is contained within the circle shown over the icon 75
 excluding the region in the inner zone 77. That is, the outer zone 76 is an
 annular ring having two boundaries defined by the outer circle of Figure 2f and
 15 the inner circle of Figure 2g; this annular ring may be thought of as the
 subtraction of the area within the inner circle from the area within the outer
 circle. The outer zone 76 is used to indicate to the computer system that the
 user desires to allow the computer system to control the positioning of
 documents in the pile. The user may when adding one document to the pile
 20 use the inner zone 77 to control positioning, and then when adding another
 document to the pile the user may use the outer zone 76 to allow the computer
 system to control the positioning of the new document in the pile. The circles of
 Figures 2f and 2g would typically not be shown during use of the invention, and
 they are shown in these figures for purposes of illustration. The shape and size
 25 of the inner and outer zones is a design choice which may be varied to suit the
 user's desire. The operation of using the inner and outer zone will be described

zone of the document 450. Then the user indicates that the selected document is no longer selected (or that a pile is to be created) and the computer responds with the animation shown in stages 457 and 459 to create a neat graphical representation for the pile over the base 461.

5

The operation of creating a new pile while maintaining user control via the appearance of the pile is shown in Figure 11b and proceeds in a manner similar to that described for Figure 11a except that the user positions the cursor 467 over the inner zone 466 of the document 450. Thus, as shown at stage 465 of Figure 11b, the cursor 467, after having selected a document and moving the document and the cursor while keeping the document selected, is positioned over the inner zone 466 of the document 450. This is also shown at stage 470 of Figure 11b where the cursor 467 is shown over the original document 482 which is selected, and the cursor 467 is also over the inner zone 466 (not shown). Upon the user's reaching the inner zone of the bottom document icon 450, the computer responds at stage 473 by inclining and rotating the document icons and then by observing the position the user indicates by moving the cursor 467 relative to the bottom, document icon 481. As long as the top document icon is left touching the bottom document icon, the computer follows the positioning of the top document relative to the bottom document and allows the user to position the document at any location. If the user moves the cursor outside of the bottom document icon such as the top document icon is no longer touching the bottom document icon then the computer system reverts back to showing two separate document icons which are not inclined and rotated.

2.5 When the user is satisfied with the position of the top document icon relative to the bottom document icon user releases the selection on the top document (by releasing the mouse button) which causes the computer to respond as shown at

02756222 022704

FOR OFFICIAL USE ONLY

stage 475 by creating a pile having a base 461 with the top document aligned according to the user positioning of the top document relative to the bottom document as described previously. In an alternate embodiment of the use of the inner zone 466, the user moves the selected first document over the inner zone 466 of the document 450 so that at least a portion of the inner zone (or some other central "hot spot") of the selected document 482 is over a portion of the inner zone of the bottom document 450 for at least a predetermined period of time. Then the user indicates that the selected document is no longer selected (or that a pile is to be created) and the computer responds with the animation shown in stages 473 and 475 to allow the user to position the top document 482 over the bottom document 450 in the manner described above.

Figures 12a and 12b show the outer and inner zone designs respectively for the process of adding an item to a pile. The inner and outer zones are shown in Figures 12a and 12b for the purpose of illustration. The operation of adding to the pile and allowing the system to control the location of the new document, as shown in Figure 12a, is similar to the operation described relative to Figure 11a. In particular, the user selects a document and moves the document over to the pile and positions the cursor 503 (or alternatively the inner zone/hot spot of the selected document) in the outer zone of the pile as shown in stage 501. As shown in stage 505, the document 506 which has been selected and moved along with cursor 503 is positioned over the pile and the cursor 503 is positioned in the outer zone. Upon releasing the selection of the document while the document and the cursor is positioned over the pile and particularly the cursor is positioned over the outer zone, the computer responds at stage 511 by inclining the document 506 as it is positioned by the system on top of the pile. At stage 514, the operation is complete as the icon 506 is shown

09756392 022701

on top of the graphical representation of the pile, which representation is dynamic as illustrated in the figures. The operation shown in Figure 12b allows the user to add a document anywhere in the pile. This is accomplished by selecting the document and taking the document with the cursor 517 and
5 positioning the cursor (or alternatively the inner zone/hot spot of the selected document) over the pile and particularly over the inner zone 516 of the pile shown at stage 515. After a predetermined period of time, the computer responds as shown at stage 520 with the view cone 518 and the proxy 519 within the view cone, which proxy shows the document immediately underneath
10 the current location in the pile which the user has selected by placing the cursor 517 (or alternatively the inner zone/hot spot of the selected document) into the inner zone. If the user is satisfied with this location the user releases the selection by typically releasing the mouse button on the mouse causing the pile appear as shown at stage 525.

15 The invention provides a method for the user to include a user defined specification, such as a script, for a pile. A user defined specification may be provided by the user (e.g. typing keywords as a specification or selecting a sample document, having an internal representation which is described below
20 and which provides the specification) or may be suggested by the system and ratified by the user. Normally, the user can control whether a pile has such a specification and thus the system will not create such a specification without the user's approval. Not every pile needs or will have a user defined specification, although every pile will have an internal representation (described below) or a
25 user defined specification.

09796882 022701

A tool for visualizing and organizing the contents of a pile will now be described by referring to Figure 13a and 13b. A pile 550 is shown within the visualization window 551 of Figure 13a. Typically, the user would select the pile 550 and then select a command from the computer which would invoke and display this visualization window with the selected pile 550 within the window 551. The window 551 includes several command options which allow the user to view the pile in different ways and to order the contents of the pile in different ways and to select between those different ways in order to obtain different appearances of the pile or to otherwise reorganize the pile (e.g. create subpiles from a single original pile). For example, by placing the cursor over the arrow icon 555 and by selecting the icon 555 by depressing the mouse button or otherwise signalling to the computer to select the arrow icon 555, the user causes the computer to display a pull down menu indicating the various ways in which the contents of the pile may be ordered. The system shows in box 556 that the pile is currently ordered by date which means either that the most recent items are on the top or the oldest items are on the top depending on the designer's preference. Other options which may be included in the "order" pull down menu (activated by selecting arrow 555) may include: ordering by content; ordering by size; ordering by author (alphabetically). The operation of the pull down menu is well known in the art and has been described for example in U.S. Patent Reissue 32,632. The arrow icon 560 allows the user to select the way in which the pile is colored. As shown in box 561 the system currently has colored the pile by date in which, in one embodiment, the most saturated shade of a color is displayed near the bottom of the pile (indicating the older documents) then the top of the pile has a lesser saturated shade of the same color. It will be apparent to those in the art how to implement this function by assigning colors; for example picking a color in a perceptual color space

05756222 002701

(e.g. Munsell color space) and varying the saturation of the color holding the luminance constant and the hue of the color constant. The range of the dates would be used to set the outer limits of the saturation. That is, the oldest and most recent documents would form the extremes of the saturation range and a date in the middle of that range would specify a color in the middle of the saturation range.

Other ways of presenting various colors according to a measure of the date/age of the document will be appreciated by those in the art, particularly by referring to art relating to color management systems in computers. For example, the hue may be varied (e.g. from red to blue) while keeping a constant saturation and luminance. Alternatively, preselected colors (and the color values) may be used in a look-up table for mapping the measure of date/age of the document to one of the preselected colors in the look-up table.

In a similar manner, the user may desire to view the color of the pile according to the content of the pile. This would be done by the user selecting the arrow icon 560 to cause a pull down menu to appear below the arrow icon 560, which pull down menu would include the command option of "coloring by content." The user would then select this command option placing the cursor over that command option and releasing the selection (e.g. by releasing the mouse button) which then causes the word "content" to appear within the box 561 indicating that the system will color the pile by content. This will be described in more detail below particularly with reference to Figure 20.

The visualization window 551 also includes an arrow icon 565 which activates a pull down menu to allow the user to select the manner in which the

05756292.002701

pile is organized (e.g. "pile by content"). As shown in Figure 13a, the user has selected "pile by content" which may cause the system (depending on how related all documents are within the pile 550) to create subpiles from this pile 550. The user could have selected other options available in the pull down menu from icon 565, such as, "pile by date," "pile by stamps," "pile by data type", "pile by source," etc. Let us assume that pile 550 is a pile of mail messages obtained through electronic mail which the user has moved out of a mail window such as window 61 and has selected this pile for visualization in a visualization window 551. Because it is a mail message pile, it is likely that the contents of the various documents in the pile will vary over a large range and that the pile would be better organized into several subpiles according to content. Accordingly the user has placed the pile into the visualization window 551 and desires to pile by content; that is, the user desires to make subpiles from the original pile. The user does this by selecting a command to indicate that the pile should be piled by content. In the embodiment shown in Figure 13a, the user selects this by selecting the "arrow" button 565 which the user may do, for example, by placing the cursor over the region of the button 565 and pressing and releasing the mouse button or by otherwise signalling to the computer to make this selection. Upon making the selection, visualization window 13b appears wherein three organized piles 575, 576 and 578 are displayed along with a miscellaneous pile ("other") 577. The system has, in addition to creating the subpiles, provided a name for each pile in the manner described below, particular with reference to Figure 15. At this point, the user may instruct the computer system to infer a user defined specification, for example, a script, from the system created subpiles which could then be used to automatically file new mail documents received over the electronic mail network into one of the subpiles according to the particular script of the subpile. The

05736292 022704
102220 262626

system infers the script by using the system provided internal representation of each subpile; for example, the 'N' (where N is an integer) most characteristic words in the pile's internal representation (e.g. the pile's vector described below) could be used. The user could then review the list of the 'N' (set by
5 some system threshold value) most characteristic words and approve or disapprove of all or some of these words and then create the script. More details concerning scripts will be presented below with reference to Figure 14.

The piles in Figure 13b were created from the original pile 550 by
10 allowing the system to create the subpiles as described in the manner below. In this way, the user of the computer may more efficiently review the various mail messages and other documents within the original pile rather than having to browse through the pile to see the keyword proxy of Figure 4f or otherwise select documents from the original pile. Thus, for example, if the user is most
15 interested in "art, design" documents, then the user may begin looking at documents in that pile first and defer examining the other subpiles or perhaps the "notebook computer" pile depending on the interests and priorities of the user. As will be described in more detail, the system analyzes each document in the original pile based on the content of the document and then determines
20 the most appropriate subpile in which to place the document. Each subpile will have a representation of the content of the documents within the subpile and that representation will be compared against the representation provided by the system for the particular document. The closer the match between the two representations and the more likely the particular document will end up in the
25 particular subpile.

CONFIDENTIAL

If the user is satisfied with the results of the subpiling process shown in Figure 13b, the user may retain this structure by selecting the "Ok" button icon 571. Otherwise, the user may cause the documents in the subpiles to be reassembled into the original pile 550 by selecting the "Revert" button icon 570.

5 If the user selects "Ok" 581, the filing system creates a new pile for each subpile, which in one implementation involves changing the pathname of each document in each subpile in the manner described above.

Figure 14 shows another tool, a script window 601, for manipulating piles

10 which have a user defined specification (e.g. a script) and controlling the way they work in the computer system. The user may obtain the window 601 or other windows for creating and/or editing scripts in various ways, such as providing a command to the computer to display such a window on a display screen 22 (e.g. a "Create script" command) or adding a document to a pile that

15 already contains a script where the document's content does not match the script in a close enough way and where the system has been instructed to so notify the user when this occurs. In either instance, the script window 601 appears on the display screen 22. If the window was invoked by adding an item to a pile and the user does not mind that the content of the document does not

20 precisely match the script of the pile, the user may select the command "just add this item" by selecting this command in the various ways known to select the command in a graphical user interface. This would typically take place by positioning the cursor near the button icon 602 and selecting this command while the cursor is positioned over the button 602. If the user desired to modify

25 the pile's script to accommodate the new document which was added then the user may select this option by placing the cursor in the button 603 and selecting that option (e.g. by depressing and releasing the mouse button while the cursor

09756292 002704

and releasing the mouse button. This causes the system (after the user selects the command) to pull in all documents into the pile controlled by this script which contain "design competition update" in the regarding line/field of the document. In this manner, the system may automatically classify information for the user and the user may instruct each pile separately according to the needs and criteria of the user or the basis of a user defined specification.

Region 611 of the script window 601 contains a list provided by the system which indicates the most unique used words in the particular document from Richard Mander (i.e. the document which was added to the pile and then caused the system to display the script window 601 on the display screen 22). As will be described below, the system can provide a ranked list of the most frequently used words in a document which also best characterize the document relative to all other documents in the file system of the computer. This is done by the use of, in one implementation, the vector described below for the document. In this implementation, each word listed in region 611 is one component of an n dimensional vector which describes the document and provides a representation of the contents of the document. As shown in script window 601, the word "design" is the most characteristic word followed by "competition" and "interface". The system provides this listing in region 611 and the user may scroll through the various words in this listing by selecting the scroll arrows 627 and 628 or by using the scroll bar 629. The scrolling of the contents of region 611 is accomplished in the normal manner of the prior art. The user may select any one or all or some of the words in region 611. When these words are selected, they appear in region 612 indicating to the system that the pile's script should be modified such that the representation of the pile (e.g. the pile's vector), as described below, is modified to take into account the

09796633-022701

words selected by the user. Alternatively, the user may type words into region 612 to thereby modify the script of the pile. In this manner, the script of the pile is modified to take into account the new document from Richard Mander which was added to the pile by the user before the script window 601 was displayed.

5 For example, the user could select "competition" and the word "children" and these words would then appear in region 612; when the user selects "add to script" button 620 this causes the system to update the script for the pile causing the internal representation of the pile to be modified in order to allow the user to control the way the script behaves in the computer system. The selection of the

10 words in region 611 may be accomplished in any of a number ways which are familiar to user's of computer systems having graphical user interfaces (e.g. positioning the cursor over a word and selecting the word or "shift-clicking" to select multiple words).

15 In a similar manner, the script of the pile may be modified to take into account labels which specify certain criteria such as label 614 shown in region 630. These labels contain certain "canned" criteria specified by the user and the user may select these labels by positioning the cursor over the particular label and selecting the label causing the icon of that label to appear in region

20 617. These labels may be the programmable user interface elements described in the application having serial number 07/700,729 which was filed May 15, 1991 and is entitled "A User Interface System Having Programmable User Interface Elements" and is assigned to the assignee of this application. These labels provide a programmable means for causing the execution of a

25 series of instructions and may include, for example, a requirement that all documents in a pile be dated after a certain date.

09756532 082701

Once the user has determined and selected the various criteria which are to be used to modify the internal description of the pile (pile's script) then the user may indicate to the system that the script should be modified by selecting the "add to script" button 620 and then by selecting the okay button 623 which
5 causes the execution of the script wherein the system checks for documents managed by the filing system which match the modified script. If the user does not like certain criteria which have been previously selected, the user may cancel the selection by selecting the cancel button 625. If the user selects the "script..." button 621 the system provides a window for the user to type in words
10 or other criteria which will then be used to modify the script of the pile. This modified script may then be selected by selecting the add to script button 620 and okay button 623. If the user has requested a script window similar to window 601 while not adding to a pile, then the user desires to modify the script of the pile; this is accomplished by entering commands into a script window in a
15 manner similar to that described for window 601.

The manner in which the filing system of the computer takes into account the representation of each document and each pile and performs the various operations with piles which have been described above, will now be described
20 by referring to the various flow charts of Figures 15 through 21.

Figure 15 shows one method according to the present invention for creating the internal representation used by the filing system in organizing piles, which internal representation is for each document in the filing system (or for a
25 certain selected group of documents as specified by the user) and for every pile maintained in the filing system (or for piles which have been selected by the user to have their internal representation determined). Typically, the filing

09796232 022704

system will perform this operation for every document (usually user documents, not system documents such as operating system files) and for every pile in the system. If the filing system is using the method of the invention for the first time, then the method shown in Figure 15 must be applied to every document in the system; that is, every document in the system must be treated as a new document. The method 700 for indexing documents begins at step 701 which is merely a node point in the process. The first operational step of the system involves a test for whether the document is a new document at step 702. A modified document is considered a new document as well as a new document which the user creates or which comes into the user's computer system via electronic mail. It is also noted that a document that is deleted affects the internal representation of all other documents in the system and the designer of systems using the present invention may want to eventually (e.g. in background processing on the computer system) recompute each document's vector and each pile's vector in this circumstance also. If there is a new document in the computer system processing proceeds to step 703 in which the computer system counts the number of times certain words are used in a document. Only certain words are counted because many words (e.g. the, it, a, and, etc.) are used too commonly and would not provide a good internal representation of the contents of any document. Therefore, these words (referred to as "stop words") are discarded in analyzing and indexing documents. This may be accomplished by the system establishing a "stop word" table which is examined for each word in a document which is being indexed. The system should also "stem" each word being indexed in order to prevent unnecessary redundancies between the singular and plural of a word (e.g. "horse" and "horses") and different tenses of a verb and other techniques which are known in the art may be applied for stemming the words (e.g. child and children would be considered

09756252 002704

After indexing a new document, step 703, processing proceeds to determining the number of documents in the system (documents which are previously indexed) which use each of the certain words in the new document. This is determined by examining the global document frequency table, which
5 may be a hash table, for each word in the document which has been indexed. If all documents are new in the system, a special initialization process which is accomplished in two phases must be performed in order to properly index all documents in the system. In the first phase, word frequencies are computed for all the documents; in other words, step 703 is performed for each document in a
10 system and then the global document frequency count table will then be completed. In the second phase the document vectors are calculated, which is shown as step 707, for each document in a system. Step 704 performs, in effect, this two phase initialization process by forcing step 703 to be repeated when system initialization occurs; step 703 is repeated until all documents are
15 indexed and then the global document frequency table is completed. System initialization may be determined by searching for the global document frequency table; if it exists when a new document is indexed in step 703 then the system has been initialized and otherwise the system has not been initialized and initialization proceeds until all documents are indexed in step
20 703.

In step 707 the internal non-normalized representation of a document is determined by calculating the document's vector. This is done by determining, for each of the non-stop words in the document the ratio of the number of times
25 the particular word in the document is used ("tf"--obtained from the document's table) divided by the number of documents in the system with the same particular word ("df"--obtained from the global document frequency table). This

09798222 002704

established weights. It will be appreciated that if the method 700 for indexing a document is being performed during initialization of the system then the determination of a pile vector will occur after all documents have been indexed and the global document frequency table has been completed. After step 719, processing loops back to node A wherein the system waits for an interrupt or other messages indicating a new document has been placed into the filing system of the computer.

The methods employed in steps 703, 705, 707, 709 and 711 are known in the art and various ways of determining an internal (or content) representation of a document are available in the information retrieval art. See, e.g. "Term-Weighting Approaches in Automatic Text Retrieval," G. Salton and C. Buckley, Information Processing & Management, Vol. 24, No. 5, pp. 513-523, 1988. However, these methods have not been used in the context of piles as described in this invention. In one implementation of the invention, the normalized weight for a particular word ("tw₁") in a particular document is determined according to the formula:

$$tw_1 = \frac{tf_1 \left(\log \frac{N}{n_1} \right)}{\sqrt{\sum_{i=1}^a \left(tf_i \left(\log \frac{N}{n_i} \right) \right)^2}}$$

20

where:

N = total number of documents in the filing system which are selected for indexing (default selection is all user documents);

25

0970688E 00E704

5
10
15
20

n_i = df for the word "i" (df is document frequency for word "i" from the global document frequency table)

tf_i = term frequency for the "ith" non-stop word in the particular document

a = number of unique non-stop words in the particular document (i.e. the number of components in the vector for the particular document)

n_1 = the df for the particular word having its normalized weight being determined

tf_1 = tf (term frequency) of the particular word in the particular document; tw_1 is the normalized weight for this particular word in this particular document

Other implementations may be used according to the needs of the system designer.

2.5 Figure 16 shows a method for performing the operation of browsing of a pile. From node E, the system begins at step 751, in which the system examines the location of the cursor and determines whether or not it is pointing to a pile. It will be appreciated that the system will typically monitor the position of the cursor in any one of the many ways known to those in the art. If the cursor is not pointing to the pile processing loops back to node E. If the cursor is

09796992 082701

pointing to a pile then processing proceeds to step 752 in which the computer determines whether the cursor has continued to point to a document on the pile for a predetermined period of time. If not (e.g. cursor moved away from pile) processing loops back to node E in which the system continues to monitor the position of the cursor. If the cursor has pointed to a document on a pile for a predetermined period of time then processing proceeds to step 753 in which the system arranges the pile for viewing. This step may involve nothing or it may involve shifting a portion of the pile as shown in Figure 4e. Processing then proceeds to step 755 in which the view cone is shown and a proxy of the selected document in the pile is displayed. Then the system continues to monitor the position of the cursor to determine whether the user has pointed to another item in the pile (step 757). If this is true, processing loops back to step 753 to arrange the pile for viewing (e.g. pile is re-arranged to reflect a new shifted portion) and processing continues to step 755 in which the view cone continues to be displayed and a new proxy of the new selected document is also displayed in the view cone. If the same document is still being displayed because the pointer has not moved away from the document processing proceeds from step 757 to step 758 in which the computer determines whether or not the user has signalled to the computer to page through the selected proxy to the extent the proxy has multiple pages. If not, then processing goes to step 760; otherwise processing proceeds to step 759 in which the system shows the desired page of the multiple page proxy in the view cone on the display screen 22. Since the computer continues to monitor the position of the cursor and the status of the mouse's button, the computer system continues to determine whether the cursor is pointing to the pile (step 760); if not processing loops back to node E and otherwise processing loops back to step 755.

5
10
15
20
25

As noted above, the system provides various automatic or user invoked actions on piles in order to assist the user in organizing the documents stored stored in the user's filing system on the computer. The various ways in which the system of the present invention performs these functions will now be described with reference to Figure 17. Figure 17 shows three distinct processing flows from three nodes (G, G1 and G2). It will appreciated by those in the art that any of these nodes may be entered into on the basis of an interrupt provided to the system or some other user action so that entry may occur into the flow chart of Figure 17 at any node point, such as node point G, G1, or G2 depending on the action by the user. Processing begins from node G at step 801 which determines whether or not the user has given a document to the system to file in existing piles. This may occur by the user providing a "put away" instruction to the filing system after selecting a document to be put away. If this occurs, then processing proceeds to step 803 in which the system attempts to find a pile with the closest pile vector to the document's vector and places that document in that pile. This is typically done in such a manner that the similarity between the content of the document and content of each pile is determined. In a typical embodiment this similarity is determined by computing the unit dot product of the pile's vector and the document's vector. A system or user supplied predetermine threshold may then be used to compare against each dot product and the highest dot product above the threshold will indicate which pile to place the document into. Alternatively, the document (copies thereof) or aliases (pointers) thereof may be placed into all piles which produce dot products with the document's vector which exceed the preselected threshold. It should be noted that step 803 could be performed automatically by the system for any new or modified document if the user so instructs the computer system to perform this function automatically. That is, the user may

instruct the filing system on the computer to always be active in putting away documents into piles rather than doing this only when specifically invoked by the user (e.g. a "put away" command). An example of a method for determining a measure of similarity will now be provided. Consider a document vector and

5 a pile vector shown below:

	<u>Document Vector</u>		<u>Pile Vector</u>	
	<u>index</u>	<u>tw</u>	<u>index</u>	<u>tw</u>
	5	0.1	5	0.2
	27	0.2	16	0.17
10	503	0.15	257	0.17
	600	-	-	-
	-	-	-	-
	-	-	-	-

00798232-022701

Each index may be a pointer/index to an entry to the global document frequency table where the index "5" is for one word in the table and index "27" is for another word in that table. Note that in this example there is only one word (non-stop word) which is common between the document's vector and the pile's vector. The dot product, which is a way to measure the similarity between the content of the document and the content of the pile, for this example is determined by searching for matches in the indexes (e.g. index 5) and multiplying the tw's for each matched index. In this case, the dot product is: $0.1 \times 0.2 = 0.02$ since there is only one match. This dot product may then be compared to other dot products showing the similarity between the document and other piles.

2.5

Step 805 is performed any time after entry from node g1. In step 805, the system determines whether or not the user has given the system a sample

00756666.000704

the user whether the pile's script should be modified. This dialog box or window may be similar to the window 601 shown in Figure 14. If the user indicates that the pile's script is to be modified, processing proceeds to step 813 in which the system displays existing criteria and/or the script for the pile.

- 5 Alternatively, the script and criteria could be displayed in step 811 with a decision box requiring the user to decide whether or not to just add the document to the pile or modify the pile's script. At step 815, the system displays proposed script/criteria changes to the pile's criteria or script. This is shown in Figure 14 by the words found in the Richard Mander document which are
- 10 shown in the region 611 or by the items in region 605 such as the author of the document (from "Richard Mander"). In step 817, the user selects changes to the criteria and indicates to the computer that these changes are acceptable. This causes the system to respond in step 819 by pulling in copies or aliases of all documents on the system having vectors which match the new pile's script
- 15 vector. After step 819, the system proceeds back to monitoring any of the various actions/interrupts which cause entry into the processing shown in Figure 17. The use of a script has many possibilities, for example, it is noted that the script could include the use of the vector representation of the pile as well as other features which may be matched in a boolean "AND" fashion or
- 20 matched in a boolean "OR" fashion.

Figure 18a shows a method of the present invention for creating subpiles from a pile or other selected set of documents (e.g. documents within a folder). This method would be invoked in various ways by the user or perhaps under

- 25 system control automatically (e.g. as a result of a search where the user requests the search results pile to be processed into subpiles after the search). For example, the user could place a pile such as pile 550 into the visualization

007588888.002704

If the result of the test in step 859 is a "yes" then processing proceeds to step 863, in which the system sorts the matching subpiles (i.e. subpiles producing a similarity measure with the document which exceed the threshold) by similarity measurement to the current document. Then in step 865, the system determines whether a document is allowed to be in more than one subpile. If not, processing proceeds to step 869 in which the current document is added to best matching (on the basis of a similarity measure) subpile. Also in step 869, the system updates the running similarity score by adding the similarity score, which may be the dot product between the document's vector and the best matching subpile's vector, to the running total of the similarity score for the entire subpiling system, which is initialized at zero prior to beginning the process shown in Figure 18a. If a current document is allowed to be in more than one subpile, then processing proceeds to step 867 in which the current document is added to all matching subpiles having a similarity measurement which exceeded the threshold. As in step 869, the running similarity score for the system is updated by adding the dot products or other similarity scores (one for each matching subpile to which the current document was added) to the running total for the system. Following step 867 or 869, processing loops back to node F1 and the various steps described above. It is noted that if the document is to be added to more than one subpile, either a copy of the document or an aliases/pointer to a copy of the document may be stored in the subpile, in a manner which is similar to the conventional prior art technique of storing a file or an aliases/pointer of a file in a subdirectory or folder.

If at any step of the process of creating a subpile, there are no documents left in the original pile, then the decision required in step 853 results in the system proceeding to step 871 in which the system determines whether there

5
10
15
20
25

are any subpiles which contain only a single document. If no such subpile exists then processing proceeds to node F3 which results in step 876 being performed, otherwise, processing proceeds from step 871 to step 873 in which a miscellaneous subpile is created. Then, in step 874, each document from a single document subpile is transferred to the miscellaneous pile (and the running similarity score is updated) and the single document subpiles are removed from the system in step 875. Then, in step 876, the system determines whether any improvement to the total system similarity score has occurred since the last total similarity score. This system similarity score is computed as described above in steps 867 and 869 each time a document is added to a matching subpile and after all documents have been removed from the original pile this test in step 867 is typically performed; obviously, in the first pass through this step, the system will place all documents back in the original pile by proceeding to step 877 and then back to node F1, repeating the process while maintaining in storage a record of the prior total similarity score. It will be appreciated that this total similarity score is only for documents in the original pile and only for similarity measurements between those documents and subpiles created during the subpiling process described here. The test in step 876 may include a comparison to a threshold where the difference, which should be an improvement, between the prior total system similarity score and the current total system similarity score is compared to a threshold and if that difference does not exceed the threshold, then no effective improvement has occurred and processing is completed. A designer of a system using the present invention may decide that it is efficient and reasonably accurate to merely repeat the process of subpiling only once or some fixed number of times rather than attempting to access an improvement at the end of each subpiling pass.

5
10
15
20
25

Figure 18b shows an alternative subpiling process according to the present invention where the user chooses the number of subpiles which are to be created from the original pile. This is typically done in order to avoid unnecessary clutter of subpiles. Without this feature, it is possible that the computer filing system of the present invention could create too many subpiles making it more difficult for the user to manage and organize the piles created in this manner. The process of Figure 18b beings at step 880 in which the system determines the number of subpiles which are allowed by the user when a subpiling process is invoked. Processing then proceeds to step 882 to determine whether the desired number of subpiles have been created so far in the subpiling process. If they have, processing proceeds through node J2 to step 891, and otherwise processing proceeds to step 884 in which the system computes the total similarity of each remaining document in the original pile to all existing subpiles. In the special case of the first document which is removed from the original pile, there is nothing to compute since no subpile exists. Then, in step 886 the remaining documents are sorted from least to most total similarity. Then in step 888, the document in the original pile with the least total similarity is removed from the original pile. In the case of the first document which is removed from the original pile, any document will do. Then, in step 890 a new subpile is created with the document which was removed in step 888, using that document as the seed for the similarity measurement comparisons (e.g. the document's vector is used for the subpile's vector). Then processing proceeds back to step 882 in which this process repeats through steps 884 through 890 unless processing proceeds through node J2 to step 891. In the second pass through steps 884 through 890, step 884 does involve the computation of the total similarity of each remaining document in the original

5
10
15
20
25

pile to all existing subpiles. This is done in the fashion described above wherein the internal representation of each document is used with the internal representation of each subpile to compute a similarity measurement between the document and each subpile. The total similarity for a document to all existing subpiles is a summation of each similarity measurement between the document and all existing subpiles. Thus, for example, if four subpiles exist when the computation is made, four dot products may be computed as the similarity measurement, each dot product being between the document's vector and the seed vector of the four existing subpiles; these four dot products are then added together to provide a total similarity measurement for the document. Then in step 886, these total similarity measurements for each document are sorted. The document with the least total similarity in this sorted list is removed from the pile in step 888 and a new subpile is created with this document as a seed as shown in step 890 and the processing continues back to step 882. When the number of subpiles which exist by virtue of the subpiling process equals the desired number selected by the user, processing proceeds to step 891 which determines whether there are any documents left in the original pile. If the answer is no, the process is completed. However, if the answer from step 891 is yes, processing proceeds to step 893 in which the system takes the first remaining document from the original pile and labels it the current document and then proceeds to step 895. In step 895, the system compares the similarity of the current document to each existing subpile and then in step 897, the system sorts the matching subpiles by similarity to the current document. In step 899, the system adds the current document to the best matching subpile and proceeds back to step 891. The forgoing description of the process of subpiling from an original pile has been described with reference to a specific procedure; however, using known information retrieval technology, other implementations

will be available to those in the art upon reference to this description and the information retrieval art.

An alternative to the process of creating a subpile according to either of
5 the methods shown in Figure 18a or 18b will be described. Prior to starting the
subpiling process, the user may want to specify certain subpiles which must be
created; this may be done by specifying a "seed" document which is
representative of a desired subpile or a "virtual" seed document (specified by
10 keywords which may be used to create an internal/content representation of the
subpile). These desired subpiles are then used by the system as subpiles
which exist immediately upon beginning the process (e.g. entering step 853) of
creating subpiles from a pile. It should also be noted that the process of
creating subpiles may be performed from several original piles merely by
15 combining them into one original pile or by using one pile as an original pile
and the other original piles as pre-existing subpiles.

The system and method of the present invention also provide a
mechanism for searching documents in the file system of the computer whether
those documents are in piles or conventional subdirectories/folders. This
20 method is shown in Figure 19 and begins at step 901, in which the system
determines whether the user wants to search for documents matching a set of
terms which are specified by the user. Typically, the user will indicate that a
search is desired and the system will respond with a question asking the user
whether the user wishes to define a set of terms or to specify a sample
25 document. It will be appreciated that, other possible means of requesting a
search or formulating a search may be used. If the user wants to specify a set of
terms, the system allows in step 903 the user to do so and then creates in the

5
10
15
20
25

step 905 a virtual document vector (or other representation of the terms) for the set of terms (and their user selected weights, unless by default the terms are equally weighted). That is, the user specifies terms and then those terms are used to specify a vector of a virtual document which is used later when

5 processing. Processing from step 905 proceeds directly to step 911. If the user wishes to search using a sample document (e.g. create a pile from a sample document manually by searching and obtaining a list of the documents or automatically create a pile containing the search results) then processing proceeds from step 901 to step 907 in which the system requests the user to

10 specify the sample document; if no sample document is specified processing loops back to node C waiting for an interrupt or other instruction indicating to begin the search process. After the sample document is specified in step 907, the system proceeds to step 911 in which it compares each document's vector (or other representation of the document) in the system with the document

15 vector (or other representation) of the sample document or the document vector (or other representation) of the virtual document created in step 905. This may occur by any of a number of ways of determining a similarity measure, including by determining the unit dot product of the sample document's vector against the vector of all documents in the system. It will be appreciated that alternative

20 similarity measures or search routines may be employed, such as measuring the similarity between the document and each pile or folder and then searching within those piles or folders having the best matches to the document. It is noted that this comparison can eliminate documents which are not close enough by testing each dot product against a preselected threshold.

25

In step 915 the system sorts the list of documents based on the comparison with the document vector of the sample (or virtual) document. This

list may be limited by the comparison against a preselected threshold as described above. Then in step 917 the system creates and displays a pile of the sorted documents which are closest to the sample (or virtual) document. Following step 917 processing proceeds back node C which waits for a
5 command to search from the user. Alternatively, the pile may be divided into subpiles according to the techniques of the invention. The user may also, after the search pile is created, manipulate that pile to perform other searches such as a new or revised search.

10 Figure 20 shows a method for using color to assist in organizing information in a computer and to assist a user in understanding the content and organization of piles within the following system. This method begins in step 951 wherein the user selects a pile and then in step 953 selects a method of visualization such as the visualization window 551 shown in Figure 13a. This
15 causes the system, in step 955, and display the visualization window showing the pile within the window (or the user may have to move the pile into the window). Then in step 957, the user selects the command "order by date" and the system orders documents in the pile by the date of the document. It is appreciated that step 957 is a two-part process even though it is shown as one.
20 This may be done in any of the well known ways of sorting and listing documents by the date of the document. This date may be either the date maintained by the filing system of the document or may be a date line (field) of the document which is contained in the contents of the document and was provided by the author of the document. After step 957, the user in step 959
25 selects the command "color by date" and the system displays documents with different colors and organized according to the date. It is appreciated that step 959 is a two-part process even though it is shown as one. That is, the color of

00798882 000001

the document is correlated with the age of the document in the manner described above. This is done typically by determining the range of dates for the entire pile by determining the oldest document and the most recent document (and thereby determining the range) and thereby determining the
5 range of the variable saturation for a particular color (e.g. 256 different saturation values for a given luminance and hue of a color). The oldest document would then be assigned the most saturated color (e.g. saturation level equal to 255 in an 8 bit color system) and the most recent document would have the least saturation (e.g. saturation level equal to zero for the same
10 luminance and hue of a color). A document with a date in the middle of the date range would have half of the saturation of the oldest document and other documents would be proportioned accordingly. Other well known techniques may be used for assigning colors on the basis of the dates of the various documents in the pile. The user may next select "color by content" in step 961
15 in the manner described above in conjunction with Figure 13a. This causes the system in step 963 to display the documents in the pile with different saturations of color according to the closeness (determined by a similarity measure) of the document to the internal representation of the pile. This may be done by comparing the document's vector to the pile's vector by the using the unit dot
20 product of the two vectors in the manner described above. The computer would determine all dot products for all documents in the pile and then sort those dot products from highest to smallest. The highest dot product indicates a document which most closely matches the pile's internal representation (e.g. pile's vector) and the smallest dot product would indicate a document which is
25 farthest away from the pile's internal representation. This sorted list of dot products corresponds to a sorted list of documents since each dot product is associated with a particular document and thus different saturations of a given

05756292 002704
10220 2629265

color at the same luminance and hue may be supplied to the different documents according to their dot product. In other words, the dot product for a particular document maps to a particular saturation of a color the same way that the date maps to a color as described above. The user may desire that this pile
5 retain its colored form based on coloring by content by selecting the okay button 571 and then by closing the visualization tool window. This will cause the new colored pile to appear on the display means 22 in the filing system of the present invention. It is noted that the order of many of the steps shown in Figure 20 is not critical to the invention and these steps may be practiced in other
10 sequences which are not shown.

In order to summarize many of the features of the present, a typical example will be given of how a user may interact with piles. Figure 21 represents a flow chart showing this typical example; the order of many of these
15 steps is not critical to the invention and they may be practiced in other sequences which are not shown. Beginning in step 1001, the user creates a pile by moving one document on top of another. This has been described above in connection with Figure 3 and other figures. The user may also create a pile by selecting two different documents and selecting a command available
20 in the computer system to create a pile. Next the user in step 1003 moves the newly created pile by positioning the cursor over the base of the pile and signaling the selection of the pile and then moving the cursor to move the pile. In step 1005 the user selects another pile by positioning the cursor over the base of the pile and signalling a selection of that pile. In step 1007, the user
25 opens the pile by any one of the mechanisms for obtaining an alternate view as described above. For example, by rapidly depressing and releasing (double clicking) the mouse button while positioning the cursor over the base of the pile.

The system responds in step 1009 to the actions of the user in step 1007 by opening a pile window with the contents of the pile displayed in that window or alternatively by spreading out the documents in the form shown in Figure 8d.

An example of a pile window containing the contents of the pile is shown Figure
5 8a.

In step 1011, the user may close the pile window by positioning the cursor over the close box and signalling a selection or by using any one of the other known techniques for closing a window in a graphical user interface.

10 Then in step 1012, the user selects a document which is not in a pile and drags it to the top of an existing pile to add it to the pile. This is typically done in one of many ways for moving a document in a filing system having a graphical user interface, such as positioning the cursor over the icon of the document and selecting the document and keeping the document selected while the cursor is
15 moved thereby moving the cursor and the document on the display screen. The system's response to step 1012 is step 1015 in which the system displays a script dialog box since the targeted pile to which the document was added is a scripted pile and since the new document has an internal representation which does not match the current criteria of the pile. The user selects in step 1015 to
20 modify the pile's script. This takes place in step 1017 in which the system modifies the pile's script on the basis of the user's input. For example, the user specifies particular criteria to use in modifying the pile's script such as selecting two unique words from the added document or by using a particular label such as the label 614 shown in Figure 14.

25

Then in step 1019, the user instructs the system to add documents to the pile on the basis of the pile's modified script or the system does so automatically

after modifying the pile's script. Whether automatic or upon user instruction, the system adds to the pile on the basis of the new script in step 1019. The computer may display an animation of document icons moving from their old locations on the display screen into the pile.

5

In step 1021, the user instructs the system to create subpiles of a selected pile based on the contents of the documents. That is, the system is instructed to create subpiles on the basis of the contents (rather than for example the dates of the documents or other criteria upon which subpiling may be based). The system responds in step 1023 by examining the contents of each document in the selected pile and creating several subpiles from the original selected pile, each subpile containing similar content because of the operation of the present invention. The system as shown in step 1023 then assigns a name to each subpile on the basis of the words having the strongest weights for each subpile.

10
15
20
25

The user may then select, in step 1025, one of the subpiles by positioning the cursor over the base of the new pile and signalling a selection. The user then instructs the system to color the pile by date by selecting the command "color by date." This causes the system to respond in step 1027 by using the date of documents in the subpile to select a saturation of a color which is dependent on the date of the document. For example, the most saturated color represents the oldest document. The user in step 1029 then removes a document from this subpile by selecting the document and dragging it out of the pile. For example, the user may select the oldest document, based on the saturation color and remove it from the pile and examine the document. In step 1031, the user selects a document which is not in any pile and selects the

command "put away" which causes the system to put the document into the closest, in relevancy, pile or piles depending upon user preference.

Another example of a user's interaction with piles according to the present invention will now be described by referring to Figures 22a, 22b, 22c, 22d and 22e. As shown in these figures, the method and apparatus of the present invention may be applied to an object oriented programming environment where the use of piles is made for programming documents which are not the typical text/based office document, even though programming files contain text. As shown in Figure 22a, an object finder window 2201 includes three components which are the find box 2202, the results box 2204 and the discard box 2203. In this example, a search is specified by placing a sample document or a pile in the find box 2202 and the system will find objects that are similar to the objects in the find box 2202 provided they are not present in the discard box 2203. That is, the discard box 2203 acts as a "not" boolean logic search parameter. The results window or box 2204 shows the results of a search which have been automatically placed in three subpiles according to the present invention where the original search results were placed in a pile which was then divided into three subpiles 2210, 2211, and 2212.

Figure 22b shows a method and apparatus of the present invention where piles may be used to specify the search query rather than words or other query techniques. In this case, the find box 2202 contains the piles 2219 and 2220 which specify the search query (e.g. by using either internal representation of either pile as the search query). That is, the internal representation of pile 2219 is OR'ed together with the internal representation of the pile 2220 so that objects which are similar to either internal representation

05750002-002701

of either pile will show up in the results window 2204. As shown in Figure 22b, the discard box 2203 contains the pile 2218 which is used to discard objects which are found in the search which match the internal representation of the pile 2218. It is important to realize that the descriptions for the search query are "fuzzy" in that they are specified by the pile's internal representation rather than a specific list of key words which are connected by boolean logical operators. The results of the search are shown in the box 2204.

10 Figure 22c shows another use of the object finder 2201 where attributes of objects 2227, 2228, and 2229 are used to search the computer's filing system for documents and other objects which match the search criteria or have those attributes. These objects may be piles or other objects which may be treated as a text document and from which an internal representation of the document may be derived.

15 Figures 22d and 22e show a use of the invention in connection with media objects which represent digitized movies. Each digitized movie ("clip") have all been indexed by key words when they were created or when they were edited. The user has placed two clips (digitized movies) into the find box 2202 in order to find similar clips on the basis of the text/keywords for each clip. The user instructs the computer to find such clips on the basis of either clip 2231 or clip 2232 within the find box 2202. The system responds displaying four piles 2233, 2234, 2235 and 2236 which match the internal description (or exceed a threshold set for comparison purposes against a similarity measure). In this case, the four piles shown in the results window 2204 of Figure 22d were created by subpiling the original search results pile created from the find operation. The user may the browse one of the piles as shown in Figure 22e to

look for certain items which are desired. As shown in Figure 22e, the user has browsed the pile 2233 to find a particular digitized movie, the indicia or proxy 2238 of which is shown in the view cone 2237 as shown in Figure 22e. It will be appreciated that the proxy in the case of a movie may actually be an animated
5 dicon of the digitized movie in that various frames of the movie may be displayed rapidly in sequence in order to convey the impression the movie normally conveys when it is played. Similar techniques may be employed for other multimedia objects, such as digitized sound, etc.

10 The foregoing method and apparatus of the present invention may be conveniently implemented in a computer program in a computer system that is based upon the flow charts described above. No particular programming language has been indicated for carrying out the various procedures described because it is considered that the operations, steps and procedures described
15 above and illustrated in the accompanying drawings are sufficiently disclosed to permit one of ordinary skill in the art to practice the invention. Moreover, there are many computers and operating systems which may be used in practicing the invention and therefore no detailed computer program could be provided which would be applicable to these many different systems. Each user of a
20 particular computer will be aware of the languages and tools which are most useful for that user's needs and purposes.

While the invention has been specifically described with reference to the figures and with emphasis on certain computer systems, it should be
25 understood that the figures are for illustration only and should not be taken as limitations upon the invention. It is contemplated that many changes and modifications may be made by one of ordinary skill in the art to the apparatus