PROPERTY TREE FOR METADATA NAVIGATION AND ASSIGNMENT

FIELD OF THE INVENTION

[0001] The present invention relates generally to data storage in computer systems, and more particularly to methods and apparatus for organizing and locating data items by way of metadata properties.

BACKGROUND OF THE INVENTION

[0002] Today most computer users interact with the machine by way of a graphical user interface which, by exploiting the computer's visual display and intuitive input devices such as the mouse, mediates access to the operating system, application programs, and stored data. The standard approach to the design of the user interface has relied on a simple visual metaphor in which data files are contained in folders organized within a hierarchically-structured file system tree. The user interface provides controls that permit the user to navigate through the file system tree in order to locate and act upon data. Although such systems are easier to use than the command-line shell interfaces that preceded them, they generally have not provided a greater degree of abstraction from the underlying physical storage of data than was available in earlier systems.

[0003] The real-world familiarity of the file/folder model of storage contributed to its acceptance and popularity among computer users, but such user interfaces were also successful because users typically stored a relatively small number of data files on their machines. The ease with which computer systems can now be used, coupled with the availability of massive amounts of inexpensive disk storage, have to some degree made the standard data storage interface a victim of its own success. Computers are now being used to store large quantities of personal data in a variety of different formats for use with many different applications. It is not uncommon for a user to have hundreds or thousands of text documents, photographs, audio files, and other data records stored across multiple hard disks, shared networks and other storage media. Given such a development, drawbacks to the conventional folder tree storage model have become evident to computer users.

[0004] For the present-day user, the conventional storage model is undesirably and inflexibly one-dimensional. Items of data exist in one location and relate to other files in one way, by their relative position within the hierarchical folder tree, which ignores the many logical relationships files may have to one another. In previous versions of Microsoft® Windows®, files are associated with a limited number of properties over which the user has limited control, and those properties are accessible through a control that is difficult to locate and use. In other systems file metadata may be limited to the file name

and the user's informal knowledge of the file's type. Such limitations have made it difficult for users to organize their data in new ways based on the content and use of the data. As a result, the growth in the amount of data stored by users has made searching for data less efficient and more cumbersome.

[0005] Particular applications have offered domain-specific solutions to the problem of user data management. Two examples involve applications for storing and organizing digital photograph files. The Digital Image Library in Microsoft® Digital Image Suite 9.0 includes a "Keyword Painter" control that enables the user to easily organize and subsequently locate images stored in disparate locations throughout the file system. The user clicks on identifying keywords of the user's own choosing and then clicks on thumbnail images of pictures to which the selected keywords are to be assigned. The user can now filter the view of the library of pictures by particular keywords. Adobe Photoshop Album 2.0 permits the user to associate content-descriptive tags with pictures by drag-drop actions on thumbnail images. The user can then quickly search the collection of pictures by subject matter using one or more tags.

[0006] Another example is the Grand Central e-mail interface, a project of Microsoft Research. Hierarchical properties are used to categorize e-mail messages. An arbitrary number of categories can be assigned by way of the user interface. The categories facilitate searching for messages and the organization of messages into conversational threads.

[0007] Disclosed features of the Microsoft® Windows® Code-Named "Longhorn" platform point to a more comprehensive solution. A storage subsystem, WinFS, relies on an extensible scheme of metadata properties to enable the user to search for, organize and share data. The basic unit of data container is the item, which is associated with multiple properties set to specific values. Items are organized by properties most relevant to the user at a particular time. Powerful querying capabilities over the store of items are provided. Users can thus find data more quickly and can organize and operate upon data more efficiently.

[0008] Prior to the present invention, however, there has been no general mechanism for metadata assignment and metadata-based navigation comparable in ease and generality of use to such conventional storage system interfaces as the file system Explorer tree of previous versions of Microsoft® Windows®, which permit the user to navigate to a folder and to modify its contents in intuitive ways such as by dragging and dropping with the mouse.

SUMMARY OF THE INVENTION

[0009] The following represents a simplified summary of some embodiments of the invention in order to provide a basic understanding of the invention. This summary is not an extensive overview of the invention. It is not intended to identify key or critical elements of the invention or to delineate the scope of the invention. Its sole purpose is to present some embodiments of the invention in simplified form as a prelude to the more detailed description that is presented below.

[0010] The invention is intended to be practiced in the context of a computer system having a graphical user interface, or another kind of user interface, and a data storage subsystem in which data items are organized and queried by way of metadata properties. In accordance with one embodiment of the invention, a computer-implemented method of assigning values to properties is provided. The method includes displaying a property tree having nodes that expose properties of a set of data items. The method further includes, in response to user input that is indicative of an assignment of a property exposed by the property tree, modifying the value associated with the property in the data storage subsystem.

[0011] In another embodiment of the invention, a computer-implemented method of user navigation of data is provided. The method includes displaying a property tree, and, in response to user input indicative of a query of data storage, such as selection of a node that represents the query, displaying a browsable view of data items responsive to the query.

[0012] In yet another embodiment of the invention, a display device for user assignment of values to properties and user navigation of data is provided. The display device includes a property tree user interface control for displaying nodes that expose properties of a set of data items. The property tree user interface control is additionally for displaying nodes that represent a query of data storage.

[0013] In some aspects of the aforementioned embodiments, the property tree also provides nodes that represent folders within a hierarchical file system, in addition to nodes that represent metadata properties. In other aspects of these embodiments, the property tree has nodes that represent properties with inherent hierarchy, and nodes that represent user-defined hierarchical properties.

[0014] The invention may be implemented by way of software, hardware, or a combination thereof. The invention may be implemented, for example, within an operating system shell or within a running application program. It is contemplated that the invention may be embodied in one or more computer-readable media.

[0015] Other features of the invention will become apparent from the following detailed description when taken in conjunction with the drawings, in which:

BRIEF DESCRIPTION OF THE DRAWINGS

[0016] FIG. 1 is a schematic diagram showing an exemplary computer operating environment within which the present invention may be incorporated.

[0017] FIGS. 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18 and 19 are screenshots of a user interface for viewing and organizing stored music data, within which an embodiment of the present invention is incorporated.

DETAILED DESCRIPTION OF THE INVENTION

[0018] The present invention includes a tree-shaped set of user interface controls that exposes properties of data items, which will henceforth be referred to as the "ProperTree." The ProperTree provides two principal functions. First, the ProperTree enables the user to navigate across the virtual namespace of the user's data. Navigation is discussed further below. Second, the ProperTree permits the user to quickly and easily assign values to data item properties by adding or removing metadata with respect to property nodes exposed by the ProperTree. The user can add metadata to, and remove metadata from, many nodes at the same time. Because the ProperTree mimics the traditional folder tree control in certain respects, it enables users familiar with the conventional approach to storing, organizing and retrieving their data to work comfortably in a system with a radically different approach to storage that relies upon an extensible scheme of metadata properties and querying.

[0019] In one embodiment, the ProperTree control is displayed by default in a window within a pane to the left of a pane displaying a list view of items. When the user selects an item, the user can click on a property node exposed by the ProperTree to expand the node in place; the node can be collapsed if it is already expanded. A small widget, such as one displaying '+' or '-', may be used to indicate whether a node is collapsed or expanded. The expansion of a node shows the property values set for the currently selected item, as well as additional values that can be applied. In an embodiment, a most-frequently-used subset of values across a set of items is shown; the user can expand the list of values for the expanded property fully by, for example, clicking on a button. Having expanded a property node in the ProperTree, the user can change the property values; the changes are immediately applied to the item or items that have been selected by the user. The system may assign initial values to properties based on most-frequently used values.

[0020] Properties are displayed in the ProperTree in a manner that is appropriate to their type. Certain properties have only one value with respect to an item. A photograph, for example, might be either "Black/White" or "Color," with users being permitted to change this property. The user interface for changing the value may, in such a case, comprise a

clickable radio button adjacent to each possible value. Other properties may be capable of having multiple values with respect to an item. For example, a document may have multiple authors. In this case the user interface may comprise a checkbox adjacent to each possible value, with the user being permitted to check one or more boxes as desired. Richer properties are also supported appropriately. For example, a rich calendar control is shown for dates; for ratings, a five-star control may be shown. Properties that cannot be changed by the user, such as the date on which a document is written, do not have value selection controls in the tree, and the nodes for such properties can only be used for navigation. The system is extensible with respect to adding new controls appropriate to particular data types.

[0021] For any given property or property value in the ProperTree, the user can navigate to a view of storage that is pivoted to that property or value. Thus, navigation does not simply involve taking the user to a single location, unlike navigation using a conventional folder tree. Rather, navigation by way of the ProperTree causes all relevant data to be rearranged into a new structure through which the user can browse. For example, the user can use the ProperTree to query storage for all documents having a particular "Author" value. The list view then changes to show documents meeting that criterion. Alternatively, the user can use the ProperTree to display all document authors. This causes a stacked view of authors within which the user can then browse.

[0022] Turning now to the drawings, FIGS. 2 through 19 are screenshots of an exemplary user interface 201 for viewing and organizing stored music data, incorporated an embodiment of the present invention. On the right is a list view 205, and on the left is a ProperTree property tree 203. In an embodiment, similar interfaces are provided for documents, messages, video files, and contacts, with the ProperTree in each case being specifically adapted for the kind of data item that is presented. Such content-oriented interfaces may be provided with an operating system product as a component of the shell. Additionally, in an embodiment of the invention, the ProperTree is a control made available through the Common File Dialog in Microsoft® Windows® for application developers to reuse in their own applications. As noted above, the ProperTree 203 is displayed on the left side of the window 201 by default.

[0023] The ProperTree 203 includes a header 207, which in the depicted case is titled "Music Links". Below the header 207 are the nodes of the ProperTree 203. The nodes labeled Rating 217, Artist 221, and Genre 223, represent single properties. Certain other nodes exposed by the tree 203, All Songs 209, Online Store 211, and Recently Played 215, are constructs that represent a query that searches for particular items in the list view 205 that satisfy conditions of the query. The nodes iPod 213 and Playlists 219, which represent static lists, are similar. If the user selects All Songs 209, for example, all the songs in the

list view 205 are displayed. If the user selects Recently Played 215, however, only songs that have recently been played are shown.

[0024] Turning to FIG. 3, the effect of an initial user interaction with the ProperTree 203 is shown. "Expand" widgets 327, 329, 331, 333, 335 are displayed next to the expandable property nodes 213, 217, 219, 221, 223, respectively, when the user hovers in the area of the ProperTree with the mouse, as indicated by the arrow cursor 325. The property nodes 213, 217, 219, 221, 223 can be expanded by the user by clicking on the displayed '+' in the widgets 327, 329, 331, 333, 335.

[0025] Turning to FIG. 4, there is shown the state of the ProperTree 203 after the user hovers with the mouse over the text label of one of the property nodes, in this case Artists 221, as indicated by the hand-shaped cursor 437. If the user double-clicks on the text label of the property node, the list view 205 will be updated so that it is organized by the particular property that is selected, and the node will expand. In FIG. 5, there is shown the state of the ProperTree 203 when the user hovers with the mouse over the '+' of the expand widget 333 associated with the Artists node 221, as indicated by the mouse arrow cursor 539.

[0026] Turning to FIG. 6, there is shown the state of the ProperTree 203 after the Artists node 221 is expanded. The "Artists" text label is rendered in boldface, and the expand widget is replaced by a "collapse" widget 641. A list of five Artists values 643, 645, 647, 649, 651 is displayed, representing the most frequently used and most recently used values, below which is an overflow arrow button 653.

[0027] FIG. 7 shows the effect of the user clicking on the overflow arrow 653, as indicated by the mouse arrow cursor 755. A pop-up menu 757 is displayed, showing all the values for the Artists property as a flat list. Alternatively, as shown in FIG. 8, the complete list of values can be displayed in a pop-up menu 859 as a list of several subnodes 861, 863, 865, 867, 869, each subnode representing a range within the full list that can be expanded further by the user by clicking on one of the rightward-pointing arrow buttons 871, 873, 875, 877, 879. An alternative rendering of the range pop-up menu 881 is also shown.

[0028] Returning briefly to FIG. 6, it may be noted that a property value name 649 that is too long to display in full is truncated with an ellipsis or similar symbol. In FIG. 9, the effect of hovering over such a truncated value is shown, as indicated by the mouse hand-shaped cursor 983. A pop-up element 985 displays the full name of the value.

[0029] Turning to FIG. 10, there is shown the effect of selecting an item in the list view 205. Checkboxes 1087, 1089, 1091, 1093, 1095 are immediately displayed next to each value in the expanded property node 221. The checkboxes serve as controls by means of which metadata can be assigned to the selected item. In this case, the user has selected the

song item "Encounter" 1097. The checkbox 1089 is checked, indicating that the existing Artist value for that item is "50 Cent." The user can check a different box to change the value by assigning a different artist name to the song item. Similarly, checkboxes can be used in a list view of photographs to assign keywords to a selected photograph item. FIG. 11 shows the effect of the user clicking on the overflow arrow button 653 when an item in the list view 205 has been selected, as indicated by the mouse arrow cursor 1101. The pop-up menu 1103 is displayed, each value in the complete list now having a checkbox in front of it.

[0030] In general, a property node in the ProperTree 203 may have a context menu that is specific to that node. An example is shown in FIG. 12, where the context menu 1205 for the node Artists 221 is displayed. The top element in the menu 1205 is "Show All Music By Artist" 1207. The effect of selecting this task is identical to the navigation effect of clicking on the "Artists" label, as described above. "Add new Artist" 1209 provides an entry point for the user to add additional values to the list of values for this property node. "Rename" 1211 can be used to rename the property. In this case "Rename" is grayed out because Artists is a property that end users cannot rename. A user can devise new properties which can be made renameable. The "Move Up" 1213 and "Move Down" 1215 menu elements can be used to organize the order of property nodes in the ProperTree 203.

[0031] FIG. 13 shows the state of the ProperTree 203 after the user has selected "Add new Artist" 1209 in the context menu 1205 shown in FIG. 12. A type-in box 1317 is displayed, permitting the user to enter a new value for the Artists property. Once a new Artists value is entered, it can be assigned to a selected song item.

[0032] FIG. 14 illustrates the fact that the ProperTree 203 is not restricted to containing assignable property nodes, but may contain several kinds of navigable sets, pages or collections. Here the Playlists node 219 has been expanded. Playlists in the music data item context are one example of user-defined lists that function as data item containers, similar to traditional file system folders. As with traditional folders, the Playlists 219 may have an arbitrary number of nested components in a hierarchical structure, unlike property nodes that comprise only values. Here a most-frequently-used subset 1419, 1421, 1423 of the sublists of Playlists is displayed, below which is an overflow arrow button 1425. FIG. 15 displays the state of the ProperTree 203 after the user clicks the overflow button 1425 of FIG. 14. The full list hierarchy of Playlists 219 is displayed, comprising sublists 1421, 1527, 1529, 1531, 1533, 1423.

[0033] Turning to FIG. 16, there is shown the state of the ProperTree 203 after the user selects the label 1637 of the "Recently played" node 215, as indicated by the mouse arrow cursor 1635. As noted above, this node is not a single property, but represents a query. The

user can pivot the list view by selecting the glyph 1639 to the left of the node label 1637, as explained above. As shown in FIG. 16, selecting the node label 1637 displays a context menu 1641. Selecting the top element 1643 of the context menu 1641, "Show 'Recently Played'", has the same navigation effect as selecting the glyph 1639. Below this are two specific node tasks 1645, 1647 (not further specified here). The remainder of the context menu 1641 is similar to the context menu 1205 for the property node Artists 221 in FIG. 12. The user can select Delete 1649 to delete an item, Rename 1651 to rename it, and Move Up 1653 and Move Down 1655 to reorder it in the ProperTree 203.

[0034] Turning to FIG. 17, there is shown the state of the ProperTree 203 after the user clicks on the Music Links header 207, as indicated by the mouse arrow cursor 1759. A command menu 176 is displayed, permitting the user to perform certain tasks applicable to the entire tree control 203. By selecting the task "Add Current Page to Music Links" 1763, the user can define new nodes to add to the tree 203. By selecting "Sort Music Links by" 1765, the user can sort the nodes in a desired order. The user can hide the entire tree 203 from view by selecting "Hide Pane" 1767. By selecting the entry point 1769 at the bottom of the menu, "Add/Remove MusicLinks", the user causes a dialog to be brought up, allowing the user to select among different properties to add or remove properties to or from the ProperTree 203.

[0035] Similarly to the conventional folder tree control, the ProperTree 203 supports drag-and-drop semantics, but the behavior is different. An item in the list view 205 can be dragged and dropped onto a node in the ProperTree 203, but the item does not disappear from the list view; the behavior is more like tagging information to a file. A drag/drop action is shown in FIG. 18. The user clicks on an item in the list view, here the item 1871 corresponding to the song "Encounter," and drags it to the Artists node 221 in the ProperTree 203, as indicated by the mouse arrow cursor 1873 and the blurred selected item image 1875. In this case, Artists 221 is not itself an assignable property. Turning to FIG. 19, if the user hovers over the Artists node 221 for a sufficient length of time, the node 221 expands, permitting the user to drop the dragged item 1871 onto the name of a displayed artist. In this case the user is dropping the item 1871 onto the Artists value "Kruder & Dorfmeister" 649, and the checkbox 1093 associated with that value is highlighted. This is thus another mechanism by which a user can assign metadata to items.

[0036] In one embodiment of the invention, the ProperTree is integrated with a conventional folder tree control. In this embodiment, the tree has nodes that represent properties and expand to reveal values, as well as nodes that represent folders located in the hierarchical file system. In this embodiment, the advantages of both the hierarchical model

and the metadata model of storage are present. The tree may additionally include nodes representing other constructs, such as a link to a web page.

[0037] In an embodiment, the ProperTree supports "hierarchical properties," properties that inherently have hierarchy. For example, a date property may be represented in the tree as the hierarchy:

Year
    Month
        Day

In addition, in an embodiment, the ProperTree supports free-form hierarchical properties that are set by the user. For example:

Animal
    Dog
        Poodle
            Cat

Mineral
    Gold

Plant
    Palm Tree

[0038] FIG. 1 is a highly simplified schematic diagram showing an example of a suitable operating environment 100 in which the invention may be implemented. The features of such environments are well-known to those having skill in the art and need not be described at length here. The operating environment 100 is only one example of a suitable operating environment and is not intended to suggest any limitation as to the scope of use or functionality of the invention. Suitable computing environments for use with the invention include any computing device or computing system that supports interaction between user and machine.

[0039] With reference to FIG. 1, an exemplary system for implementing the invention includes a computing device, such as device 101. Device 101 typically includes at least one processing unit 103 and main memory unit 105, and at least one level of cache memory 107 connected to or situated within the processing unit 103 and serving as a buffer for the main memory 105. Device 101 has additional storage, including at least one magnetic hard disk 109 that serves as nonvolatile secondary storage and which is additionally used along with the main memory 105 in providing virtual memory. Device 101 may also have other

storage 111, such as optical disks, removable magnetic disks, magnetic tape, and other removable and nonremovable computer-readable media capable of nonvolatile storage of program modules and data and accessible by device 101. Any such storage media may be part of device 101. To facilitate user-machine interaction, device 101 has input devices 113, such as a keyboard 115 and a mouse 117 or other pointing device, and output devices 119, including a monitor or other display device 121. Device 101 also typically includes one or more communication connections 123 that allow the device to communicate data with other devices.

[0040] Programs, comprising sets of instructions and associated data for the device 101, are stored in the memory 105, from which they can be retrieved and executed by the processing unit 103. Among the programs and program modules stored in the memory 105 are those that comprise or are associated with an operating system 125 as well as application programs 127. The device 101 has one or more systems of logical data storage, such as a file system or alternative systems using database-related techniques, associated with the operating system 125. Such systems of logical data storage serve as interfaces that map logically-organized data to data physically located on secondary storage media, such as data stored in clusters or sectors on the hard disk 109.

[0041] Computing device 101 includes forms of computer-readable media. Computer-readable media include any available media that can be accessed by the computing device 101. Computer-readable media may comprise storage media and communication media. Storage media include volatile and nonvolatile, removable and non-removable media implemented in any method or technology for storage of information such as computer-readable instructions, object code, data structures, program modules, or other data. Communication media include any information delivery media and typically embody data in a modulated data signal such as a carrier wave or other transport mechanism.

[0042] All references cited herein are hereby incorporated by reference to the same extent as if each reference were individually and specifically indicated to be incorporated by reference and were set forth in its entirety herein.

[0043] The use of the terms "a," "an" and "the" and similar referents in the context of describing the invention, especially in the context of the following claims, is to be construed to cover both the singular and the plural, unless otherwise indicated herein or clearly contradicted by context. The terms "comprising," "having," "including," and "containing" are to be construed as open-ended terms (meaning "including, but not limited to,") unless otherwise noted. Recitation of ranges of values herein are merely intended to serve as a shorthand method of referring individually to each separate value falling within the range, unless otherwise indicated herein, and each separate value is incorporated into the

specification as if it were individually recited herein. The use of any and all examples or exemplary language herein (e.g., "such as") is intended merely to better illuminate the invention and does not pose a limitation on the scope of the invention unless otherwise claimed. No language in the specification should be construed as indicating any non-claimed element as essential to the practice of the invention.

[0044] Preferred embodiments of this invention are described herein, including the best mode known to the inventors for carrying out the invention. Variations on those preferred embodiments may become apparent to those having ordinary skill in the art upon reading the foregoing description. The inventors expect skilled artisans to employ such variations as appropriate, and the inventors intend for the invention to be practiced otherwise than as specifically described herein. Accordingly, this invention includes all modifications and equivalents of the subject matter recited in the claims appended hereto as permitted by applicable law. Moreover, any combination of the above-described elements in all possible variations thereof is encompassed by the invention unless otherwise indicated herein or otherwise clearly contradicted by context.

WHAT IS CLAIMED IS:

1. In a computer system having a graphical user interface and a data storage subsystem, a computer-implemented method of assigning values to properties, comprising:

first displaying a property tree, the property tree having a plurality of property nodes that can be expanded to reveal sub-nodes, each of the sub-nodes representing metadata properties of a set of data items, wherein the metadata properties are of a type consistent with the expanded one of the property nodes;

receiving an indication of a selection of one or more data items of the set of data items residing in the data storage subsystem, wherein the set of data items are organized and queried by way of the metadata properties;

receiving a user input indicating one of the revealed sub-nodes thereby defining a new value for the selected one or more data items;

in response to the user input, modifying the metadata properties of the selected one or more data items associated with the indicated one of the sub-nodes to be the new value as stored in the data storage subsystem; and

second displaying the property tree such that the new value is associated with the indicated one of the sub-nodes, wherein displaying the property tree comprises displaying one of the sub-nodes with an adjacent button for indicating a new value.

2. The method of claim 1, wherein the user input is indicative of adding metadata to at least one property node in the property tree.

3. The method of claim 1, wherein the user input is indicative of removing metadata from the at least one property node in the property tree.

4. The method of claim 1, wherein first displaying the property tree comprises displaying the property tree in a pane in a window, the window having a second pane displaying a list view of data items within the set of data items.

5. The method of claim 1, wherein the user input includes expanding the at least one property node to reveal the sub-nodes.

6. The method of claim 1, wherein displaying the property tree further comprises displaying a most frequently used subset of sub-nodes associated with one of the plurality of property nodes.

7. The method of claim 6, wherein the user input indicates an expansion the one of the plurality of property nodes, and wherein second displaying the property tree further comprises, displaying a full set of sub-nodes of a type associated with the expanded one of the plurality of property nodes.

8. The method of claim 1, wherein displaying the property tree further comprises displaying the adjacent button as a radio button for indicating a new value.

9. The method of claim 8, wherein the user input includes clicking the radio button.

10. The method of claim 1, wherein displaying the property tree further comprises displaying a sub-node associated with a multiple-valued property node of the plurality of property nodes with an adjacent checkbox.

11. The method of claim 10, wherein the user input includes checking the checkbox.

12. The method of claim 1, wherein first displaying the property tree further comprises displaying a sub-node associated with a property node of the plurality of property nodes by way of a user-defined control.

13. The method of claim 1, wherein the user input includes dragging and dropping the selected one or more data items of the set of data items onto a sub-node in the property tree.

14. The method of claim 1, wherein first displaying the property tree further comprises displaying a tree that has nodes that represent folders within a hierarchical file system.

15. The method of claim 14, wherein first displaying the property tree further comprises displaying a tree that has a property node of the plurality of property nodes that represents a link to a web page.

16. The method of claim 1, wherein first displaying the property tree further comprises displaying a tree that has a property node of the plurality of property nodes that represents a type of property that has inherent hierarchy.

17. The method of claim 1, wherein first displaying the property tree further comprises displaying a tree that has a property node of the plurality of property nodes that represents a user-defined hierarchical property.

18. The method of claim 1, wherein the set of data items residing in the data storage subsystem includes data items that comprise one or more of text data, audio data, video data, and image data.

19. The method of claim 1, wherein the method is performed within an operating system shell.

20. The method of claim 1, wherein the method is performed within a running application program.

21. In a computer system having a graphical user interface and a data storage subsystem in which data items are organized and queried by way of metadata properties, a computer-implemented method of user navigation of data, comprising:

displaying a property tree, the property tree having a plurality of nodes exposing properties of the data items, wherein displaying the property tree comprises displaying one of the plurality of nodes with an adjacent button for exposing the properties of the data items;

simultaneously displaying a list view, wherein the list view is a set of the data items organized by a particular property;

receiving a user input that queries the data storage subsystem with a criterion;

in response to the user input, displaying in the list view the set of data items organized by the particular property meeting the criterion;

receiving an indication of a selection of one or more data items of the set of data items displayed in the list view;

receiving a second user input indicating one of the plurality of nodes; and

applying the exposed properties associated with the indicated one of the plurality of nodes to the particular property of the selected one or more data items such the organization of the set of data items in response to the criterion is modified.

22. The method of claim 21, wherein the user input includes selection of a node in the property tree that represents the query.

23. The method of claim 21, wherein displaying the property tree includes displaying at least one node that represents a navigable set containing data.

24. The method of claim 23, wherein the at least one node that represents a navigable set has at least one nested component.

25. The method of claim 21, wherein the user input includes expanding at least one node in the property tree.

26. The method of claim 21, wherein the data items include items that comprise one or more of text data, audio data, video data, and image data.

27. The method of claim 21, wherein the method is performed within an operating system shell.

28. The method of claim 21, wherein the method is performed within a running application program.

29. A computer-readable medium having computer-executable instructions for performing a method for user assignment of values to metadata properties of stored data items, the method comprising:

first displaying a property tree, the property tree having a plurality of property nodes that can be expanded to reveal sub-nodes, each of the sub-nodes representing metadata properties of a set of data items, wherein the metadata properties are of a type consistent with the expanded one of the property nodes;

receiving an indication of a selection of one or more data items of the set of data items residing in the data storage subsystem, wherein the set of data items are organized and queried by way of the metadata properties;

receiving a user input indicating one of the revealed sub-nodes thereby defining a new value for the selected one or more data items;

in response to the user input, modifying the metadata properties of the selected one or more data items associated with the indicated one of the sub-nodes to be the new value as stored in the data storage subsystem; and

second displaying the property tree such that the new value is associated with the indicated one of the sub-nodes, wherein displaying the property tree comprises displaying one of the sub-nodes with an adjacent button for indicating a new value.

30. A computer-readable storage medium having computer-executable instructions for performing a method for user navigation of data in a data storage subsystem in which data items are organized and queried by way of metadata properties, the method comprising:

displaying a property tree, the property tree having a plurality of nodes exposing properties of the data items,wherein displaying the property tree comprises displaying one of the plurality of nodes with an adjacent button for exposing the properties of the data items;

simultaneously displaying a list view, wherein the list view is a set of the data items organized by a particular property;

receiving a user input that queries the data storage subsystem with a criterion;

in response to the user input, displaying in the list view the set of data items organized by the particular property meeting the criterion;

receiving an indication of a selection of one or more data items of the set of

data items displayed in the list view;

receiving a second user input indicating one of the plurality of nodes; and

applying the exposed properties associated with the indicated one of the plurality of nodes to the particular property of the selected one or more data items such the organization of the set of data items in response to the criterion is modified.