The present invention relates to a nesting technique for presenting data in a nesting navigator. The nesting navigator allows a user to navigate seamlessly between different views/aggregations of data using a uniform interface. One embodiment of the nesting navigator is a universal period selector that allows a user to select a point in time, a period in time, or a range using one or more nest control viewers tailored to such purpose.

Abstract

Methods, systems, and techniques for presenting data using a nesting navigator are provided. Example embodiments provide a nesting navigator that allows a user to navigate seamlessly between different views/aggregations of data using a uniform interface. In one embodiment, the nesting navigator is a universal period selector that allows a user to select a point in time, a period in time, or a range using one or more nest control viewers tailored to such purpose. This abstract is provided to comply with rules requiring an abstract, and it is submitted with the intention that it will not be used to interpret or limit the scope or meaning of the claims.
Fig. 1

Nesting

Generations of the nesting

1 2 3

Cousin objects

Sibling objects

Intermediate generation of the nest

Bottom generation of the nest

Generators of the nest

100

120

121

122

Gen 1 2 3
nest

Gen 3 4 5
nest-set

Gen 5 6 7
nest-set

Intermediate nests not drawn
Example: Timeline Nesting

- Year
- Quarter
- Month
- Week
- Day
- Shift
- Hour

Fig. 2
Fig. 5h

Fig. 5i

Fig. 5j
Nest navigation routine

1. Aggregate data objects into multiple nests

2. Display a nest viewer control of one of the multiple nests (e.g., a month-week-day viewer)

3. Receive user selection

4. Display indication that data object as selected

5. Display a nest viewer control of a sub-nest, with the selected data object as the selected generator object of the sub-nest (e.g., a day-shift-hour viewer)

6. 2nd selection of bottom level data object

7. Display a nest viewer control of a supra-nest, with the selected data object as a selected bottom-level data object of the supra-nest (e.g., a year-quarter-month viewer)

8. More selections?

9. Yes

10. Optionally perform other operations

11. End

12. No

Fig. 7
NESTING NAVIGATOR USER INTERFACE CONTROL

TECHNICAL FIELD

[0001] The present disclosure relates to methods, systems, and techniques for presenting and managing data and, in particular, techniques for the viewing of a hierarchically ordered set of data using a user interface control adapted to same.

BACKGROUND

[0002] Oft times applications require the ability to display data that may be part of a larger hierarchical organization of the data. For example, data regarding a family tree natural falls into a hierarchy of children, parents of those children, parents of the parents, etc. User interface controls for examining the data at any particular point in the hierarchy are typically specific to the application. For example, a user interface control for navigating a family tree is typically different that a user interface control for browsing parts of a book (see the browse capabilities of websites such as www.amazon.com for examples).

[0003] In addition to these difficulties, users are often confused when using such tools as it is easy to get lost within the hierarchy of data. This is especially true when a particular level of data within the hierarchy is no different than any other level of data. For example, in the family tree example, each object is a person. When viewing the data at one level the user may not at all be sure where the data is with respect to other data within the hierarchy. Different techniques for addressing these issues have been used. For example, additional annotations identifying the data and navigational visual cues such as expand (+) and contract (-) symbols with lines showing the various levels within the hierarchy have been used. As the data becomes more complex, even these visual cues can be lacking, resulting in user confusion.

BRIEF DESCRIPTION OF THE DRAWINGS

[0004] The patent or application file contains at least one drawing executed in color. Copies of this patent or patent application publication with color drawing(s) will be provided by the Office upon request and payment of the necessary fee.

[0005] FIG. 1 is an example diagram that illustrates an example nesting.

[0006] FIG. 2 is an example diagram illustrating an example timeline nesting.

[0007] FIG. 3 is an example of a time period nesting navigator.

[0008] FIG. 4 is an example diagram of an example nest viewer control with an interface to navigate to other nest viewer controls.

[0009] FIGS. 5a-5j illustrate an example Universal Period Selector as an example embodiment of a nesting navigator.

[0010] FIG. 6 is an example block diagram of an example computing system that may be used to practice embodiments of an example nesting navigator.

[0011] FIG. 7 is an example flow diagram of an example routine to navigate between multiple nests of an example nesting.

DETAILED DESCRIPTION

[0012] Embodiments described herein provide enhanced computer- and network-based methods, techniques, and systems for navigating between objects in a hierarchically organized set of objects. Example embodiments provide a concept of a “nesting navigator,” which enables users to move seamlessly between objects of different levels of the hierarchically organized set of objects to view and manipulate them. The nesting navigator presents levels of data grouped into nests (sets of nests) of different types, each of which can be viewed using its own type of nest viewer control (also referred to as a nest viewer). Each nest of data contains a designated (but consistent from nest to nest within the same nest-set) number of levels of data. In at least one embodiment, the number of levels of data grouped into a nest is three, so that each nest contains a top level that has single generator object (e.g., an item), an intermediate level of groups of items, and a bottom level of items (for each group). Other embodiments that contain a different number of designated levels may specify different sub-groupings in one or more intermediate levels. Although, the embodiments described herein generally use the term “nesting” to refer to the entire hierarchy and the term nest to refer to a defined portion or the nesting, it is to be understood that each nest is also a sub-nesting, and that a nesting may be instead recursively defined as containing one or more nestings.

[0013] The system is premised on the notion that navigation from one view of the objects to another consists of changing the view of a single generator object to show it as an item (or nested view) of the next level in the hierarchy, or changing the view of an item (or nested view) to show it instead as a single generator object of the next lower level in the hierarchy. Thus, to navigate down (i.e., deeper within) a hierarchically organized set of objects, the user needs only learn how to select an object using a nest viewer, and once an object is selected, to drill down, which will present the next applicable nest viewer with the selected object as the nest generator. Similarly, to navigate up a hierarchically organized set of objects, the user needs only to learn how to select the generator object (for example, the object that appears to be the root, title, or initiator of the data currently being displayed), and once the generator object is selected, to jump-up, which will present the previous applicable nest viewer with the selected generator object as a (selected) object of another nest. Accordingly, the user need only learn a single paradigm for navigating between different views (e.g., aggregations) of the same data.

[0014] More specifically, the nesting navigator is a type of user interface control that assists a user in navigating seamlessly among a hierarchically ordered set of objects, regardless of the actual object type. For example, in some embodiments, a single nesting navigator (potentially having one or more nest viewer controls) may be used to navigate data that are amenable to being aggregated into different groupings of time such as hours, shifts, days, weeks, or even months. The nesting navigator, through its one or more nest viewer controls, may also assist the user in selecting one or more objects from the set.

[0015] A nesting navigator defines an abstraction referred to as a “nesting” which it uses to organize and navigate among
objects. Nestings may contain one or more other “nests” (or sub-nestings) as described in more detail below. FIG. 1 is an example diagram of a nesting. A nesting 100 is an organized collection of objects. In at least some embodiments each object has a one-to-many relationship with its child objects; each child object has one parent. Within a nesting, the objects are organized into hierarchical levels referred to as “generations” (here labeled generations 1 through 7). An object in any specific generation must be related to one and only one object (referred to as its parent) in the generation immediately above it. An object in any specific generation may also be related to one or more objects (referred to as its children) in the generation immediately below it. The relationship can be one of ancestry, containment, categorization, or some other type of relationship. Objects sharing the same parent are referred to as siblings, and objects within the same generation are referred to as cousins. Note that in at least one embodiment of the nesting abstraction, siblings are also cousins.

[0016] A nest, such as example nest 102, is a substructure within a nesting and may cover one or more generations of the nesting. A nest can also be viewed as a sub-nesting of the nesting. The top object of a nest is referred to as its “generator,” or generator object. The generator of a nest may be any object in its associated nesting. A nest includes all of the objects which are descendents (e.g., children, grandchildren etc.) of its generator down to a designated level of depth. This level of depth is referred to as the depth of the nest. Within a nesting, specifying the generator object and the depth of a nest is necessary and sufficient to define a nest. The lowest generation within a nest is referred to as its “bottom generation.” The generations in between are referred to as “intermediate generations.” For example, a 3-level nest embodiment would include a generator, and the generator’s children and grandchildren. In such a 3-level embodiment, grandchildren would constitute the bottom level, and children would constitute the intermediate generation. A 4-level nest embodiment would also include the generator’s great-grandchildren, and the great-grandchildren would constitute the bottom level of the nest, with children and grandchildren being intermediate generations of the nest. Thus referring to the example of FIG. 1, which represents a 3-level nest scheme, nest 102 contains a single parent generator, 3 intermediate children, and 11 bottom generation grandchildren.

[0017] FIG. 1 shows an example of three hierarchically arranged overlapping sets of nests 120, 121, and 122. Nest-set 120 contains a single nest 102; nest-set 121 contains at least nests 105 and 106; and nest-set 122 contains at least nests 110 and 111. Each of the nests 102, 105, 106, 110, and 111 contain 3 generations of objects, of which at least one generation of objects is shared with another higher or lower nest in the hierarchy. For example, a given nest, such as nest 106, may have a supra-nest which is the higher order nest that contains the generator of the given nest, such as nest 102. A given nest, such as nest 106, may also have one or more sub-nests, such as nest 111, the sub-nests each having a bottom level object of the given nest (e.g., nest 106) as the sub-nest’s (e.g., nest 111) generator.

[0018] A nest-set, such as nest-set 121, includes all of the nests that have the same depth and whose generators are cousins in the nesting, here nests 105 and 106. The nest-sets overlap in a way that allows a bottom level object contained in one nest-set to be a generator of a nest in the subordinate nest-set (a sub-nest). This concept of overlapping nest-sets contained within a nesting is fundamental to the workings of the nesting navigator. This overlap provides a method of navigating from a nest to its supra-nest or to one of its sub-nests.

[0019] As another way of viewing the nesting abstraction, each object in a nest may itself be another nest. Thus, nesting can be viewed as a recursive construct, where a sub-nesting is an “order-n” of the nesting.

[0020] FIG. 2 is an example diagram illustrating an example timeline nesting. In FIG. 2, nesting 200 represents a timeline having objects that are aggregated into hours, shifts, days, weeks, months, and quarters, in a single year. Nesting 200 contains a first order nest 201 (a year-quarter-month nest); at least two second order nests 205 and 206 (month-week-day nests); and at least two third order nests 210 and 211 (day-shift-hour nests). Note that at least one generation of objects is shared between the various levels. For example, month objects are shared between the first order nest 201 and the second order nests 205 and 206.

[0021] The timeline represented by (e.g., mapped into) nests shown in FIG. 2 is an example of one possible type of nesting. Other types of nestings can be similarly incorporated, such as a nesting relating to books that would include levels for the book, sections, chapters, etc. As another example, a nesting can be defined relating to linear space whose levels could be mile, quarter mile, furlong (¼ mile), yard, foot, inch, etc. Accordingly, the nesting and nesting navigator concepts can be applied to any hierarchically organized set of objects where objects in any layer have a one-to-many relationship with objects in the next lower layer.

[0022] Various users may use a timeline nesting 200 of FIG. 2 to select and view various periods of time, such as by selecting different aggregations of the objects in the timeline nesting. For example, an accountant may use the timeline nesting 200 to select a particular quarter of a given year, such as to review company sales for that quarter. An employee may use the timeline nesting 200 to select a particular week, such as to schedule a vacation, or to select a particular hour of a particular day to schedule an appointment or use of a resource.

[0023] Using the timeline 200 to select a specific period is a matter of providing an user interface allowing the user to navigate the timeline nesting to view an appropriate nest and allowing the user to pick an appropriate object or objects from that nest. A nesting navigator may be provided to perform at least these two tasks. For example, a nesting navigator for navigating the timelines shown in FIG. 2, allows a user to view either a year-quarter-month, month-week-day, or day-shift-hour nest, and then to select one or more of the objects in the nest. If the nesting were instead a 4-order nesting, the nesting navigator would be adjusted accordingly.

[0024] FIG. 3 is an example of a time period nesting navigator. This nesting navigator may be used to navigate the timeline represented by the example timeline nesting shown in FIG. 2. The illustrated time period nesting navigator 300 includes several different nest viewer controls 303-305, which are used to navigate to and select objects within each of the different types (orders) of nests of the timeline nesting 200. In particular, the year-quarter-month viewer control 303 is used for navigation of the year-quarter-month nest 201; the month-week-day viewer control 304 is used for navigation of the month-week-day nests, including nests 205 and 206; and the day-shift-hour viewer control 305 is used for navigation of the day-shift-hour nests, including nests 210 and 211. In addition, in the example of FIG. 3, the time period nesting
navigator 300 includes a button 301 for entering multiple selection mode (where multiple objects may be selected) and a button 306 for confirming a selection. A title 302 for the current period that has been selected is also displayed.

To note, the example nesting navigator illustrated in FIG. 3 demonstrates the concurrent availability of the various nest viewer controls 303-305, arranged vertically. In other embodiments, the various nest viewer controls are available individually, for example, such that they appear stacked in z-order (along a z-axis plane of windows). In these embodiments, a button, or other user interface control may be provided to navigate (move between) the various controls. An example is shown in FIG. 4. In yet other embodiments, the nest viewer controls are arranged in other manners such as across the screen horizontally. Other arrangements are also possible.

FIG. 4 is an example diagram of an example nest viewer control. In particular, nest viewer control 401 represents the month-week-day viewer 304 shown in FIG. 3. In FIG. 4, nest viewer control 401 comprises a title “December 2007” 402, which indicates the generator object (the month December in the year 2007) for the currently viewed nest; indicators to the intermediate level objects, which are weeks (here labeled W1, W2, W3, and W4); and indicators to the bottom level objects which are the days within each week (here the numbered days of each week of the month). The green lines shown throughout the diagram represent an availability flag, signifying that a user can select the corresponding object. When a user selects the title 402 to select the generator object of the nest, details such as those shown in detailed diagram 410 are presented. In particular, the nesting navigator adds a selected flag, shown here as an orange colored line, a label that corresponds to the object (which here is the label for the entire nest), and a button (the jump-up flag) for navigating to the next higher level nest viewer control. In this example, when the jump-up flag is selected, the nesting navigator will present (when available) the next higher level nest viewer control, for example the year-quarter-month viewer control 303 in FIG. 3.

Each of the intermediate level objects, such as Week 3, is represented by a button (or other similar user interface control) that is available to be selected as a current week, by for example, selecting the button 406 labeled “W3” with an input device. Each of the bottom level objects, such as Day 15, is represented by a button (or other similar user interface control) that is available to be selected as a current day, by for example, selecting the button 403 labeled “15” with an input device. When the day button, for example button 403, is selected, the nesting navigator presents information as shown in the detail diagram 420. In particular, the nesting navigator adds a selected flag, shown here as an orange colored line, a label that corresponds to the object (here the number “15” for Day 15 of the selected week and month), and a button (the drill-down flag) for navigating to the next lower level nest viewer control. In this example, when the drill-down flag is selected, the nesting navigator will present (when available) the next lower level nest viewer control, for example, the day-shift-hour viewer control 305 in FIG. 3.

The nest viewer control 401 of FIG. 4 also shows a selection-stepper control 404, which conveniently allows a user to progress to the next or previous selection in the current nest-set (e.g., siblings or cousins) being presented and a confirm-selection button 405, which “accepts” the selection for whatever purpose the software incorporating the nesting desires.

Some embodiments of the nesting navigator include a toggle control that allows a user to place the navigator into either a selection mode or a navigation mode. This may be useful, for example, in situations where the selection process triggers a data fetch that takes significant amount of time (e.g., a few seconds or more). If the user wanted to navigate to another selection that was some distance away, the user could put the nesting navigator into navigation mode so that intermediate selections that are made only for navigation purposes do not result in data actually being fetched. Then, when the user has reached the desired item, the nesting navigator can be placed back into selection mode and any ensuing data fetching can then take place.

The other nest viewer controls can be constructed similarly to that described with respect to FIGS. 3 and 4. For example, the illustrated embodiment of the year-quarter-month viewer control 303 comprises a title “2007”, which may indicate the generator object for the currently viewed nest; indicators to the intermediate level objects, which are quarters (here labeled Q1, Q2, Q3, and Q4); and indicators to the bottom level objects which are individual months of the year (indicated here by month abbreviations). The illustrated embodiment of the year-quarter-month viewer control 303 also includes availability flags, selected flags, and a selection stepper, and may include jump-up and drill-down flags (not shown), such as similar to those discussed elsewhere.

Also for example, the illustrated embodiment of the day-shift-hour viewer control 305 comprises a title “Dec. 6, 2007”, which indicates the generator object for the currently viewed nest; indicators to the intermediate level objects, which are shifts (here labeled S1, S2, and S3); and indicators to the bottom level objects which are individual hours (1-12). In this illustrated embodiment, the P.M. hours have green availability flags indicated in the outside ring, and the A.M. hours have green availability rings indicated in the inside ring. The light and dark shadings within the ring represent daytime and nighttime hours. The three shift controls S1-S3 are used for selecting specific shifts. Notice in this illustrated example that the A.M./P.M. indicator appears in the center of the clock. The illustrated day-shift-hour viewer control 305 also includes various other availability flags (e.g., corresponding to the shifts, etc.), selected flags, and a selection stepper, and may also include jump-up and drill-down flags (not shown), such as discussed elsewhere.

The example nest viewer controls described in FIGS. 3 and 4 may be presented in various ways in various embodiments. For example, in some embodiments, the nest viewer controls may present additional and/or other information, and/or may present information in other manners than those described. For example, in some embodiments, the availability flags and selected flags may be indicated in other manners using other techniques (e.g., highlights, lowlights, 3-dimensional indicators, etc.). In addition, in some embodiments, the drill-down and jump-up flags may be indicated in various other manners. In addition, in some embodiments, the various indicators, flags, and controls may be provided by and/or accompanied by a corresponding audio queue.

In general, a nesting navigator may have one or more of the following attributes:

- Use common visual metaphors, for example a clock face, or a calendar page for picking periods of time. Users already understand how these devices are structured.

- Clearly show what items are available for selection, such as discussed elsewhere. As one example, assume for instance that one is looking at a month-week-day nest. Further assume, in this example, that it may not be possible to schedule appointments on a specific day of the week, say Sunday. So Sundays should somehow
be visually marked as unavailable, while other days are visually marked as available.

[0036] Allow the user to select one or more objects from the nesting.

[0037] When a selection of an object or objects is made, the selected object or objects are visually marked as such.

[0038] Give the user an intuitive and quick way of navigating from a nest to one of its cousin nests, and from a nest to its supranest or to one of its sub-nests. For example, a user looking at a month-week-day nest might want to select a day and then immediately view that day’s sub-nest (a day-hour nest) to select a specific hour for an appointment.

[0039] Appendix A, incorporated herein by reference in its entirety, is an example design specification for nesting navigators. It includes example behaviors, data structure considerations, events, and methods for implementing a nesting navigator.

[0040] In one example embodiment, a specific type of nesting navigator, referred to as Universal Period Selector is provided. The Universal Period Selector (“UPS”) is a nesting navigator that may be used to select a specific range, point, and/or period in time. FIGS. 5a-5j illustrate an example Universal Period Selector as an example embodiment of a nesting navigator. This selector provides a year-quarter-month viewer control, a month-week-day viewer control, and a dayshift-hour viewer control, such as those described with respect to FIG. 3 and discussed below. In general, using the UPS of FIGS. 5c-5j, a period in time can be selected directly, instead of only by a start and end date or time. In addition, common periods can be selected in a single action (e.g., a single mouse click), without requiring a drag operation. Accordingly, user actions can be minimized and are simplified so as to avoid complex multiple key/selection (augmented selection like CTRL-click) types of interactions.

Appendix B, incorporated herein by reference in its entirety, also illustrates an example UPS.

[0041] In the illustrated example of FIG. 5a-5j, a user is interacting with an illustrated embodiment of a UPS, such as to select one or more specific ranges, points, and/or periods in time. In some embodiments, a user may use a UPS to select various ranges, points, and/or periods of time, such as to cause statistics and/or other types of data or tools (e.g., performance statistics, sales records, schedulers, etc.) corresponding to the selected various times to be displayed to the user (although such a display is not illustrated here).

[0042] FIG. 5g illustrates an example month-week-day viewer control 500, showing the month of December 2007, which may be displayed as part of a data management/viewing application or tool. Notice that all of the days are available for selection as indicated by the green availability flags in the day cells, as is the entire month, also indicated by a green availability flag. With a single movement (e.g., a single mouse click), the user can select any day in December, any one of the four weeks in December, or the entire month. In some embodiments, if any of the days presented in the month-week-day viewer control 500 are unavailable (e.g., if no data corresponds to those days), there would be no green availability flags indicated corresponding to those days. In this illustrated example, only full weeks are displayed, with the year being organized such that each month has either 4 or 5 full weeks (although other configurations are possible). Thus, in this example embodiment, December 1 is treated as part of November, likewise December 30 and 31 are treated as part of January. Other arrangements are possible.

[0043] In FIG. 5b, which continues the example of FIG. 5a, the user has selected calendar cell 501 corresponding to December 6 (e.g., by a mouse click). The selection is indicated by the orange selected flag that appears in calendar cell 501. In addition, as a result of the selection, a drill-down flag, indicated by the downward pointing green arrow, is displayed in calendar cell 501 to indicate that, if the user clicks the calendar cell a second time (or, for example, double clicks the first time), a day-shift-hour viewer control appears. The user can then select a specific shift or hour on this day (e.g., such as illustrated in FIGS. 5h-5i, discussed below).

[0044] In FIG. 5c, which continues the example of FIG. 5b, the user has selected week button 502 (W1) (e.g., by a mouse click) to select the entire first week of the month (here, December 2007). The selection is indicated by the orange selected flag that appears under each day of the selected week and appears adjacent to W1. Note, in this example, the green availability flags still appear under the days of the selected week, indicating that these days are still individually selectable by the user.

[0045] In FIG. 5d, which continues the example of FIG. 5c, the user has selected month button 503 (e.g., by a mouse click) to select a period of time corresponding to the entire month of December 2007. Notice that all the days and weeks in the month are illustrated as selected by the orange selected flags. In addition, as a result of the selection, a jump-up flag has been displayed (upward pointing arrow to the right of “December”) to indicate that a second selection of the month button 503 (or a double click) will display the year-quarter-month viewer control as shown in FIG. 5e.

[0046] In FIG. 5e, a year-quarter-month viewer control 540 is displayed. Notice in this example that the previously selected period (here, the month December) is shown as still selected with the selected flag in month cell 541. Accordingly, a user is able to shift focus from a day in month view to a month in year view without losing track of the selected object. Specifically, the same period (December) that was previously selected is now shown using a different viewer control, the year-quarter-month viewer control 540. The viewer control 540 allows the user, with a single selection, to select a month, one of the four quarters (Q1-Q4), or the entire year (2007). The viewer control 540 functions similarly to the month-weekday viewer control 500. In this illustrated example, the drill-down flag (green arrow) is displayed in month cell 541, such as to indicate that the user may select the selected month a second time (e.g., by a mouse click) to return to the month-weekday viewer control 500, with the month still shown as selected (e.g., see FIG. 5d). For example, FIG. 5f illustrates the results of the user selecting month cell 541 to drill back down into the month of December, with the month-weekday viewer control 500 being displayed. Accordingly, through selection of the jump-up flag or the drill-down flag, the user can navigate the views using viewer controls (e.g., the nests) up and down different hierarchical groups of data.

[0047] In FIG. 5g, which continues the example of FIG. 5f, the user again selects December 6 to select that period, and, as a result of the selection, the drill-down button is displayed in cell 504 to indicate that a second selection of this cell will navigate to a day-shift-hour viewer control with the whole day (December 6) selected.

[0048] FIG. 5h illustrates the result of the user navigating using the drill-down button in cell 504 of FIG. 5g to display
the day-shift-hour viewer control 570. Once again, the previously selected period (here, the day December 6) has not changed, but is instead displayed using a different viewer control, in this case the day-shift-hour viewer control 570. In this illustrated embodiment, the P.M. hours are indicated as selected (the orange selected flags) in the outside ring, and the A.M. hours are indicated as selected (the orange selected flags) in the inside ring. The light and dark shadings within the ring represent daytime and nighttime hours. The three controls S1-S3 are used for selecting specific shifts and, here, are also illustrated as selected (the orange selected flags). Notice the green jump-up flag appears to the right of the date when the entire day has been selected (as is the case in this example), so as to indicate that the user can select it to return back to the month-weekday viewer control (e.g., viewer control 500).

[0049] In FIG. 5j, which continues the example of FIG. 5h, the user has selected a period corresponding to the hour ending at 7 PM by clicking the availability flag 571. In FIG. 5j, the user has selected the second shift S2, which in this example ends at 4 PM, by clicking the S2 button 572. In addition, in this example, the group selectors (S1-S3) shown on the left are configured for three 8 hour shifts, although in other embodiments the number of such selectors and/or the number of specific hours in each group is configurable.

[0050] The approach used in the illustrated embodiment of FIGS. 5a-5j can be easily and naturally extended to higher level viewer controls (e.g., century-year-decade) and/or to lower level viewer controls (e.g., hour-sub-hour-minute). In addition, the system used here for the UPS can be generalized and applied to other types of hierarchically nested object trees, such as described with respect to nesting navigators generally.

[0051] In some embodiments, applications and/or tools may provide a UPS to a user, such that the user can specify various periods of time for various purposes of the application and/or tool. For example, as previously noted, in some embodiments, a user may use an embodiment of the illustrated UPS to navigate and select various ranges, points, and/or periods of time, such as to cause statistics and/or other types of data corresponding to the selected various times to be displayed to the user. For example, in some such embodiments, selection of the various periods (e.g., selection of quarters, months, weeks, days, shifts, hours, etc.) using the nest viewer controls described above (e.g., year-quarter-month, month-week-day, and day-shift-hour viewer controls) may cause such statistics and/or other types of data corresponding to those statistics and/or other data to be appropriately displayed to the user. Example embodiments of user interface controls that use the nesting navigators described here are described in detail in U.S. patent application entitled “Multi-mode Viewer Control for Viewling and Managing Groups of Statistics,” filed concurrently herewith and in U.S. patent application entitled “Multi-mode Viewer Control for Viewling a Series of Statistical Values,” filed concurrently herewith, which are incorporated herein by reference in their entirety. In some such embodiments, a UPS may also provide a user with the ability to toggle between a navigation mode (that navigates the UPS but does not fetch and/or display/update any corresponding statistics and/or other data) and a selection mode (that does fetch and/or display/update corresponding statistics and/or other data), such as discussed elsewhere.

[0052] FIG. 6 is an example block diagram of an example computing system that may be used to practice embodiments of an example nesting navigator, such as the example Universal Period Selector, as described herein. Note that a general purpose or a special purpose computing system suitably instructed to perform the functions and behaviors described herein may be used. Further, the nesting navigator may be implemented in software, hardware, firmware, or in some combination to achieve the capabilities described herein. In a typical embodiment, the nesting navigator is implemented by or used by one or more other code modules as a data display, organization, management, and navigation tool. In some embodiments, the nesting navigator may support an application programming interface (“API”) to communicate with an instance of a nesting navigator or with one of the nest viewer controls implemented thereby. Also, in some embodiments, the nesting navigator may communicate with other modules and/or code logic using events or messages.

[0053] A computing system 600 used to implement a nesting navigator may comprise one or more server and/or client computing systems and may span distributed locations. In addition, each block shown may represent one or more such blocks as appropriate to a specific embodiment or may be combined with other blocks. Moreover, the various blocks of the application or other code module or logic 510 incorporating a nesting navigator may physically reside on one or more machines, which use standard (e.g., TCP/IP) or proprietary interprocess communication mechanisms to communicate with each other.

[0054] In the embodiment shown, computer system 600 comprises a computer memory (“memory”) 601, a display 602, one or more Central Processing Units (“CPU”) 603, Input/Output devices 604 (e.g., keyboard, mouse, CRT or LCD display, etc.), other computer-readable media 605, and one or more network connections 606. An example application (code module or other programming logic) 610 that incorporates one or more nesting navigators 612 is shown residing in memory 601. In other embodiments, some portion of the contents, some of, or all of the components of the application 610 or the nesting navigator 612 may be stored on or transmitted over the other computer-readable media 605. The components of the application 610, including the code used to implement the nesting navigator preferably execute on one or more CPUs 603 and manage the presentation of data, as described herein. Other code or programs 630 and potentially other data repositories, such as data repository 620, also reside in the memory 610, and preferably execute on one or more CPUs 603. Note, one or more of the components in FIG. 6 may not be present in any specific implementation.

[0055] In a typical embodiment, the application 610 includes one or more components including the application logic 611 and data 615 used to implement nesting navigator 612. Other and/or different components (and/or modules) 613 may be implemented. In addition, components of the application 610 may interact via a network 650 with application or client code 655, one or more client computing systems 660, and/or one or more data provider systems 665, for example, to supply the data presented by the nesting navigator 612. Also, of note, one or more data repositories 615 or 620 that contains the source data used to generate data displayed by the nesting navigator 612 may be provided external to the nesting navigator 612, for example in a knowledge base accessible over one or more networks 650.
In an example embodiment, the logic of the nesting navigator 612 is implemented using standard programming techniques. However, a range of programming languages known in the art may be employed for implementing such example embodiments, including representative implementations of various programming language paradigms, including but not limited to, object-oriented (e.g., Java, C++, C#, Smalltalk, etc.), functional (e.g., ML, Lisp, Scheme, etc.), procedural (e.g., C, Pascal, Ada, Modula, etc.), scripting (e.g., Perl, Ruby, Python, JavaScript, VBScript, etc.), declarative (e.g., SQL, Prolog, etc.), etc.

The embodiments described above may also use well-known or proprietary synchronous or asynchronous client-server computing techniques. However, the various components may be implemented using more monolithic programming techniques as well, for example, as an executable program running on a single CPU computer system, or alternately decomposed using a variety of structuring techniques known in the art, including but not limited to, multiprogramming, multithreading, client-server, or peer-to-peer, running on one or more computer systems each having one or more CPUs. Some embodiments are illustrated as executing concurrently and asynchronously and communicating using message passing techniques. Equivalent synchronous embodiments may also be supported.

In addition, programming interfaces nesting navigator API 617 to the nesting navigator 612 can be made available by standard means such as through C, C++, C#, and Java APIs; libraries for accessing files, databases, or other data repositories; through scripting languages such as XML; or through Web servers, FTP servers, or other types of servers providing access to stored data. The data repository 620 may be implemented as one or more database systems, file systems, or any other method known in the art for storing such information, or any combination of the above, including implementation using distributed computing techniques.

Also the example application 610 may be implemented in a distributed environment comprising multiple, even heterogeneous, computer systems and networks. For example, in one embodiment, some or all of the application logic 611, the nesting navigator 612, the nesting navigator API 617, the other components 613 and the data 615 are located in physically different computer systems. In another embodiment, various modules of the application 610 are hosted each on a separate server machine. Also, one or more of the modules may themselves be distributed, pooled or otherwise grouped, such as for load balancing, reliability or security reasons. Different configurations and locations of programs and data are contemplated for use with techniques described herein. A variety of distributed computing techniques are appropriate for implementing the components of the illustrated embodiments in a distributed manner including but not limited to TCP/IP sockets, RPC, RMII, HTTP, Web Services (XML-RPC, JAX-RPC, SOAP, etc.) etc. Other variations are possible. Also, other functionality could be provided by each component/module, or existing functionality could be distributed amongst the components/modules in different ways, yet still achieve the functions of a nesting navigator.

Furthermore, in some embodiments, some or all of the logic of the nesting navigator 612 may be implemented or provided in other manners, such as at least partially in firmware and/or hardware, including, but not limited to one or more application-specific integrated circuits (ASICs), standard integrated circuits, controllers (e.g., by executing appropriate instructions, and including microcontrollers and/or embedded controllers), field-programmable gate arrays (FPGAs), complex programmable logic devices (CPLDs), etc. Some or all of the programming logic and/or data structures may also be stored as contents (e.g., as executable or other machine-readable software instructions or structured data) on a computer-readable medium (e.g., a hard disk; a memory; a computer network or cellular wireless network or other data transmission medium; or a portable media article to be read by an appropriate drive or via an appropriate connection such as a DVD or flash memory device) so as to enable or configure the computer-readable medium and/or one or more associated computing systems or devices to execute or otherwise use or provide the contents to perform at least some of the described techniques. Some or all of the logic and data structures may also be transmitted as contents of generated data signals (e.g., as part of a carrier wave or otherwise included as part of an analog or digital propagated signal) on a variety of computer-readable transmission mediums, such as media 605, including wireless-based and wired/cable-based mediums, and may take a variety of forms (e.g., as part of a single or multiplexed analog signal, or as multiple discrete digital packets or frames). Such computer program products may also take other forms in other embodiments. Accordingly, embodiments of this disclosure may be practiced with other computer system configurations.

FIG. 7 is an example flow diagram of an example routine to navigate between multiple nests of an example nesting. The routine may be implemented, for example, by the nesting navigator 612 shown in FIG. 6. In block 701, the routine aggregates data objects into multiple nests of a nesting (e.g., such nests corresponding to a UPS described above). In block 705, after possible other processing, the routine displays a nest viewer control of one of the multiple nests (e.g., a month-week-day viewer). For example, the displayed nest viewer control may present multiple objects that are indicated as available for selection by a user (e.g., using availability flags). In block 710, the routine receives an indication of a user selection (e.g., a mouse click) of one of the data objects that is available for selection in the displayed nest viewer control. If the user selection is a first selection of a data object, the routine continues to block 715 to display an indication that selected data object has been selected (e.g., a selected flag). In some embodiments, the user selection of the data object may cause other flags or indicators to be displayed, such as a drill-down flag (e.g., if object is a bottom level object that is also a generator object of a sub-nest of the displayed nest) or a jump-up flag (e.g., if the object is a generator object that is also a bottom level object of a supra-nest of the displayed nest).

If instead the user selection received in block 710 is a second selection of a previously selected data object that is a bottom level object of the displayed nest (or a selection of a drill-down flag), then the routine continues to block 720 to display a nest viewer control of a sub-nest (e.g., a day-shift-hour viewer), with the selected data object presented as a selected generator object of the displayed sub-nest (e.g., a selected day of the day-shift-hour viewer). If instead the user selection is a second selection of a previously selected data object that is a generator object of the displayed nest (e.g., such as selection of a jump-up flag), then the routine continues to block 725 to display a nest viewer control of a supra-
nest (e.g., a year-quarter-month viewer), with the selected data object presented as a selected bottom-level data object of the supra-nest (e.g., the selected month in the year-quarter-month viewer). After either the view control for the sub-nest or the viewer control for the supra-nest is displayed in blocks 720 and 725, respectively, the user may interact with these viewer controls to select data objects that are available for selection in the corresponding nests and to navigate to other nests (e.g., sub-nests and/or supra-nests, etc.), such as described above in blocks 710-725.

After blocks 715, 720, and 725, the routine may determine whether there are more user selections to be received for a displayed nest viewer control, and if so, the routine returns to block 710 where an indication of a user selection is received. If instead, at block 730, it was determined that there are no more user selections to be received for a displayed nest viewer control, the routine continues to block 740 to optionally perform other operations. After block 740, the routine ends.

It will be appreciated that in other embodiments of the routine of FIG. 7 various other types of selections and navigation may also be available for navigating between multiple nests of a nesting. For example, in some embodiments, a selection-stepper control (discussed elsewhere) may be available for navigating between nests in the same nest-set (e.g., sibling or cousin nests), etc.


From the foregoing it will be appreciated that, although specific embodiments have been described herein for purposes of illustration, various modifications may be made without deviating from the spirit and scope of the invention. For example, the methods, systems, and techniques for navigating and selecting data discussed herein are applicable to other architectures. Also, the methods and systems discussed herein are applicable to differing protocols, communication media (optical, wireless, cable, etc.) and devices (such as wireless handsets, electronic organizers, personal digital assistants, portable email machines, game machines, pagers, navigation devices such as GPS receivers, etc.).

1. A computer-implemented method for providing navigation between a plurality of data objects aggregated into a plurality of hierarchical levels divisible into one or more nests of data objects, each nest containing a determined number of levels, each nest that is a sub-nest of another nest containing a level that is also a level contained in the another nest, comprising:

- under control of a computing system,
- presenting on a display screen a first nest viewer control for viewing a first nest of the one or more nests;
- in response to a first selection of an item that corresponds to a data object of the first nest, causing the item to be indicated as selected on the display screen;
- in response to a second selection of the item that corresponds to the data object of the first nest, presenting on the display screen a second nest viewer control for viewing a second nest of the one or more nests, the second nest being a sub-nest of the first nest, without changing the item indicated as selected; and
- in response to a selection of an other item that corresponds to an aggregation of at least some of the plurality of data objects of the first nest, presenting on the display screen a third nest viewer control for viewing a third nest, the third nest being a supra-nest of the first nest, without changing the other item indicated as selected.

2. The method of claim 1 wherein the determined number of levels is at least three.

3. The method of claim 1 wherein the level in the sub-nest that is also the level contained in the another nest is a generator object for the sub-nest.

4. The method of claim 1 wherein the level in the another nest is a bottom level item that corresponds to a selectable data object.

5. The method of claim 1 wherein the one or more nests of data objects define a time period navigation control.

6. The method of claim 1 wherein at least one of the first, second, or third nest viewer controls is a month-week-day selector control.

7. The method of claim 6 wherein the month-week-day selector control resembles a calendar.

8. The method of claim 1 wherein at least one of the first, second, or third nest viewer controls is a year-quarter-month selector control.

9. The method of claim 1 wherein at least one of the first, second, or third nest viewer controls is a day-shift-hour selector control.

10. The method of claim 9 wherein the day-shift-hour selector control uses a visual metaphor of a clock.

11. The method of claim 10 wherein the clock includes 24 separately selectable visual items, twelve of which correspond to an hour before noon and twelve of which correspond to an hour after noon.

12. The method of claim 9 wherein a time period of the day-shift-hour selector control is selectable in response to a single click of a mouse button.

13. The method of claim 1 wherein the second selection of the item that corresponds to the data object is performed using a separate user interface control.

14. The method of claim 1 wherein the user selection of the other item that corresponds to the aggregation of at least some of the plurality of data objects is performed using a separate user interface control.

15. The method of claim 1 wherein the first, second, and third nest viewer controls comprise a universal period selector.

16. A computing system for providing navigation between a plurality of data objects aggregated into a plurality of hierarchical levels divisible into one or more nests of data objects, each nest containing a determined number of levels, each nest containing a determined number of levels, each nest
that is a sub-nest of another nest containing a level that is also
a level contained in the another nest, comprising
one or more memories, and
a display device for displaying a visual representation of a
data structure stored in the one or more memories, the
visual representation configured to:
present a first nest viewer control for viewing a first nest
of the one or more nests;
in response to a first selection of an item that corresponds
to a data object of the first nest, cause the item to be
indicated as selected;
in response to a second selection of the item that corre-
spends to the data object of the first nest, present a
second nest viewer control for viewing a second nest
of the one or more nests, the second nest being a
sub-nest of the first nest, without changing the item
indicated as selected; and
in response to a selection of a different item that corre-
sponds to an aggregation of at least some of the plu-
rality of data objects of the first nest, presenting a third
nest viewer control for viewing a third nest, the third
nest being a supra-nest of the first nest, without chang-
ing the different item indicated as selected.

17. The computer system of claim 16, wherein the first,
second, and third nest viewer controls comprise a universal
period selector.

18. A computer-readable storage medium containing con-
tents that when executed cause a computer processor to pro-
vide a user interface for navigating between a plurality of data
objects aggregated into a plurality of hierarchical levels divis-
able into one or more nests of data objects, each nest contain-
ing a determined number of levels, each nest that is a sub-nest
of another nest containing a level that is also a level contained
in the another nest, by performing a method comprising:
presenting a first nest viewer control for viewing a first nest
of the one or more nests;
in response to a first selection of an item that corresponds to
a data object of the first nest, causing the item to be
indicated as selected;
in response to a second selection of the item that corre-
sponds to the data object of the first nest, presenting a
second nest viewer control for viewing a second nest of
the one or more nests, the second nest being a sub-nest of
the first nest, without changing the item indicated as
selected; and
in response to a selection of an other item that corresponds
to an aggregation of at least some of the plurality of data
objects of the first nest, presenting a third nest viewer control
for viewing a third nest, the third nest being a supra-nest of
the first nest, without changing the other
item indicated as selected.

19. The computer-readable storage medium of claim 18,
wherein the medium is a memory of the computing device
and the contents are instructions for controlling a computer
processor to perform one of the above methods.

20. The computer-readable medium of claim 18, wherein
the first, second, and third nest viewer controls comprise a
universal period selector.

* * * * *